

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
ÉCOLE POLYTECHNIQUE DE LOUVAIN
LABORATOIRE DE MICROÉLECTRONIQUE

Authentication Management for Secure Seamless Mobility

Nidal Aboudagga

*Thèse soutenue en vue de l'obtention du grade de
docteur en sciences de l'Ingénieur*

Composition du jury:

Pr. Jean-Jacques Quisquater (UCL - DICE) – Promoteur
Pr. Mohamed El-Toweissy (Virginia - Tech -USA)– Copromoteur
Pr. Gildas Avoine (UCL - INGI)
Dr. Yves Deswarte (LAAS - CNRS, France)
Pr. Olivier Pereira (UCL - DICE)

Louvain-la-Neuve, Belgique
2008

ABSTRACT

The immense success of wireless and broadband technologies over the last decade has been accompanied with an increasing number of wireless networks owned by different authorities. The main successful and attractive feature of these networks is the continued connectivity of mobile users, which has lead the network owners to tolerate increasing security issues. The challenged issues concern the authentication, the authorization and the accountability of users for fast roaming within and across networks.

The research presented in this thesis is focused on authentication to enhance the secure seamless mobility of wireless users. I first provide a classification and an analysis of wireless authentication schemes. This investigation clarifies the drawbacks and uncovers aspects of the existing authentication schemes. Based on this knowledge, I provide a common authentication model for wireless technologies. Subsequently, the thesis presents authentication schemes for fast roaming in WLAN, authentication architectures for mobile groups across network, an Imbricated protocol suite for both, within and across networks authentication, and finally a scheme suitable for first responder mobile networks. All these proposals are based on the generic authentication model.

ACKNOWLEDGMENTS

First of all, I would like to thank Jean-Jacques Quisquater for welcoming me at the Crypto Group and being my advisor. His support and criticisms were precious for the progress of my work. I am grateful also for the freedom of research he granted me.

I would also like to thank Mohamed El-Toweissy for accepting to be my co-advisor. His encouragement and help made me believe in my ideas. I acknowledge also the president of my jury Jean-Pierre Raskin and the members of my jury, Gildas Avoine, Yves Deswarte and Olivier Pereira for their valuable suggestions.

Working with Crypto Group members was a great experience. I am particularly grateful to Julien Cathalo for his friendship, Philippe Bulens for sharing his knowledge in latex, Olivier Pereira for his technical advices and Sylvie Baudine for her valuable administrative and human support.

For several years this work has benefited from the financial support of the Cooperation Technique Belge (CTB). Thanks to them for their flexibility and comprehension.

I am grateful to Skerdilajda Zana for her friendship, advices and corrections of my thesis. She changed my way of thinking and gave me the force and confidence to finish this work.

Thanks to my friends for their support during the PhD years, especially Karima Benameur who brought me back on the path of success, thanks to my friends Simone Moriconi, Emanuele Forlani and Miloud Zitout for their fidelity in my hard times.

DEDICATION

I would like to dedicate this work to all the children of my country; I hope they will have soon the freedom and the chance to fulfill their dreams.

I dedicate this work to my parents, for all the sacrifices they accepted for my education and my studies.

I dedicate this work to my wife for the support she gave me and to my babies Yassar and Iyass for the happiness they bring in my life, to Nader the best brother ever also to my sister Njoud and brothers Mohamed, Fares and Nour for their presences and affection.

Finally, I dedicate this work to Marianne Blume she made me discover and love Belgium.

CONTENTS

1. <i>Introduction</i>	1
1.1 Wireless Networks Characteristics	1
1.1.1 Advantages	1
1.1.2 Inconveniences	2
1.2 Principle of Authentication	3
1.3 Cryptographic Authentication Protocols	5
1.4 Motivation and Scope	8
1.5 The Prospects of Authentication and Security for Mobility . .	9
1.6 Work Organization	11
2. <i>Analysis of Wireless Authentication Protocols</i>	13
2.1 Introduction	13
2.2 Authentication in Ad Hoc Networks	14
2.2.1 Authentication Process Components	15
2.2.2 Authentication States for a Supplicant	18
2.3 Authentication Protocols Taxonomy	20
2.4 Classification Based On Authentication Function	20
2.4.1 Homogeneous	20
2.4.2 Heterogeneous	21
2.5 Classification Based on Type of Credential	22
2.5.1 Identity-based Credentials	23
2.5.2 Context-based Credentials	24
2.6 Classification Based on Credential Establishment	24
2.7 Towards an Authentication Management Architecture	25
2.7.1 Load Effect	27
2.7.2 Authentication Flows	27
2.7.3 Number of Servers	28
2.7.4 Hierarchical Deployment Model	29
2.8 Conclusion	31

3. <i>Authentication Across Wireless LANs</i>	33
3.1 Introduction	33
3.2 Background	34
3.3 Proposed Authentication Ticketing Scheme	37
3.3.1 At the Authentication Server:	38
3.3.2 At the Access Point (<i>AP</i>)	39
3.3.3 At the Station	40
3.4 The Proposed Token Authentication Scheme	41
3.4.1 <i>PMK</i> Distribution	42
3.4.2 At the Server	42
3.4.3 At the Access Point	43
3.4.4 At the station	44
3.5 Related Work	44
3.6 Analysis	46
3.7 Conclusion	48
4. <i>Group Authentication Protocol for Mobile Networks</i>	50
4.1 Introduction	50
4.2 The Network Model	50
4.3 Authentication Management Architecture	52
4.3.1 Architecture Components	52
4.3.2 Example of the Architecture	55
4.4 The Authentication Protocol	58
4.4.1 Secure Custodian Hiding Verifiable Encryption Scheme	58
4.4.2 The group authentication protocol	61
4.4.3 Mobile Group Operations	63
4.5 mGAP Security and Performance Analysis	66
4.5.1 Security Analysis	66
4.5.2 Performance Analysis	67
4.6 Related Work	68
4.7 Conclusion	69
5. <i>An Authentication Protocol Suite for Roaming Users and Groups</i> .	71
5.1 Introduction	71
5.2 Protocol Constructs	72
5.3 Imbricated Protocol Suite	73
5.4 Analysis	77
5.4.1 Security analysis	77
5.4.2 Performance Analysis	79
5.5 Implementation and Results	79

5.6	Related work	85
5.7	Conclusion	86
6.	<i>Two Level Signature Delegations For Mobile First Responder Authentication</i>	87
6.1	Introduction	87
6.2	Two-Level Delegation Authentication Scheme	90
6.2.1	Network Model	90
6.2.2	The Proposed Scheme	91
6.3	Background	93
6.3.1	Schnorr Signature	93
6.3.2	Schnorr Based Proxy Signature	94
6.4	Two Level Delegation Scheme	95
6.4.1	Threshold Delegation	95
6.5	Scheme Evaluation	97
6.5.1	Security Evaluation	97
6.5.2	Efficiency Evaluation	98
6.6	Related Work	101
6.7	Conclusion	102
7.	<i>Simulation Results</i>	103
7.1	Introduction	103
7.2	TOSSIM Simulator and TinyOS	103
7.3	Simulated Network Topology	104
7.3.1	Metrics and Mobility	104
7.3.2	Authentication Using mGAP and Proxy scheme	105
7.3.3	The results	106
7.3.4	Timeout Impact	107
7.3.5	Load Impact	109
7.3.6	Impact of the number of nodes	111
7.4	Conclusion	113
8.	<i>Conclusion</i>	115

LIST OF FIGURES

2.1	Functions in a generic authentication process	17
2.2	Node authentication states	19
2.3	classification based on nodes role	22
2.4	Classification based on type of credentials	23
2.5	Classification based on credentials establishments	25
2.6	10 × 10 Grid Topology. First AS placement model is shown in gray. Second AS placement model is shown in black	27
2.7	Authentication delay as the number of flows increases from 20-80 flows for 1 to 4 AS placed according to model I (a) and model II (b). The delay of each flows is averaged over 10 simulation runs	28
2.8	Packet loss when no authentication required and as the number of AS increases from 1 to 4 placed according to model I. Packet loss is averaged over 10 simulation runs	29
2.9	Authentication delay as the number of flows increases from 20 to 200 flows, for 1 to 4 AS placed according to model I. The delay for 20 to 200 flows is magnified in the embedded Figure	30
2.10	Flat Authentication Model. “S” denotes a source node, “D” denotes a destination node, and “A” denotes an authentication server	30
3.1	Latency sources in WLAN	34
3.2	The elements of the system	34
3.3	The trust relationship in the system as defined by the IEEE 802.11i security standard	35
3.4	The authentication ticketing exchanges.. . . .	41
3.5	Tokens distribution and utilization	43
3.6	Neighbor graph for mobile station authentication path	47
4.1	Mobile network model	51
4.2	Elements of authentication architecture with mobile access point . . .	53
4.3	Input and output of the home domain architecture elements	56
4.4	The protocol exchanged messages	64

5.1	Network model authentication possibilities	81
5.2	Generic proxy authentication exchange	82
5.3	Latency variation of Imbricated protocol for within and across network authentication VS symmetric proxy authentication. The same load is applied on V and HD and no relay between V and HD	83
5.4	Latency variation with increasing number of relay between V and HD and increasing load on the relay	84
6.1	The network model for two level delegation scheme	91
6.2	Certificates and shares distribution	93
6.3	Threshold authentication	93
6.4	The authentication process	96
7.1	TinyViz framework	104
7.2	Authentication latency for different flows' timeout	108
7.3	Authentication latency as the timeout of the authentication flows increases for mGAP(a) and proxy scheme (b). The delay of each flows is averaged over 20 simulation runs	109
7.4	Success ratio of authentication for different load at the server	110
7.5	Authentication latency increase as the load on the authentication server increase for both mGAP (a) Proxy scheme (b). The latency of each flows is averaged over 20 simulation runs	111
7.6	Success ratio of mGAP authentication as the number of nodes in the network increases	112
7.7	Authentication latency as the number of nodes in the network increases. The latency is averaged over 20 simulation runs	113

LIST OF TABLES

3.1	Comparison between our proactive scheme and the one in [48] during different path.	48
3.2	Comparison between our hybrid scheme and the reactive one in [48] during different path.	49
4.1	The protocol notation	62
4.2	Comparison of authentication protocols for low power devices.	69
5.1	The protocol notations	74
5.2	The protocols latencies	81
6.1	Exchanged messages.	97
6.2	Authentication cost comparison in infrastructure and threshold mode for two-level delegation scheme and hierarchical scheme based on the values of [25] on P3-977MHZ device.	100
7.1	Simulation metrics description	105

1. INTRODUCTION

Over the last decade, our communication ways have changed radically. Our life and work styles have become increasingly mobile. Consequently, our traditional ways to connect to each other and to data resources have become non convenient to our life style. Today the need of permanent connectivity to telephone and computer networks in business and personal life is an inherent aspect of the modern world. The need of networking persists in many areas where wired connection is either costly or impossible; moreover, the mobility of a part of network users has imposed a new cordless connectivity. Recently wireless telephony connectivity has proved to be a huge success, because it combines permanent reachability (anywhere, anytime) and mobility.

Wireless technologies combine the same features in computer networks and Internet access. Thus, many protocols have been developed for different wireless technologies and continue to be enhanced by researchers. This thesis studies the security aspects of wireless networks in different mobility contexts. The security in wireless networks, as in any other network, consists of fundamental functions like ensuring confidentiality, integrity of data, authentication of communicators, and other advanced security functions like privacy, authorization, non repudiation, availability and accounting.

The focus of this work is on entities authentication, because it is the main building block of the network security. However, designing and deploying a cryptographic security solution in a network depend strongly on its characteristics. Thus, we dedicate the next section of this chapter to the description of some wireless network particularities.

1.1 Wireless Networks Characteristics

Wireless networks present some advantages and disadvantages that influence any wireless security protocol design.

1.1.1 Advantages

There are many reasons to adopt wireless network services, for instance:

Mobility: The most obvious advantage of wireless connectivity is mobility. We can use our wireless mobile phone for the same call while traveling for hundred of kilometers. We can access Internet and home network resources from anywhere inside the company buildings or from outside, while visiting clients and partners. This connectivity enhances business' and employee's productivity. In fact, wireless networks can transform any place at any time into a work office.

Cost: Wireless devices become more powerful and cheaper every day. Wireless networks save cabling costs and headaches. Sometimes cabling is even impossible or dangerous.

Flexibility: The deployment and removal of wireless networks is very easy and rapid. The administrator can add and delete users without physical intervention on the network. The infrastructure for connecting one user or thousands of users is almost the same; we only need to deploy antennas and base stations.

Simplicity of use: Devices for wireless technologies can access to any allowed network without any further configuration.

Temporary deployment: Some wireless networks can be deployed only temporarily, for instance during the length time of a project or a meeting, or for a task duration (sensor networks).

1.1.2 Inconveniences

The main resource of wireless networking is radio waves. Wireless devices are constrained to operate in a certain frequency band. Each band has an associated bandwidth, which describes the quantity of data flows on a given transmission path. The use of the radio frequency (spectrum) is rigorously controlled by regulation licensing authorities like: The Federal Communication Commission (FCC) in USA, the European Radio communications Office (ERO) in Europe and the International Telecommunications Union (ITU). These organizations allocate the frequency ranges of operation for each type of application to avoid overlapping problems.

Wireless networks do not replace wired networks. Wireless networks bridge wireless users to wired resources given that the servers' location are fixed.

Usually, the speed of a wireless network is slower than that of a wired network, unless the regulatory authorities are willing to make the unlicensed radio spectrum bigger.

Radio waves, as network medium, can suffer from many propagation problems that may interrupt the radio link, such as multi-path interferences,

shadows and jamming attacks.

The security of wireless networks is more sensitive to a variety of problems that are not present in wired networks. In wireless networks, the data as well as the network management frames fly in the air, which makes them easy for sniffing by eavesdroppers. Moreover, the wireless medium is vulnerable to messages injection by intruder. Wireless networks have also fuzzy boundaries which are not obvious to control.

the networked wireless devices can be a target of physical attacks. hence, the secrets and sensitive data shared with the network could be extracted.

1.2 Principle of Authentication

Authentication is one of the most important objectives in information security. Its aim is the verification of the claimed identity or the guarantee of non manipulation of data by unauthorized parties. Before 1970, the secrecy and authentication were not separated. This changed with the discovery of hash functions and the digital signatures making secrecy and authentication become separate information security objectives.

1. **The data origin authentication** or message authentication techniques provide the receiving party with some certainty (through corroborative evidence) that the message indeed originated at the expected party's side. This implicitly ensures the non alteration of the received data by an unauthorized party. Frequently, either keyed or non keyed hash functions are used for data integrity. Knowing that the hash function is a function H that takes at the entrance an input x of finite arbitrary bitlength and outputs $H(x)$ of fixed bitlength n . The hash functions are used for data integrity in two ways.

Message authentication codes (MACs): are keyed hash functions whose specification dictates two distinct inputs, a message and secret key. Thus, the receiver that possesses the same symmetric key can check the message integrity by verifying the MAC correctness. The MAC provides also data origin authentication

Modification detection codes (MDCs): are also called message integrity codes MIC, and are non keyed hash function, whose specification requires a single input parameter. Differently from MAC, MDCs are combined with additional mechanisms such encryption to provide data integrity.

Digital Signature: was created by W. Diffie and M.E. Hellman in the

mid seventies [24] to get an equivalent of a manual signature. Differently from MDC and MAC, the digital signature provides integrity and prevent conflict between receiver and sender since it provides the undeniability. Thus a signer cannot deny its signed data. The digital signature is based on asymmetric cryptography, where the private key is the signing key and the public key is the signature verification key. No need for shared secret between receiver and sender.

2. **Entity authentication** or identification techniques convince one party (through acquisition of corroborative evidence) of the identity of other party involved, while also ensuring that the second party was active at the time the evidence was created or acquired. Communicating partners can be persons, users, services, or applications. This was the definition provided by Menezes et. al. in [46] where there is no distinction between Authentication and identification

For this thesis we consider the following definition and clarifications inspired from the Federal Information Processing Standards (FIPS 83 and FIPS 48): Authentication is the process of verifying and confirming the identity of the entity that is attempting to access a system. It is also the process of associating an individual with his unique identity, that is, the manner in which the individual establishes the validity of his claimed identity

Authentication is distinct from identification, which concerns the manner in which a user provides his unique identity to the IT system. The identity may be a name (e.g., first or last) or a number (e.g., account number). The identity must be unique so that the system can distinguish among different users. Identity should also be non-forgible so that one person cannot impersonate another

Authentication is also distinct from authorization, which is the process of giving individuals access to system objects based on their identity. Authentication merely verifies the claimed identity of the entity, but says nothing about its access rights.

Once a user/ entity authenticated, access control techniques may be used to mediate the user's access to data. A variety of methods are available for performing user authentication.

- Knowledge: e.g. passwords, Personal Identification Number (PIN), and secret or private keys whose knowledge is demonstrated in challenge-response protocols.

- Possession: e.g Physical keys, magnetic card, smart card, and hand-held customized calculators (time dependent passwords generators).
- Inherent characteristics: e.g: biometric properties like fingerprints, voice, retinal patterns.
- Location: evidence is presented that an entity is at a specific place for instance through GPS localization, like for a bank agent in his office.
- Others: reputation and role based authentication.

1.3 Cryptographic Authentication Protocols

The main building blocks of cryptographic authentication protocols are the encryption algorithms (symmetric and asymmetric) and integrity check values (MDC, MAC and signature). In this section, as well as in this entire thesis, we focus on entity authentication protocols. Most of the cryptographic authentication protocols can be classified in two categories:

- Authentication protocols involving a trusted third party (TTP), where two entities or more need to pass by a TTP to authenticate each other. A trusted third party is an entity or authority trusted by both, who uses this trust to secure the communication between them, e.g. certificate authority.
- Direct authentication protocol, in which the authentication is handled by direct involvement of a TTP. Frequently in direct authentication protocols the TTP exists offline and is not involved in each authentication exchange. As an example for this two types of protocols, we present Needham-Schroeder and X.509 strong two-way authentication protocol.

1. Needham-Schroeder Protocol (enhanced)[49] permits two entities Alice (A) and Bob (B) to authenticate each other with the help of a trusted third party(TTP). The protocol is based on symmetric cryptography. The TTP shares a secret key with each user, and uses these secrets to setup mutual authentication between the communicating parties and it also establishes a session key between them. Alice and Bob respectively share the symmetric keys $K_{A,TTP}$ and $K_{B,TTP}$ with the *TTP*.

- a- Alice generates a message containing an index number, i_A , her name A, Bob's name B, and the encryption with $K_{A,TTP}$ of these three parameters and an additional random number r_A

$$A \longrightarrow B : (i_A, A, B, \{r_A, i_A, A, B\}_{K_{A,TTP}})$$

- b- Bob generates a random number r_B , encrypts it together with i_A , A and B using $K_{B,TTP}$ the shared key between Bob and the *TTP* and sends the following message to *TTP*:

$$B \longrightarrow TTP : (i_A, A, B, \{r_A, i_A, A, B\}_{K_{A,TTP}}, \{r_B, i_A, A, B\}_{K_{B,TTP}})$$

- c- The *TTP* decrypts the received message, generates a new session key $K_{A,B}$ and generates two encrypted messages one for Alice and one for Bob, and sends both to Bob.

$$TTP \longrightarrow B : (i_A, \{r_A, K_{A,B}\}_{K_{A,TTP}}, \{r_B, K_{A,B}\}_{K_{B,TTP}})$$

- d- Bob decrypts his part of the received message using $K_{A,B}$, and verifies that r_A is identical to the random number he generated at the second step of the protocol and sends to Alice her part of the message.

$$B \longrightarrow A : (i_A, \{r_A, K_{A,B}\}_{K_{A,TTP}})$$

- e- Alice decrypts the received message using $K_{A,TTP}$ and verifies if the contained random number r_A matches the one she generated in the first step of the protocol. Now Alice can use $K_{A,B}$ for encrypted communication with Bob, she can be sure of his authenticity, since only *TTP* could have generated $\{r_A, K_{A,B}\}_{K_{A,TTP}}$ and no attacker is able to alter $\{r_A, i_A, A, B\}_{K_{A,TTP}}$ which she generated in the first step of the protocol. The same argument applies to Bob. However this protocol requires that Alice and Bob have to completely trust the correct functioning and honesty of the *TTP*.

2. X.509 strong two-way authentication protocol[46]: Direct authentication protocols, as mentioned before, do not require the online participation of the trusted third party. However, when the two parties do not know each other, neither share a secret key or know each other's public key, then some security management is needed. For this in 2000, an international recommendation X.509 was published to provide framework of the use of certificates for authentication. The standard describes certification of public keys and certificate handling.

The X.509 recommendations for public key certificate explain the way to authenticate the public keys. The public key certificate is issued by a so called certificate authority (*CA*). The *CA* signs a certificate for a public key using her private key to testify that the public key in certificate belongs to the subject named in the certificate.

The following X.509 strong two-way authentication protocol [46] provides mutual authentication between Alice (*A*) and Bob (*B*) and optional key transport and key authentication.

a- Notation:

$P_X(y)$ denotes the result of applying X 's encryption public key to data y .

$S_X(y)$ denotes the result of applying X 's signature private key to y .

r_A, r_B are never re-used numbers (to detect replay and impersonation).

$Cert_X$ is a certificate binding party X to a public key suitable for both encryption and signature verification.

b- System setup:

Each party has its public key pair for signature and encryption.

A must a priori acquire (and authenticate) the encryption public key of B and vice versa. Or they both have an authentic public key of the CA .

c- Protocol messages. (An asterisk denotes items are optional.) Let

$$D_A = (t_A, r_A, B, data_1^*, P_B(k_1)^*), D_B = (t_B, r_B, A, r_A, data_2^*, P_A(k_2)^*)$$

d- Protocol actions:

- A obtains a timestamp t_A indicating an expiry time, generates r_A , optionally obtains a symmetric key k_1 and sends the first message to B .

$$A \longrightarrow B : cert_A, D_A, S_A(D_A)$$

- B verifies the authenticity of $cert_A$ (checking the CA signature, the expiry date, etc), extracts A 's signature public key and verifies A 's signature on the data block D_A . B then checks that the identifier in the received message specifies herself as intended recipient, that the timestamp is valid and that r_A has not been replayed.

If all checks succeed, B declares the authentication of A successful, decrypts k_1 using his private key, and saves this now-shared key.

For the other authentication direction (B to A), B obtains timestamp t_B , generate r_B and sends the following message to A

$$B \longrightarrow A : cert_B, D_B, S_B(D_B)$$

- A carries out actions that are analogous to those carried out by B . If all checks succeed, A declares the authentication of B successful, decrypts k_2 using her private key, and saves k_2 for subsequent use. Now A and B are authenticated to each other and share mutual secrets k_1 and k_2 .

1.4 Motivation and Scope

This thesis provides several new authentication solutions for mobile entities in wireless environments. Today, there is no global solution for different mobility contexts in wireless networks. Nevertheless, complementary and co-existing solutions are more than welcome. Active organizations like police, army, or first responder agencies, possess different mobile parts of wireless networks, where the node authentication is crucial for their global security. Therefore, our complementary authentication solutions for these kind of organizations are more than needed (see SPEECH/04/291: European Commission, responsible for Enterprise and the Information Society: Mobile Communications: future visions and challenges).

The security of any communication between equal peers, requires the mutual authentication of the communicators. This can be established with one of the numerous authentication protocols in the literature. However, the wireless environments and the mobility make this operation difficult. The wireless nodes change their attachment point to the network in wireless local area network (WLAN), so re-authentication is needed. They change their neighbors in ad hoc network and authenticate to them. Moreover, wireless nodes move across networks where access to the services also requires authentication. This imposes different way of authentication management specifically adapted to the wireless and mobility context. This thesis presents adequate and efficient authentication solutions for secure seamless mobility in different contexts.

An organization for rapid intervention in rescue and disaster situation possesses different network parts (wired, wireless, mobile, etc). Such organization manages the access to sensitive resources and data for its members. The mobile users while roaming across the organization's buildings, change their attachment points to the network, but the time sensitive applications on their devices continues the application execution. This last needs fast authentication schemes for roaming in the WLAN. Such organizations, where mobile users recurrently roam among the organization's Access Points (*AP*) need an authentication scheme that takes into account this aspect to lighten the re-authentication process. Our Token and Ticketing authentication schemes are very well adapted to these needs and show better performance then the existing schemes.

Sometimes a part of the same organization has to intervene in disaster situation, as a specialized mobile team to execute a specific task. The mobile group (e.g. group of sensor nodes) must collect and route the information to the available networks. However the connection to an available

network involves a new authentication operation. For this purpose, we introduced mobile Access Point (*AP*) or Gateway as a support for mobile group authentication. The specialized group can be composed of one or more mobile wireless *AP* or gateways (transported by agent, robot, or vehicle); that is/are accompanied with mobile wireless sensors which collect information beforehand in order to route them to the mobile gateway. The mobile *AP* or gateway can be provided with several radio interfaces to transfer the information to the home organization network if it is in the range. The gateway can also transfer the information to another network in the range like health care partner network. Sharing and routing the collected information by the mobile group among networks, necessitates a specific authentication management architecture. This thesis presents a scheme that permits data sharing between a mobile group and different interested organizations.

Some disaster situations need a global cooperation between intervening organizations. Thus, the members of our previously mentioned organization have to cooperate intensively with the members of other organizations and first responder's agencies. The different organizations and first responder's agencies deploy their mobile wireless networks in the intervention area. To route the information to the organization's analysis bases, as well as for information sharing on the ground, the mobile nodes need to recognize and trust each other. This unavoidable trust for collaboration calls for a robust and flexible authentication scheme. The scheme must tolerate ad hoc, infrastructure communication, non centralization of decisions, new coming and leaving nodes and must be very robust to attackers. To deal with these needs we designed the two level signature delegations for mobile networks. This scheme allows local authentication of mobile nodes across networks.

In summary, this work offers complementary solutions for different wireless network needs and applications. These solutions can fit different security needs of different mobility contexts.

1.5 *The Prospects of Authentication and Security for Mobility*

Wireless and mobile networks usage has led to a radical change of the network security thinking. This is mainly due to the constrained resources, the unpredictable mobility, and the physical vulnerability of both, wireless nodes and radio channels. Consequently, security protocols for wireless portable and mobile devices have to deal with CPU limitations, small memories, narrow bandwidth and low battery power. These constraints had blocked the deployment of sophisticated and of computation intensive security protocols in wireless environments.

The work contained in this PhD thesis is based on the strategy that uses some aspects of wireless networks as an advantage to support the designed solutions. These strategies are:

(i) Transportability of credentials, which allows large mobility freedom to the mobile nodes in wireless and mobile networks. This idea is used in the solution dedicated to fast roaming in WLAN, the solution provided for cross network roaming of mobile groups in mGAP (group authentication protocol for mobile network) and for the first responder networks. The transportability of credential provides a credential similar to a passport, valid for authentication at different roaming destinations. I believe that such an instrument is going to be part of the security of a world in permanent connectivity. The transportable credential for roaming still needs some research to support reputation based and trust based credentials. For instance, a trader of eBay, with good reputation credential, would be able to use its credential to authenticate to a commercial hotspot or buy at online Amazon shop.

(ii) Dynamic mobility prediction: A minimum of mobility prediction is a non-avoidable complement to any security solution in wireless networks. This prediction must be dynamically upgradeable. It must also support, in the beginning, some default unavoidable roaming destinations. This list of destinations can anyway be updated according to mobile node preferences, network resources, policies, or geographic position. This is applied in my work in the profiling in mGAP.

This element of the secure solution design, as the transportability of credentials, must be enhanced by further research, because it is indispensable for the creation of transportable credential. The future mobility prediction scheme must not be network dependant, it can be updated on the fly and it cannot constrain roaming to non-predicted destinations. Moreover, others technique for mobility prediction in different wireless context could explore game theory for proactive credential distribution.

(iii) Support with powerful entity: The huge demand of applications and services via wireless networks have pushed our thinking to adopt ideas that were not even imaginable in wired networks. The future solutions must accept more often the use of mobile entities like access points, base station, wireless routers, and servers, to support the security for ordinary mobile nodes. As I did in mGAP, the mobile group is supported by mobile access point, thus the security operations is performed by the mobile AP on behalf of the ordinary nodes. This possibility must be exploited for new security designs that transform mobility to an advantage in services of security.

(iv) Authority delegation: We believe that centralization is A brake for

security solution design for mobility in wireless environment. Therefore, distributing and delegating the security authority among the networks nodes is one of successful strategy to be widely deployed in the future for more efficiency. This is the aim of the chapter “Two-level signature delegation for mobile First Responder Authentication”. Future research can combine the delegation of authority, concerning security, with mobility prediction and the support of mobile powerful entities.

(v) Using cheap resources consuming primitives: The mobile nodes resources are in constant growth. Thus, they can support in few years more sophisticated and resource consuming solutions. Moreover, the recent progress in elliptic curve cryptography (ECC) gives the opportunity to design and to implement solutions for wireless and mobile devices that was not possible in the past (less storage, less computation). These strategies are explored in our Imbricated protocol.

(vi) The anonymity and privacy are two security concepts that were changed by the mobility enabled by wireless communication. For instance, mobile network required mobile routing scheme in open wireless medium. This opens the network to many additional challenges of privacy and anonymity. The mobility have added more issues of privacy and anonymity that were left undressed in the solutions for fixed infrastructure. Some of theses challenges are: new venue anonymity, the privacy of ad hoc network topology and the privacy of mobility patterns. Mobility had pushed the thinking way of privacy and anonymity. Thus Anonymity and privacy have become a hot topic of research where more creativity is needed. This aspect of security is not studied in this thesis, but it is one of the extension perspective of the solutions provided in next chapters.

My PhD thesis traces a path that explores some opportunities listed above using such strategies, but this work is a beginning of a wide possibility of combinations of these strategies that constitutes a future research agenda.

1.6 Work Organization

This thesis is organized as follows. The first part is an introduction to several concepts, all related to authentication and wireless networks. Four new results are then presented in the second and third parts.

The first part develops a generic authentication process for wireless environments and node states during this authentication process. Moreover, we establish a taxonomy of existing authentication schemes in different wireless environments. While the end of this chapter presents some weaknesses of the studied schemes as well as uncovered authentication aspects, the simu-

lation results prove the unfairness of certain claims in the literature.

The second chapter introduces all the concepts and challenges of authentication while roaming across *AP* in wireless networks. This chapter provides two alternative solutions for fast authentication of roaming nodes. We compare our schemes to the existing ones and show their advantages.

The third chapter of the thesis describes new schemes for mobile group authentication. The mobile group is supported by mobile *AP*/Gateway which authenticate on behalf of the group using our new authentication scheme based on profiling architecture. This solution allows mobile group to locally authenticate at any of the set visited domain in its profile.

Chapter four combines two protocols to provide a strong authentication solution for both scenarios where roaming across networks is followed by within visited network.

Chapter five of the thesis presents our construction based on proxy signature for two levels of certification delegation. This authentication solution delegates the certificate authority of one higher authority to several sub-authorities. These certify their subscribers on behalf of the high authority and delegate to them the certification authority with some restrictions. Thus, when this solution is applied to mobile networks like first responder networks, the authentication of mobile node across networks become a local operation. Our solution succeeds where the other solution failed. It permits decentralization of authentication operation, tolerates different levels of security, requests less computation and storage and accepts leaving and joining nodes.

Finally, we conclude this thesis with some research issues in future wireless networks.

2. ANALYSIS OF WIRELESS AUTHENTICATION PROTOCOLS

2.1 *Introduction*

Wireless networks exist in two modes, infrastructure and ad hoc mode. The protocol analysis in this chapter provides a classification of the big variety of authentication schemes in ad hoc wireless networks. The same criteria are used to classify authentication scheme in infrastructure networks. The interest in ad hoc networks largely stems from the ability to rapidly deploy them under both normal and critical conditions. These networks can be quickly deployed in situations where not only no infrastructure exists but it would be even impractical or infeasible to deploy one.

In such an infrastructure-less network, nodes are expected to cooperate to perform essential networking tasks such as routing. In order to provide network-wide connectivity, nodes are expected to route data packets on behalf of other nodes, which want to reach resources out of their transmission range.

Ad hoc networks can be classified into static and mobile networks. Typically sensor networks (SensNets) are static ad hoc networks. On the other hand, mobile ad hoc networks (MANETs) are autonomous systems of mobile nodes that are free to move at will. A hybrid network may also exist. For example sensor nodes can form a tier in a network that is managed by a higher tier of mobile gateway nodes. From a security standpoint, ad hoc networks face a number of challenges. The wireless medium has no observable boundaries and is significantly less reliable than a wired one. Unlike wire-line networking, where an attacker must physically break into the network infrastructure, tap into network cables, or logically break through several lines of defenses (such as firewalls) before he can take control or tamper with any network component, wireless attacks may come from anywhere and from all directions [40]. Additionally, the lack of a clear line of defense and traffic concentration points poses a challenge to deploying security solutions in ad hoc networks. The broadcast nature of the transmission medium and the dynamically changing topology add even more complications. Furthermore, the reliance on node collaboration as a key factor of network connectivity

presents another obstacle. In order to provide network security, support for authentication, confidentiality, integrity, non-repudiation and access control should be provided. We believe that authentication is the cornerstone service, since other services depend on the authentication of communication entities [28][57]. Authentication supports other security function such as privacy protection data confidentiality by ensuring that the entities involved in an exchange verify and validate one another without ambiguity before any transaction. In addition, it supports confidentiality and access control by allowing access to services and infrastructure to authorized entities only, while denying unauthorized entities access sensitive data. A significant number of authentication protocols have recently been proposed for ad hoc networks [85][78][9][76][22][84][60] [50][51][72][?][79] [63][40][28][11]. A classification is needed to interpret the similarities between sets of related protocols and to understand the motivation behind each of them. A classification also enables us to better analyze and compare protocols with respect to their encapsulating class rather than comparing individual protocols, to identify common vulnerabilities and attacks against each class of authentication protocols, and to identify common architectural elements in each class. This chapter presents a new taxonomy for the classification of authentication protocols in ad hoc and infrastructure networks. We identify three major criteria for classification, based on a node's role in the authentication process, the type of credentials used for authentication and the phase during which the establishment of credentials takes place. The chapter also motivates the need for an authentication management architecture and presents some open research issues.

The remainder of this chapter is organized as follows. In Section 2.2 we introduce a generic authentication process model in an wireless network and the authentication states of a supplicant (the entity requesting authentication). In Section 2.3 we provide an overview of our taxonomy and present the three classification criteria proposed. In Sections 2.4 and 2.6 we discuss each of the three primary classes of the taxonomy. In Section 2.7 we present an analysis motivating the need for authentication management architecture. Finally, Section 2.8 concludes the chapter .

2.2 Authentication in Ad Hoc Networks

Authentication is a process that involves an authenticator and a supplicant using an authentication protocol to verify the credentials presented by the supplicant, to determine the supplicant's access privileges. A Trusted Third Party (TTP) may be involved as part of the authentication protocol. The

supplicant is an entity who tries to access to some protected resources after being authenticated via an authenticator. An authenticator is an entity that protects and controls access to some resources. The authenticator facilitates the authentication process and makes authentication decisions. An authentication protocol is a sequence of message exchanges between entities (supplicant(s) and authenticator(s)) that either distributes secrets to some of those principals or allows the use of some secret to be recognized [20]. A credential is an identifier that can be used to authenticate a supplicant with high confidence. Finally, a Trusted Third Party is an entity that is mutually trusted by the supplicant and the authenticator and that can facilitate mutual authentication between the two parties. An entity, be it a supplicant or authenticator, may be any of the following:

- Person: A person is a human user who is authenticating before accessing his authorized resources (for example to use the email service offered by the university).
- Agent: An agent is a program that after authentication performs some service on a regular schedule without the user's immediate participation.
- Service: To access a service, such as an online banking system, a supplicant must first authenticate itself to the service before using access rights.
- Node: A node usually refers to a computing device that is connected to the network. Networks can have tens, thousands or even millions of nodes. Laptops, personal digital assistants (PDA), sensors, and personal computers (PC) are all examples of nodes. Nodes also authenticate to the network before being part of it.

2.2.1 Authentication Process Components

A generic authentication process has six major phases as shown in Figure 2.1. Bootstrapping is the first phase, where a supplicant is securely provided, either offline or online, with something that it should have (a key) or something that it should know (a password), which authenticators would trust as a proof of the supplicant's eligibility to access protected resources or offer service. In [22], for example, bootstrapping is done by assigning a global network key to each new node joining the network, while in [78], nodes are bootstrapped by assigning each a list of trusted nodes. Once

the bootstrapping phase is completed, the supplicant is ready to participate in the network. The pre-authentication process is when a supplicant presents its credentials to an authenticator in an attempt to prove its eligibility to access protected resources or offer services. In [22] new nodes must demonstrate knowledge of the global network key (i.e. using the family of protocols based challenge-response authentication in which one party presents a question (“challenge”) and another party must provide a valid answer (“response”) to be authenticated). Once the supplicant’s credentials are verified, a credential establishment process is invoked to establish the supplicant’s new credentials, which are used as a proof of its identity and as a verification of its authorized state thereafter. A credential could be a symmetric key, a public/private key pair, a commitment of a hash key chain, or some contextual information. The established credentials might be tagged with an expiration date after which the supplicant has to re-negotiate a new “certificate” or credentials. In Deng’s et al. work [22], the node is assigned a portion of the private key of the network in a (k, n) threshold cryptography mechanism. In the Thonet et al. work [78], the authenticating parties use a chain of trust established between nodes in their trusted list to generate and perform a key exchange between them. In Weshoff et al. work [79], a commitment key to a TESLA [61] based one-way key-chain is generated and distributed as a node’s credentials. Upon success of all of the steps above, the supplicant is considered authenticated, which means that its claimed identity is verified, thus the resources protected by the authenticator could become accessible. Within the authentication state, all communication between the supplicant and the authenticator is authenticated by the source and validated at the destination using the established credentials. While authenticated, a supplicant’s behavior is monitored for fear of its being compromised or misbehaving. A compromised supplicant may get its credentials revoked (as described by Zhu et al. [84]) or its re-establishment of credentials request denied when its credentials expire. In both cases, the supplicant is isolated from the network. In this chapter, we focus on node-to-node authentication. To better understand the authentication process and protocols, we describe the authentication state diagram for a supplicant in the next section. The authenticator’s state diagram may be easily constructed following the supplicant’s state diagram and therefore it is not described here.

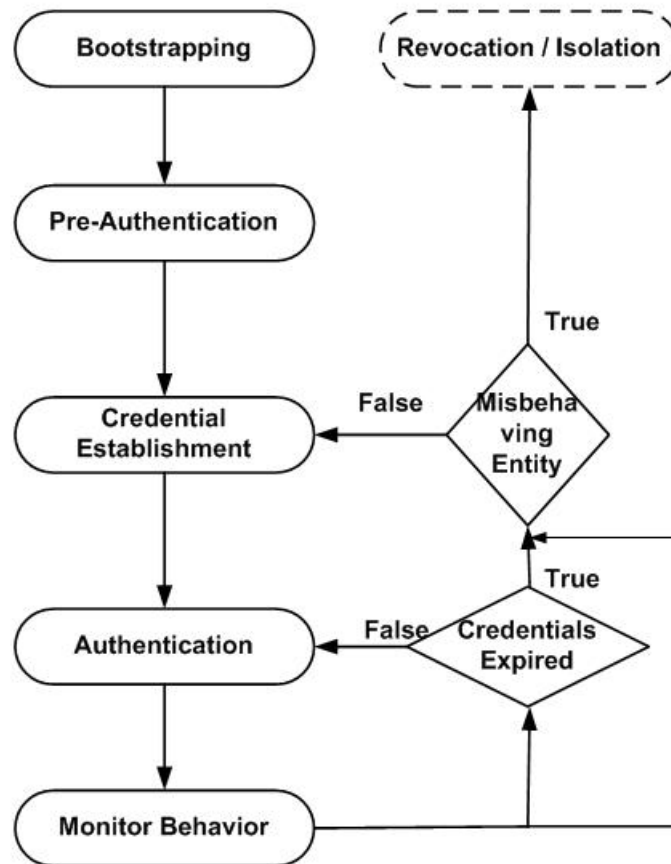


Fig. 2.1: Functions in a generic authentication process

2.2.2 Authentication States for a Supplicant

The state diagram in Figure 2.2 represents possible states of a supplicant during the authentication process. The first state initializes the supplicant. In this state, the supplicant is usually supplied with necessary tools to carry on an authentication function. These tools could be supported by authentication protocols (e.g., TESLA, 802.1x), authentication credentials (e.g., signed certificates), or identities of trusted entities. At the end of the initialization state, the supplicant and the authenticator have all necessary tools to perform successful authentication. Once a supplicant is initialized, it is ready to move on to the next state, called discovery. During the discovery state, a supplicant scans for reachable services of interest. Each available service is expected to advertise its presence and to list service-access requirements. A reachable service is one that is capable of directly making the supplicant aware of its presence (e.g., through periodic advertisements). At the end of the discovery state, a supplicant has a list of reachable services and the service-access requirements for each of them. The following state is the selection state. Based on the list of reachable services and the service-access requirements of each, a supplicant filters accessible services of interest. The supplicant matches the tools, it was supplied with during the initialization state, to the service-access requirements advertised by each service. If none of the services match, the supplicant goes back to the discovery state. At the end of the selection state, a supplicant has a list of matching accessible services that are of interest to it. The next state is the authenticating state. The supplicant uses the tools it was supplied with during the initialization state to attempt to authenticate to the authenticator. If the authentication process was successful, the supplicant moves to the authenticated state; if it fails the supplicant goes back to the discovery state. Within the authenticated state, the supplicant is considered trusted and is given appropriate access privileges to resources protected by the authenticator. The supplicant is bootstrapped with credentials that can be used to prove its access rights from there after. Following the authenticated state, the supplicant frequently enters an evaluation state where its behavior is examined. Based on the outcome of the evaluation process the supplicant could either return to the authenticated state (i.e. well behaving) or it is put under probation (i.e. selfish or malicious). Next, in the probation state, the supplicant receives a penalty if the probation result is an inappropriate behavior. Eventually, the supplicant would be re-evaluated and given a chance to recover.

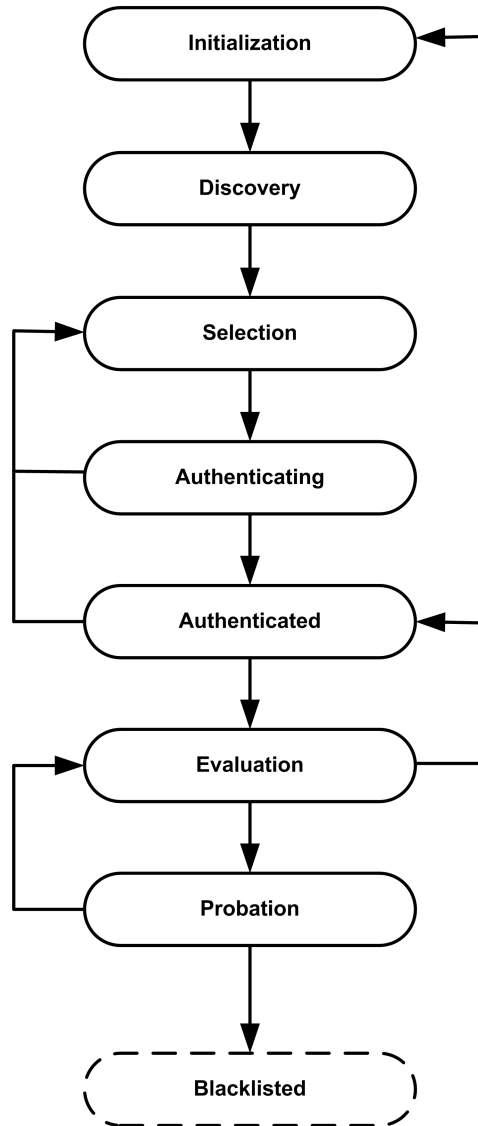


Fig. 2.2: Node authentication states

2.3 Authentication Protocols Taxonomy

We present a taxonomy based on the role played by nodes in the authentication, the type of credentials and when credentials are established. Authentication protocols described in the literature have introduced a variety of ways in which the authentication function may be carried out. Some protocols assume reliance on a third party that is trusted by all nodes. The trusted third party represents a service whose signature on a supplicant's credentials is considered a proof of its identity and on which the authentication decision are made. On the other hand, other protocols assume no such service in the network. The first classification of our taxonomy recognizes such differences by categorizing authentication protocols based on the roles assigned to nodes in the network with respect to the authentication operation. Based on that, authentication protocols can be classified into two classes: homogeneous and heterogeneous. The second classification recognizes different types of credentials used for authentication and categorizes authentication protocols based on that. As stated earlier, a credential is a unique identifier that can be used to authenticate a node with high confidence. Credentials may be classified into two classes. The first class identifies the supplicant based on a unique possession, while the second class identifies the supplicant based on context. The third classification recognizes the phase when credentials are established. Some protocols establish credentials prior to node deployment, while other protocols assume credentials are established post node deployment. A third possibility exists, when some credentials are pre-distributed offline, but the actual credentials used for authentication are derived from the pre-distributed credentials. While other bases for classification are possible, we believe that this classification is important since it captures variations shown in the literature in two important components that affect the operation of authentication, the deployment scenario of an authentication infrastructure and the means for authentication.

2.4 Classification Based On Authentication Function

2.4.1 Homogeneous

Homogeneity indicates that all nodes in the network have the same role with respect to the authentication operation. This class of authentication protocols assumes that nodes in the network either make authentication decisions autonomously or they depend on information contributed by other nodes in the network to make such decisions. Under the dependent homogeneous

class of authentication protocols, authenticators rely on information from their trusted peers to make authentication decisions. Trust based mechanisms that use trust chains (i.e., recommendations from trusted nodes) fall under this class. On the other hand, in the autonomous homogeneous class, authenticators make authentication decisions autonomously without relying on their peers or any overlaying infrastructure. The use of demonstrative identification, identity based cryptography and reputation based mechanisms such as [66] is common among protocols in this class. In general, trust based mechanisms fall under the homogeneous class of authentication protocols ([72] provides seven different classes of trust that might be required in the interaction between entities wishing to communicate securely).

2.4.2 Heterogeneous

The heterogeneous class of protocols indicates that nodes in the network have different roles with respect to the authentication operation. This suggests that there is an underlying service in the network that is meant to aid other nodes in making authentication decisions (e.g., a trusted third party). The underlying service could be centralized or decentralized.

Centralized: one specialized entity (node, server) is responsible for providing that service, like in ad hoc network with online authentication authority. Under this class can fall the authentication protocol in wireless local area network (WLAN) in the standard IEEE 802.11i as well as the the protocol for roaming node authentication across the same WLAN. Next chapter details those schemes and presents our schemes for the same purpose.

Decentralized: several specialized entities (nodes, servers) are responsible for providing that service. The decentralization can be seen in the division of the authentication function among several specialized entities in the same network or across different networks. This class is divided into three subclasses.

Distributed: service nodes are deployed anywhere in the network responding to service requests from any node, like the partially distributed certificate authority of Zhou et. al. in [82] and the symmetric key based Pirzada et. al. scheme in [63]. The specialized node in this subclass intervenes collectively and collaboratively for the authentication function.

Clustered: nodes are clustered and each cluster has a unique provider of the authentication service (the head cluster). The nodes involved in the authentication function can be in different clusters in the same network or in different networks like Venkatraman et. al. in [76] and Bechler et. al. scheme in [11]. We can extend this definition if it is considered that the

authentication servers of networks in infrastructure mode are head clusters. Thus the authentication function of nodes in ad hoc networks authenticating to each others in/across clusters through the heads clusters are in the same class as the nodes in infrastructure mode authenticating via authentication servers. The mobile node of WLAN authenticating at visited networks by proxy to its home authentication server also falls under this class (e.g., Mobile IP authentication). Finally the enlargement of clustered class permits our group authentication scheme to fall under this class.

Hybrid: Our study permits us to define for roaming nodes a new hybrid authentication scheme, where the authentication function is clustered as well as distributed in a complementary way. Thus the authentication process can be performed either with the authentication server of the visited network in infrastructure mode (in clustered way) or with the specialized collaborating nodes in the network (in distributed way). Hence our two levels delegation scheme which is detailed in chapter 6 falls under this new subclass.

The above classification is sketched in Figure 2.3

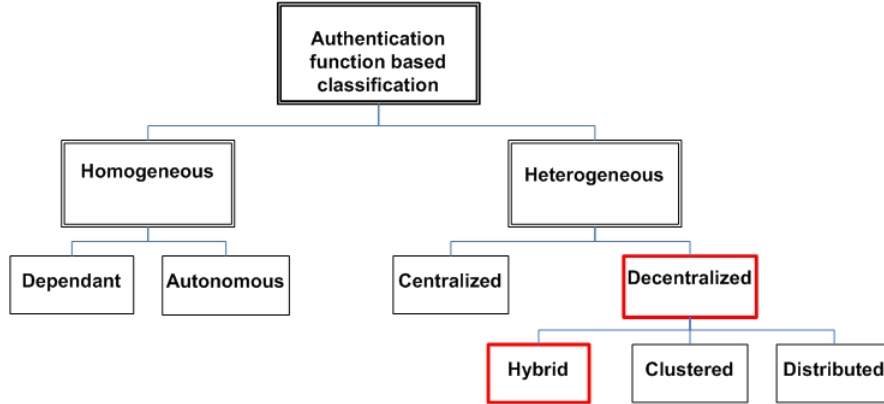


Fig. 2.3: classification based on nodes role

2.5 Classification Based on Type of Credential

This classification categorizes node authentication protocols based on the type of credentials used for authentication. Credentials can be classified into two classes: identity-based and context-based (see 2.4).

2.5.1 Identity-based Credentials

This category recognizes a unique possession owned by the supplicant that could be used to identify it with high confidence. Usually, this is in the form of a key that is known to be unique to the supplicant. The authenticator could be assured of the supplicant's identity if it is certain that the supplicant possesses that key. Identity based credentials can be further classified into encryption based and non-encryption based. An encryption based identity credential is a piece of information produced and cryptographically signed using the key possessed by the supplicant in order to verify its possession of the key, and hence prove its identity. In order to verify the supplicant's identity, the authenticator must either possess the same key (symmetric key cryptography), or the public-key component of the private-key owned by the supplicant (asymmetric key cryptography).

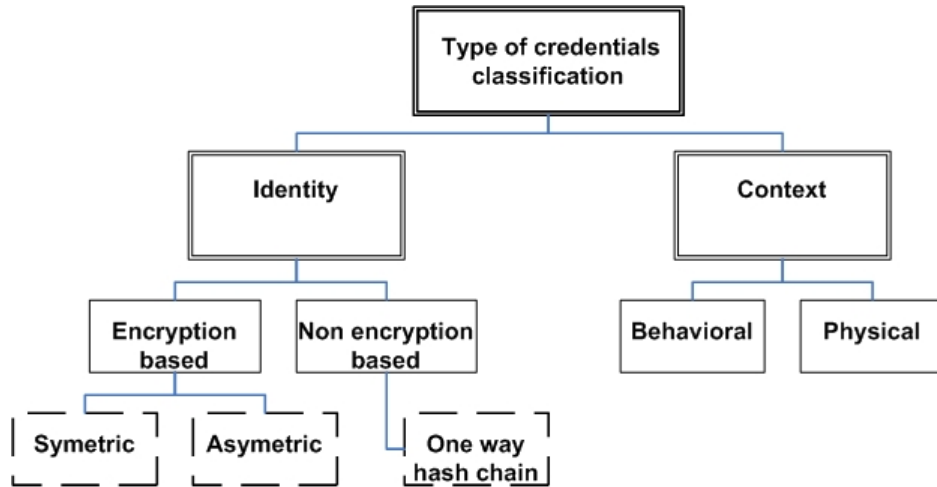


Fig. 2.4: Classification based on type of credentials

Symmetric key based authentication is more common in sensor networks since it is less resource dependent compared to the asymmetric key. On the other hand, asymmetric key based authentication, or public key cryptography, requires deployment of a public key infrastructure (PKI). In other words, it requires the presence of a trusted authority whose function is to bind entities' identities to their public keys and issue a signed certificate proving their authenticity. The service of such an authority must be available anytime anywhere.

One form of non-encryption based identity credential is information that is hashed using a one-way key-based hash function and the key possessed by the supplicant. In order to verify the supplicant's identity, the authenticator must possess the same key (symmetric key) and the hashed information as the supplicant in order to regenerate the hash value and verify the claimed identity of the supplicant.

Another form of hash based non-encryption identity credential uses delayed key disclosure as in TESLA. Another form of identity-based credential is a shared secret. A shared secret is not necessarily a key. Hence, it will not be used as the basis for any cryptographic operation. One example is root administrators of highly secure machines, who can prove their identity to the authenticator by creating a file in the root directory, which is an operation allowed only to the administrator. Thus, the root proves its identity without revealing the password. The secret can be a bit position or any other secret. The authenticator has to challenge the supplicant until the supplicant convinces the authenticator that it knows that secret. This authentication mechanism is called zero knowledge proof and it can be used in ad hoc networks [23].

2.5.2 Context-based Credentials

This category recognizes a unique contextual attribute of the supplicant that can be used to authenticate the latter with high confidence. Contextual based credentials can be behavioral or physical. Behavioral-based contextual credentials attempt to identify and authenticate a supplicant based on its pattern of behavior. In this scheme an authenticator would monitor the behavioral pattern of the supplicant with respect to certain functionality and classify it based on its performance. On the other hand, physical-characteristics based contextual credentials attempt to identify and authenticate a supplicant based on a physical characteristic that uniquely identifies it, such as its GPS location, RSSI (Received Signal Strength Indication) or SNR (Signal to Noise Ratio). The context related credentials depend on the context where the authentication process is performed. We divide this kind of credentials in two subclasses: behavior related and physical data related credentials.

2.6 Classification Based on Credential Establishment

The first category of authentication protocols under this classification assumes a pre-distribution at offline phase (before deployment) where creden-

tials are established. An example of that are pair-wise keys that are pre-distributed to all nodes to be used post deployment for node-to-node authentication. Pre-deployment of credentials is usually employed in symmetric-key-based protocols in SensNets. The second category of authentication protocols assumes that credentials are established post-deployment, such as protocols that rely on contextual information. The third category, like the first one, assumes pre-distribution of initial credentials. However, the actual credentials used for authentication are derived from post deployment initial credentials (see 2.5).

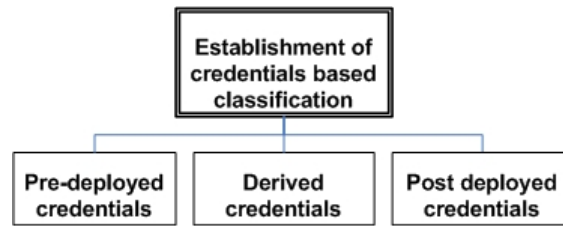


Fig. 2.5: Classification based on credentials establishments

2.7 Towards an Authentication Management Architecture

The introduction of wireless-based applications combined with their need for mobility and ubiquity introduced new challenges to conventional authentication approaches. Consequently, new and adapted authentication protocols were developed and customized to best suit the nature of these applications and their underlying networks, including ad hoc networks. As seen from our taxonomy, authentication protocols for ad-hoc networks introduced until now vary significantly with respect to their operating environment, node capability, and network configuration and functionality. An authentication protocol typically describes how the authentication operation is performed in terms of the functions of authentication as described in Section 2.1. However, none of the proposed protocols address how the authentication architecture is deployed or managed. Management of authentication is motivated primarily by the need for enhanced performance and inter-operability in today's networks. Given the dynamism of such networks, there are continual changes in the network environment in terms of time, space, and context that affect the authentication operation. Moreover, users' mobility combined with QoS and security requirements dictate the need for interaction

between the different types of autonomous networks that may be used by mobile applications. If not properly managed, the authentication operation might be rendered useless and hence might negatively impact the overall network performance and security.

Certain claims in the literature that replicating authentication server, clustering authentication process, distributing authentication process and avoiding hierarchical authentication enhance the ratio of successful authentication and the global network performance. These claims do not mention any need for authentication management in the network.

To further prove the need for the authentication management we use a demonstrative simulation study for a flat authentication server deployment model (so called server replication), which assumes that all authentication servers have knowledge of the authentication status of all nodes in the network. Among the factors that affect the performance of the authentication operation are the network traffic load, the number of authentication servers, and their placement. In Figure 2.6, we show a topology that we use to study the effect of these factors on the performance of the authentication operation. The network is a 10×10 grid of nodes in a 500×500 topography. The communication range is set such that each node has 4 neighbors, with the exception of edge nodes that have 3 neighbors, and corner nodes that have 2 neighbors. To study the effect of load in the network, we randomly generate sets of 20, 40, 60, 80, 100, 150, and 200 UDP flows. Before a flow starts, the source and destination nodes should authenticate one another through an authentication server as shown in Figure 2.7. Moreover, to study the effect of increasing the number of deployed servers, we deploy 1, 2, 3, and 4 authentication servers. Furthermore, to study the effect of placement of authentication servers, we experimented with two placement models. The first model places authentication servers in the middle of quadrants as shown in Figure 2.6. The second model places servers at the edges of the network as seen in Figure 2.6. Finally, we compare the flat deployment model used in the above simulations to a hierarchical deployment model, where the authentication status of each node is known to a single authentication server (AS).

The performance of the authentication operation is measured in terms of the delay caused by node authentication, while that of the network is measured in terms of packet loss.

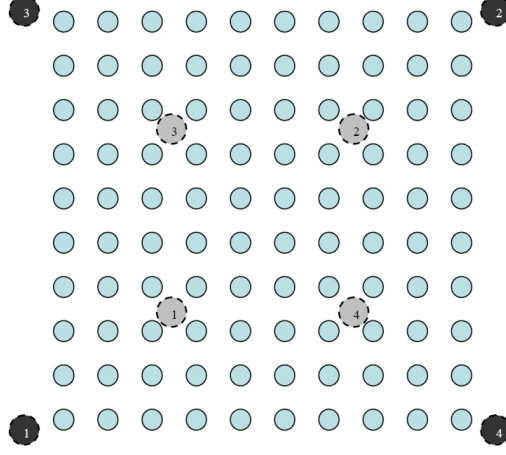


Fig. 2.6: 10×10 Grid Topology. First AS placement model is shown in gray. Second AS placement model is shown in black

2.7.1 Load Effect

Our simulation results (shown in Figure 2.7) indicate that the authentication delay increases as the load over the network increases. The results are consistent for both placement models and regardless of the number of authentication servers deployed.

Our results show that the first model of placement of authentication servers reduces the authentication delay compared to the second model as shown in Figure 2.7. However, the placement of authentication servers within the network mixes authentication traffic with regular traffic within the core of the network. This increases the contention and results in an increased packet loss as compared to the second model, which attempts to place authentication servers outside of the network to push authentication traffic outwards. These results are shown in Figure 2.7 b. This shows that there is a trade off between authentication delay and packet loss which needs to be considered when authentication servers are placed in the network.

2.7.2 Authentication Flows

While it is expected that the network performance decreases as we introduce the authentication operation into the network, our simulation results show that the packet loss decreases when authentication of nodes is mandated

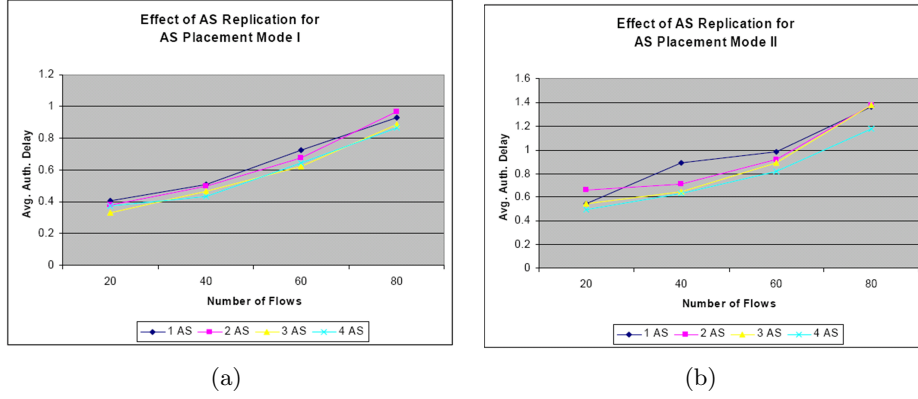


Fig. 2.7: Authentication delay as the number of flows increases from 20-80 flows for 1 to 4 AS placed according to model I (a) and model II (b). The delay of each flows is averaged over 10 simulation runs

before a flow starts. This is due to the "backoff" effect of authentication (source and destination of flows are authenticated before flows are allowed in the network). Therefore, the overhead added by authentication may be offset by the benefit of backoff. Figure 2.8 compares packet loss when authentication is mandated before a flow starts versus when no authentication is required.

2.7.3 Number of Servers

Intuitively, the objective of increasing the number of authentication servers in the network is to distribute the load over the servers, hence, to enhance the performance of the authentication operation and the performance of the network accordingly. Our simulations show that as we increase the number of authentication servers, the authentication delay is decreased when the flow is 20, 40, 60, and 80. This is expected since the replication authentication servers should distribute the authentication overhead over the servers, which is expected to positively affect the performance of the authentication operation and hence the network performance as a whole. Interestingly, at higher number of flows, these results are reversed showing an increase in delay as the number of authentication servers increases as shown in Figure 2.9. This can be explained as follows. The backoff effect of authentication decreases by increasing the number of servers. Therefore, while the increase in the number of authentication servers tend to decrease the authentication

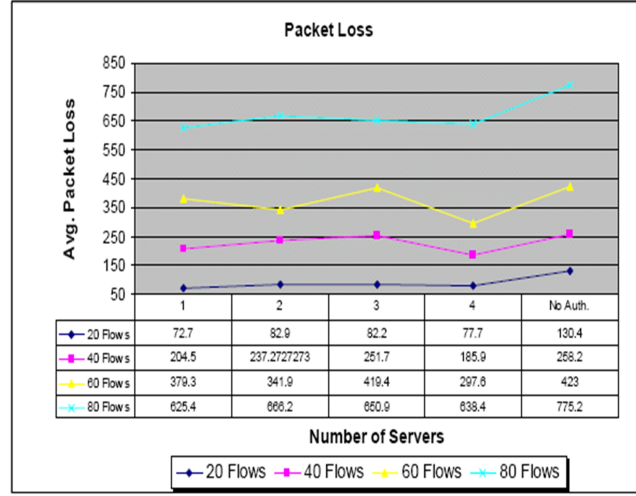


Fig. 2.8: Packet loss when no authentication required and as the number of AS increases from 1 to 4 placed according to model I. Packet loss is averaged over 10 simulation runs

delay due to load distribution, on the other hand, the load on the network increases as a result of having flows start faster. Consequently, this leads more packets in the network, which may lead to increasing the authentication delay. This is an important result indicating that the increase in the number of servers may not necessarily decrease authentication delay. System administrators need to be mindful of the different factors involved. Authentication management is therefore needed to optimize the number of active servers under different network conditions.

2.7.4 Hierarchical Deployment Model

A hierarchical deployment model is a clustering mechanism which associates knowledge of the authentication information of a node to a single authentication server rather than all authentication servers. The goal behind such deployment model is to improve the security of the network by minimizing the impact of a compromised server. A compromised authentication server in a flat deployment model exposes authentication information about all nodes in the network, while a compromised server in a hierarchical deployment model exposes only the nodes associated with that server.

On the other hand, the performance, measured in terms of authentica-

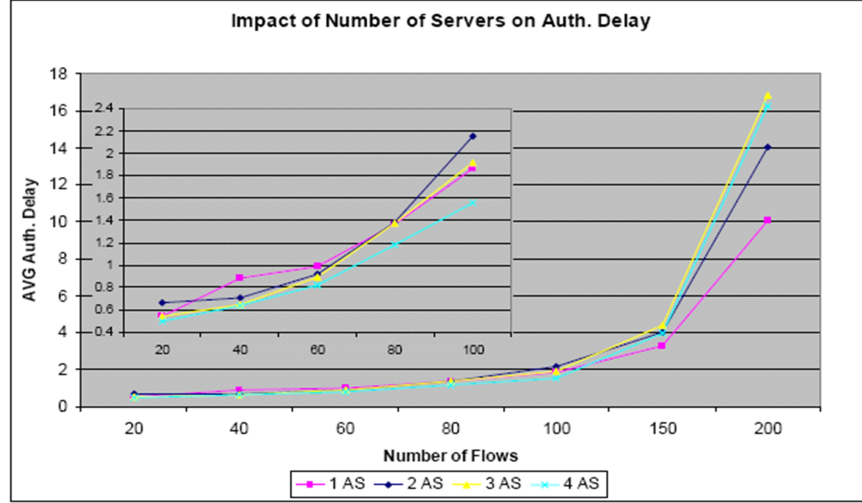


Fig. 2.9: Authentication delay as the number of flows increases from 20 to 200 flows, for 1 to 4 AS placed according to model I. The delay for 20 to 200 flows is magnified in the embedded Figure

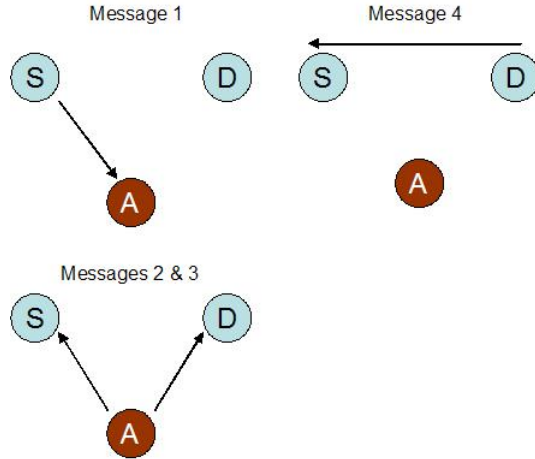


Fig. 2.10: Flat Authentication Model. "S" denotes a source node, "D" denotes a destination node, and "A" denotes an authentication server

tion delay, is decreased in a hierarchical deployment model compared to a flat model. Since the knowledge of the authentication status of a node in a hierarchical deployment model is associated with only one authentication server, the authentication model deviates from the one shown for flat deployment. A node S that is trying to authenticate to a node D will do so by having the authentication server AS whom it is associated with contact the authentication server AD with whom node D is associated as shown in Figure 2.10. This results in higher authentication delay. This indicates a tradeoff between security and performance when choosing the appropriate deployment model of an authentication infrastructure. Simulation results for hierarchical deployment model were omitted due to lack of space. Such scenarios among others motivate the need for an authentication management architecture. An authentication management architecture would overlook the authentication operation and authentication infrastructure within and across domains. Accordingly, such service would pick the appropriate deployment model of authentication infrastructure based on the security and performance requirements. Authentication management architecture would also address the deterioration in performance shown in the above example by placing authentication servers only when the need is justified and when the performance enhancement is guaranteed. Furthermore, the authentication architecture would act on clients' behalf so that functions such as roaming and handoff of clients between authentication services and across authentication domains would be carried out seamlessly.

2.8 Conclusion

We have presented a generic authentication process and developed a taxonomy of authentication protocols. The simulations, also provided in this chapter, lead to some counterintuitive results : The increase in delay as the number of authentication servers increases, for a high number of flows. This indicates that an authentication model needs to be carefully planned for a correct functioning of the authentication operation. This work focuses on developing authentication management architectures and authentication protocol for across subnetwork and domains mobility.

Our work targets also entities authentication in different wireless context mobility. We also identify the need of formal models concerning the properties of authentication protocols in wireless environments, a unified framework for the quantitative analysis of authentication protocols, and a generic architecture for the authentication management.

Related open research issues include application-aware optimization of authentication protocols and protocol survivability in presence of different attacks.

3. AUTHENTICATION ACROSS WIRELESS LANS

3.1 Introduction

Recently, both reactive and proactive schemes (as defined next section) have been proposed to reduce roaming authentication latency. In this chapter, we propose an enhanced proactive scheme and a hybrid scheme that exploits the advantages of the two approaches. We demonstrate that our schemes are highly effective in reducing authentication latency while maintaining a high level of security compared to contemporary roaming authentication schemes.

The handoff process must be fast enough to ensure a continuous connectivity, but it is influenced by several latency sources introduced at different phases of the process. As Figure 3.1 shows, the latency sources can be classified according to the phase at which they occur as follows:

- detection phase latency is the time needed to detect the need for hand-off and it generally depends on the quality of the radio link,
- search phase or probing phase latency is the time spent by the mobile station to scan radio channels in order to detect the new roaming destination *AP*.
- authentication phase latency is the time needed for the authentication process exchange.
- association phase latency is the time needed for the allocation of the service, and finally
- four-way handshake phase latency is the time taken in session key derivation.

This chapter is organized as follows. After this introduction, Section 3.2 presents the element of the system and the WLAN protocols background and a classification of different roaming authentication schemes. Section 3.3 details the proactive scheme while Section 3.4 describes the hybrid scheme.

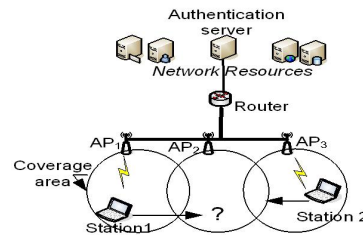


Fig. 3.2: The elements of the system

This work addresses the authentication process of mobile nodes in the same domain and in the same WLAN subnet. The elements of the system as shown in Figure 3.2 are an authentication server at the wired side with network resources, the access points (*AP*) that bridge the wireless side of the network to the wired side, and the mobile node that receives access to the network resources via multiple *AP*s due to limited coverage of each of them (25 to 200 feet).

3.2 Background

The handoff is defined as the move of a mobile station having access to the network resource from one access point AP to another and the transfer of physical layer connectivity and state information from one AP to another with respect to the considered station [47]. The causes of handoff decision can be load balancing, radio signal strength received from different AP s or any other quality of the service parameter. The handoff allows a continuous access to the network resources and mobility but must not violate the security and the trust relationship between the elements of the system. Many standards provide a framework for security support to handoff operation

without really reaching an efficient solution. In the following, we list some of them:

1. **The IEEE 802.11i security standard:** According to this standard [2], the station establishes a trust relationship with the authentication server via a strong authentication method, as the EAP-TLS (RFC 2716). The server and the *AP* have a direct trust relationship via a shared secret, but this trust relationship is a transient one between the station and the *AP* as shown in Figure 3.3. At the end of EAP-TLS full authentication, the station and the server derive a Pairwise Master Key (*PMK*) that the server forwards to the *AP* in a secure way, using the common shared secret. Only if the station and the *AP* share the same *PMK*, they can compute a common session key.

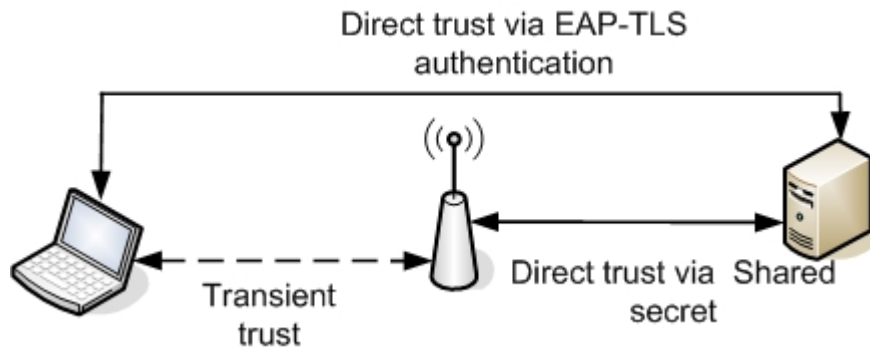


Fig. 3.3: The trust relationship in the system as defined by the IEEE 802.11i security standard

2. **RADIUS protocol (RFC2865).** Remote Authentication Dial-In User Service is commonly used to provide centralized authentication, authorization and accounting for dial-up, virtual private network, and, more recently also wireless network access. RADIUS servers share a secret with each *AP* in the network and use specific messages that were standardized in the 802.11f standard, permitting a secure inter *AP* communication.
3. **The IEEE 802.11f standard [1].** Also known as inter access point protocol (IAPP) defines *APs* communication method to facilitate a

fast and secure roaming. This standard uses a set of messages between *APs* like:

- ADD-notify: a multicast advertising packet addressed to the *APs* in the same subnet to communicate the association of a node with the advertising *AP*; this packet must be protected.
- MOVE-notify: this is a unicast message sent, in a reactive way, by the new *AP* to the old *AP* to request security information (context).
- MOVE-response: this is the answer message by the old *AP*. It includes context information related to the re-associating station like a *PMK*.
- Send-Security-Block and ACK-Security-Block: is a two-message exchange between the old *AP* and the new *AP* to setup a security association for securing the inter-access points exchange.
- CACHE-notify: this message invites the neighboring *APs* to cache proactively the security context of the mobile node. We emphasize that this message is an important part of our design.
- CACHE-response: this is an optional packet, sent as a response to the CACHE-notify to advertise that the context information is present in the cache.

Generally, the authentication schemes for roaming nodes combine the features of the previous listed standards to enhance the quality of the connectivity. We classify authentication schemes into reactive and proactive schemes with respect to *when* they derive and distribute the authentication material, namely:

The proactive authentication schemes derive and distribute the authentication material ahead of the roaming mobile node to the expected *APs*. These schemes assume that the mobile node has already performed one successful full authentication. Hence, a mobile node can skip the authentication process because its authentication material is already distributed to its roaming destination (*AP*). The proactive authentication schemes may differ at three levels: how the expected *APs* are selected for roaming, how the authentication material (the key) is derived, and last but not least, how and which network entity distributes this authentication material. We classify as proactive authentication schemes, the pre-authentication in [2], the predictive authentication in [55] and [56], the proactive key distribution in [48], as well as our own proactive scheme proposed in this chapter.

The reactive authentication schemes perform the derivation and distribution of the authentication material to the roaming destination on demand. These schemes are not necessarily independent of a previous full authentication. The reactive authentication scheme in 802.11i standard [2] performs a full authentication at each move. While the reactive keying in [48] assumes the existence of a previous full authentication and waits for the re-authentication request from the roaming nodes at the new *AP* to process the computation and the distribution of the authentication material. These schemes, the proactive ones as well as our new hybrid scheme, can differ in the derivation manner of the authentication material, in the network entity of the material distribution and how this last is processed.

3.3 Proposed Authentication Ticketing Scheme

Our authentication ticketing scheme borrows the concept of ticket from Kerberos, it is also inspired from basic Kerberos adaptation for ad hoc network in [62]. Kerberos was not used for wireless roaming support. In Kerberos the server is the median for each mutual authentication between two nodes. However, in authentication Ticketing the server plays this role only once between several entities at the same time.

Authentication ticketing scheme aims at reducing the latency of the authentication phase by pre-distributing the keys to the *APs* in the mobility pattern of the mobile node. The mobility pattern prediction is beyond the scope of this work, in fact, we assume that any mobility prediction scheme (neighbor graph, frequent handoff region, historic behavior profile) can be implemented at both *APs* and server. Our scheme seeks to reduce the authentication latency and the dependence on the server, in order to achieve a faster handoff for the mobile nodes with time sensitive applications, and to avoid the network and server load impact. We accomplish these tasks by pre-computing and pre-distributing the roaming nodes' authentication keys to the *APs* in the mobility pattern. This distribution role is assigned to the primary *AP* through which the full first authentication has been accomplished. Furthermore, the scheme aims at reducing the dependence on the server by allowing a localization of the key computation for a recurrent visiting node at the revisited *APs*. Importantly, we achieve this objective while keeping the same high level of full EAP-TLS authentication. The following subsections describe the operation steps at each unit of the wireless network.

3.3.1 At the Authentication Server:

The IEEE 802.11i standard defines the *PMK*s derivation as shown in Equation 3.1 and adopts the *PMK* caching for roaming and recurrent visits.

$$PMK = TLS_PRF(MK, clientHello.random \| serverHello.random) \quad (3.1)$$

MK being a master secret resulting from the execution of asymmetric key based challenge response protocol between the server and the mobile station. the station (client) and server random's are two random exchanged between server and authenticating station to create the pairwise master shared key *PMK*. *TLS_PRF* is the pseudo random generator function of TLS based on MD5 and SHA-1 and used as key derivation function.

We keep the Equation 3.1 of full authentication in our scheme and change the *PMK*s derivation Equation for roaming. That is:

$$PMK_i = TLS_PRF(MK, H^{i+1}(PMK) \| STA_MAC \| AP_i_MAC) \quad (3.2)$$

AP_MAC and *STA_MAC* being respectively the access point and the station MAC address, *i* is the index of the *AP* ($0 \leq i \leq n$), *n* is the number of *AP*s in the mobility pattern and *H* is a secure one way hash function. Equation 3.2 allows a recursive creation of the roaming authentications *PMK*s that are as strong as the *PMK* derived from Equation 3.1.

The pre-distribution of the *PMK*s to a roaming destination in the mobility pattern is driven through the primary *AP* ahead of the node moves. Access-accept message of RADIUS protocol achieves the full authentication and transport the *PMK*₀ to the primary *AP* and a set of tickets. The tickets include the *PMK*s for the *AP*s in the mobility pattern, the *PMK*s in the tickets should not be revealed to the primary *AP* who distributes them or to the non related *AP*s. To secure those *PMK*s we use the shared secret between each *AP* and the authentication server or a new secret derived from this one to protect what we call the ticket. Each ticket includes the *PMK* corresponding to a specific *AP* in the mobility pattern. The success-accept message has the following format:

$$(PMK_0, Ticket_1, Ticket_2, Ticket_3, \dots, Ticket_n),$$

where *n* is the number of the *AP* in the mobility pattern, and:

$$\begin{cases} Ticket_1 = (ID_{AP_1}, E_{K_1}(PMK_1, ID_{STA}, Times)) \\ Ticket_2 = (ID_{AP_2}, E_{K_2}(PMK_2, ID_{STA}, Times)) \\ \vdots \\ Ticket_n = (ID_{AP_n}, E_{K_n}(PMK_n, ID_{STA}, Times)) \end{cases} \quad (3.3)$$

E denoting an encryption function, $K_1, K_2, K_3, \dots, K_n$ being the shared keys between the server and $AP_1, AP_2, AP_3, \dots, AP_n$. $ID_{AP_i}, 0 \leq i \leq n$ represents the identity of the AP to whom the ticket is send.

However, the ID_{STA} , included in the encrypted part of the ticket, is going to inform the AP with what station to use the PMK included in the ticket. PMK_i is the Pairwise Master Key to be shared among the authentication server, the access point AP_i and the mobile station.

Finally, the field Times indicates the duration and the freshness of the ticket. The Algorithm 1 defines the PMK derivation.

Algorithm 1 The PMKs computation at the server side

STA is the mobile station, D the mobility pattern of STA, $D = \{A_i : 0 \leq i \leq n, \text{ where } A_i \text{ is one of roaming destination of STA}\}$ AP_0 is the primary Access point and $|n|$ the cardinality of $|D|$

if STA asks for authentication through AP_0 **then**
 the server computes:

$$PMK = TLS_PRF(MK, clientHello.random \| serverHello.random).$$

for all $AP_i \in D$ such that $0 \leq i \leq n$ **do**

 compute $PMK_i = TLS_PRF(MK, H^{i+1}(PMK) \| STA_MAC \| AP_i_MAC)$

 compute the tickets

$$\begin{cases} Ticket_1 = (ID_{AP_1}, E_{K_1}(PMK_1, ID_{STA}, Times)) \\ Ticket_2 = (ID_{AP_2}, E_{K_2}(PMK_2, ID_{STA}, Times)) \\ \vdots \\ Ticket_n = (ID_{AP_n}, E_{K_n}(PMK_n, ID_{STA}, Times)) \end{cases}$$

 Send all the tickets to Primary AP_0 .

end for

end if

3.3.2 At the Access Point (AP)

Our scheme uses the IAPP standardized message to transfer the mobile station authentication material from the primary AP to the AP s in the mobility pattern, in a proactive way to reduce the re-authentication latency. Two IAPP standardized messages, namely Cache-notify and Optionally Cache-response, are used to induce the AP s to cache the authentication material of the mobile station (the corresponding PMK in our case). The Cache-notify

message contains the set of tickets received from the server and it is directed to a multicast address where the *APs* are pre-registered; afterwards each *AP* picks its corresponding *PMK*. The *APs* will use the *PMK* to negotiate a session key with the proactively authenticated roaming station without going through the full authentication steps and without involving the server.

In order to keep this non dependence on the server and to alleviate the full authentication steps for recurrent visits computed in Algorithm 2, the *AP*, linked to the mobile node, increments a counter, it computes and caches the hash of the current *PMK* by detecting the imminent leaving mobile nodes. This hash will be used in the computation of the *PMK* for the next visit in the same mobile station using the following equation

$$PMK_{im} = TLS_PRF(H^m(PMK_i), Counter \| STA_MAC \| AP_i_MAC) \quad (3.4)$$

where i is the range of the *AP* and m is the number of the recurrent visit and m is the degree of the hash function. For instance, consider: $PMK_1 \rightarrow PMK_{11} \rightarrow PMK_{12} \dots \rightarrow PMK_{1m}$. With this approach, the *AP* stores one *PMK* for each station up to the first visit; one hash of *PMK* and one counter for each mobile station for any recurrent visit. In fact, if the *AP* is able to cache the *PMK* for the next visit this can alleviate the waiting time of the computation of a new one in case the *AP* receives an access request from a recurrent visiting mobile node. Algorithm 1 explains this operations.

Figure 3.4 sketches the functioning of our key distribution scheme; we have 3 *APs*, one mobile station and one full authentication *EAP – TLS*. The *AP*₁ distributes all the encrypted *PMKs* to the corresponding access point using IAPP cache notify messages.

3.3.3 At the Station

The station caches both the *PMK* of the full authentication and a chain of its hash to use the first in the computation of Equation 3.2 at each first visit, and the second in Equation 3.4 for recurrent visits. If the mobile station moves to a new *AP*, which does not have the corresponding *PMK* in its cache, it needs to perform a full authentication with a server and this new *AP* will be added to the mobility pattern. The Figure 3.4 describes the scheme exchanges.

Algorithm 2 Pairwise Master Key PMK derivation at the access point

```

if The station associates to  $AP_i$  for the first time then
    station and  $AP_i$  use  $PMK_i$  propagated by primary  $AP_0$  after the full
    authentication
else if Mobile station leaves  $AP_i$  then
     $AP_i$  hash then cache  $PMK_i$ 
    Increment a counter
else if the station comes back to  $AP_i$  then
    Station and  $AP_i$  compute  $PMK_{im} =$ 
     $TLS\_PRF(H^m(PMK_i), Counter || STA\_MAC || AP_i\_MAC)$  where  $m$ 
    is the number of the visits
else
     $AP_i$  and  $STA$  start full authentication
end if

```

We design the token authentication scheme that give more mobility management and freedom to the mobile node, as well as more dynamic role in the authentication process.

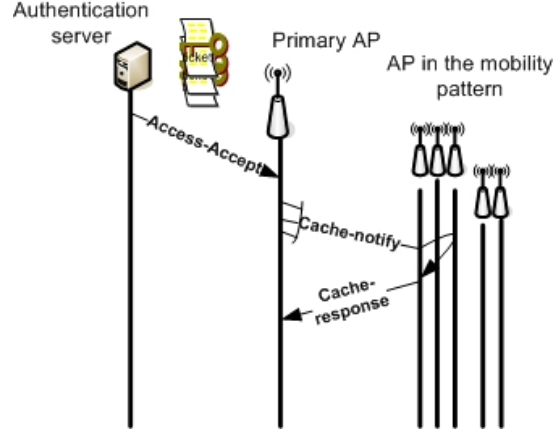


Fig. 3.4: The authentication ticketing exchanges..

3.4 The Proposed Token Authentication Scheme

This hybrid scheme, as well as the authentication ticketing scheme, aim at reducing the roaming nodes' handoff authentication latency. The mobile

station after a first full authentication receives tokens that can be used to perform a re-authentication as strong as the full *EAP-TLS* authentication to each access point in its mobility pattern. The token based scheme reduces the latency of authentication process because the mobile station replaces the authentication steps by presenting the corresponding tokens to any *AP* in its mobility pattern without interaction with the server even for recurrent visits.

3.4.1 *PMK Distribution*

The token scheme keeps secret the pairwise master key *PMK* created at the end of the classic full authentication and computes derived *PMKs* both for the primary *AP* and the *APs* in the mobility pattern. The *PMKs* of the *APs* in the mobility pattern will be enveloped in the tokens and sent by the authentication server to the mobile station through the primary *AP*. The tokens' content would be used for the re-authentication with the *APs* in the mobility pattern. This content is only readable by the corresponding *AP*. The authentication is faster since there is no interaction with the server and the exchange is limited to the *AP* and the corresponding mobile station during the re-authentication process even for recurrent visits.

3.4.2 *At the Server*

The server, as in the authentication ticketing scheme, computes and sends proactively the tokens' values to the mobile node (see Figure 3.5). The standard ACCESS-ACCEPT message at the end of the full authentication transports the tokens to the primary *AP*, which then forwards them to the mobile station. This latter uses each token with the corresponding *AP* in its mobility pattern to prove its legitimacy. The scheme assumes a centralized implementation of the mobility prediction at the server, which allows an advance computation of the specific tokens for each *AP*. It also uses either the secret shared between each *AP* and the authentication server or a derived secret from the first secret. The tokens format is presented in Equation 3.5 while the PMK_i ($0 \leq i \leq n$) derivation function uses Equation 3.2 just after the computation of the *PMK* resulting from the *EAP-TLS* full authentication of Equation 3.1:

$$\begin{cases} Token_1 = (ID_{AP_1}, E_{K_1}(PMK_1, H(K_1), ID_{STA}, Times)) \\ Token_2 = (ID_{AP_2}, E_{K_2}(PMK_2, ID_{STA}, H(K_2), Times)) \\ \vdots \\ Token_n = (ID_{AP_n}, E_{K_n}(PMK_n, ID_{STA}, H(K_n), Times)) \end{cases} \quad (3.5)$$

where n is the number of AP s in the mobility pattern, E is an encryption function, $K_1, K_2, K_3, \dots, K_n$ are the shared keys between the server and $AP_0, AP_2, AP_3, \dots, AP_n$ respectively. ID_{AP} shows the destination of the token. On the other hand, ID_{STA} informs the visited AP with which station to use the PMK included in its token. “Times” is an optional component to define the duration and freshness of the token. PMK_i is the Pairwise Master Key to be shared between the authentication server, the Access point AP_i and the mobile node.

$H(K_1), H(K_2), H(K_3), \dots, H(K_1)$ are the hash of the secret keys between each AP and the authentication server. When included in the token they allow to verify that the token comes from the server. Only the server and the access points $AP_0, AP_2, AP_3, \dots, AP_n$ can compute these hash values. Moreover, they alleviate the known clear text attacks on the token by the mobile station.

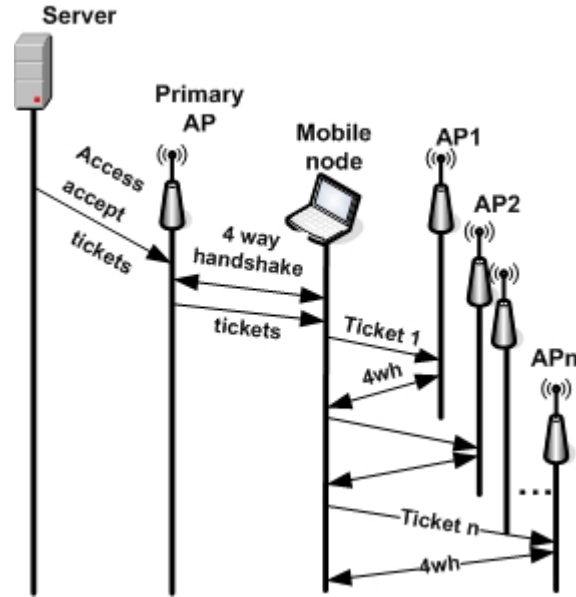


Fig. 3.5: Tokens distribution and utilization

3.4.3 At the Access Point

The primary AP receives its Pairwise Master Key PMK_0 from the server with a set of tokens as defined in Equation 3.5. The primary AP forwards

these tokens to the mobile station in a secure way after they both have establish the session key. Each AP_i in the mobility pattern, upon receiving its specific token from the roaming station, decrypts and verifies the token. Then, it establishes the session key with the mobile station using the PMK_i extracted from the token. When the mobile node leaves the AP_i versus another one, AP_i computes and caches a new PMK_{in} for the next visit using Equation 3.3. If the cache space is a limitation, the AP can cache the hash of the PMK_i for the only use of Equation 3.4, corresponding to recurrent visits. We stress the comfortable absence of a direct intervention of the authentication server at the first visit or at recurrent visits.

3.4.4 At the station

The station, upon full authentication, caches the PMK and the tokens received from the primary AP , to use them with the adequate AP in its mobility pattern. The mobile station can derive the $PMKs$ in the token form through equation 3.2 since it knows the PMK resulting from full authentication. Nonetheless, for each recurrent visits the mobile station uses Equation 3.4 to compute a new PMK , where the counter value indicates the number of visits at each AP . The station can pre-compute the hash chain of PMK to have it ready for each use in Equation 3.4. If the station moves to a new AP that is not in the mobility pattern, the costly full authentication is performed with high delay penalty, the new AP will then be added to the mobility pattern. Figure 3.6 summarizes the distribution and the utilization of tokens in the just explained scheme. In order to provide a performance comparison with our two proposed schemes (see section Analysis), in the next section we present the most relevant characteristics of existing related works.

3.5 Related Work

Previous works have addressed the issues of reducing the latency at the different phases of handoff. The analysis performed in [71] allies cache mechanisms and selective scanning of radio channel to reduce the probing phase latency. Other alternatives to reduce latency of probing operations were presented in [75] using a neighbor graph (dynamic mobility data structure) and two algorithms for channel scanning (neighbor graph algorithm and neighbor graph pruning). [47] suggests some heuristic operations to reduce the detection phase latency by shortening the beacon interval and the probing phase latency using active channel scanning and modified timer for waiting

time at each scanned channel.

Other works have focused on reducing the latency in the authentication phase for a roaming station, using different approaches that we classify as proactive and reactive with respect to when they compute and distribute the authentication material to the roaming destinations.

The proactive schemes were used first in 1995 by F. Akiildiz et al. in the context of resource allocation in ATM-based wireless networks [34] where the base station are informed by mobile station about a possible visit, so this later reserves resources for the expected mobile user to ensure high quality of service. However, the proactive schemes deal with authentication material computation and distribution, to accelerate the authentication process.

The IEEE 802.11i standard roaming authentication scheme [2] uses a pre-authentication to proactively authenticate the mobile node. The process in this scheme does not need to be combined with mobility prediction since it is triggered by the mobile node which starts a pre-authentication process with the expected *APs* before roaming in the coverage area of these *APs*. The choice of the *APs* for the pre-authentication process is based on the mobile station decision and follows both the station capability and the *APs* availability in the radio range. During the pre-authentication, the mobile node communicates with the server and the new *AP* through the old *AP* with which it is already associated to perform a full authentication with a new one (ex: EAP-TLS). If this is successful, the authentication server sends the authentication material in a unicast message to the the new *AP*. Since the mobile node can pre-authenticate with several *APs*, the scheme assumes that the *APs* can cache the authentication material until the move and re-authentication request from the mobile node; hence, the recurrent visit uses the cached authentication material or performs a new pre-authentication. This scheme makes the authentication for roaming very dependent on the mobile station.

Recently, a proactive key distribution scheme [48] has introduced the use of a centralized (at the server) dynamic data structure to describe the mobility pattern of the roaming nodes called neighbor graph (NG). This scheme includes at each position of the mobile node the *APs* that can have an association path between each other. Consequently, the server is allowed to decide the potential expected *APs* following the actual location of the mobile station. Moreover, this allows to provide *APs*, in the NG, proactively with the authentication material needed after some message exchanges with each of them. The same process is then repeated for each recursive visit. The server in this scheme assumes one previous successful full authentication and derives the authentication material using a recursive equation involving

the previously used authentication material and a master secret between the station and the server. The server then computes new authentication material for each move and provides the member of the new NG with new keys even if the NGs are overlapping. The limitation of this scheme is its strong dependence on the server.

The 802.11i reactive scheme [2] authenticates the mobile node after its motion to a new *AP* using a full authentication process (ex. EAP-TLS), like at its first authentication to the network. With this scheme the server and the mobile station create independently the same authentication material at each visit, the server then provides the *AP* with this material. Hence, the mobile node is allowed to select, independently from a server or a neighbor graph prediction, its destination of roaming after leaving its current *AP*. The reactive secure keying scheme in [48] does not use an EAP authentication method for the mobile node at the new visited *AP* nor a mobility prediction scheme, since it assumes a previous successful full authentication. The mobile node decides independently and sequentially both to which *AP* to move and to which advertise the identity (a hash) of its available authentication material (key). The selected *AP* then sends a request to the server to verify the legitimacy of the visitor; if the server recognizes the key it creates new authentication material (key) and sends it back the *AP* to be used with the mobile node. This process is repeated at each visit. The new authentication material is created simultaneously from a recursive function in the server and the mobile node, using one common secret and the last authentication material used (key). This scheme also depends strongly on the server and can lead easily to DoS attacks on the server.

3.6 Analysis

This section presents a simplistic model of the performance analysis to compare, respectively, our ticketing and token scheme to the proactive key distribution and the reactive keying scheme in [48] because of their common characteristics. We compare the two schemes with respect to two chosen extreme scenarios: A path in a straight line and a circular recurrent path. For this performance comparison, we consider the same mobility prediction scheme (NG) as defined in [48]. The NG includes at each motion position the *AP* that can be reachable with a re-association path from the current position; Figure 3.6 shows respectively (GHM), (ACH), (BD), and (CEF) that is the *AP*s included in the NG when the mobile node travel the ABCDE path. For the aim of our analysis, we define the following analysis variables.

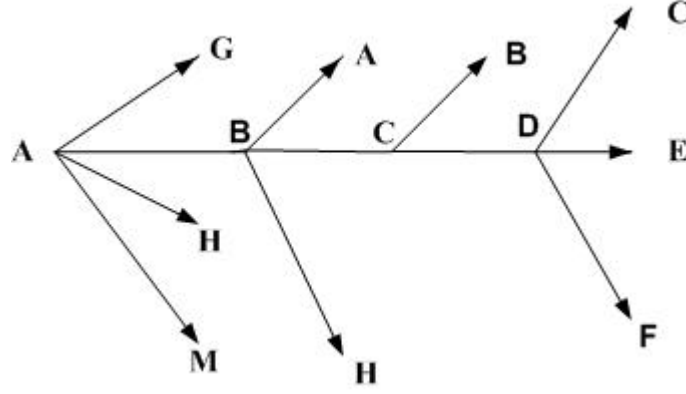


Fig. 3.6: Neighbor graph for mobile station authentication path

T_F	: the time to establish classical full authentication with EAP-TLS
$T_{encrypt}$	: the time needed to encrypt one message at the server
T_{DRV}	: the time to derive new key from the recursive equation
T'_{DRV}	: the time to derive the new key for the recurrent visit at the AP
$T_{decrypt}$	: the time needed to decrypt a message at the AP
T_{propag}	: the time needed to propagate each message from or to the server
$T_{progress}$	: the time spend at each AP before arriving to the roaming destination
$L_1 \geq 1$	: the propagation factor it depend on the load at the network
$L_2 \geq 1$	: the load factor on the server
T_{T_1}	: total time cost of AP keying in the proactive scheme of [48].
T_{T_1}	$:= L_1 * 3 * T_{propag} + (2 * T_{encrypt} + T_{DRV}) * L_2 + 2T_{decrypt}$
T_{T_2}	: total time for AP keying in the reactive scheme of [40].
T_{T_2}	$:= L_1 * (T_{DRV} + 2T_{encrypt}) + 2T_{decrypt} + 2T_{propag} * L_2$

As summarized in Table 3.1, the full authentication time in the proactive scheme of [48] is faster than our scheme but this is an expected delay for the first connection and does not affect the performance. The time cost of AP keying in the proactive scheme of [48] is equal to T_{T_1} and it depends on the server and network load; moreover the sequential relation between the keys does not allow the computation of all the keys in advance. However, the authentication ticketing needs only once, at the beginning, the cost $T_{propag} + T_{decrypt}$ for the key distribution and this occurs while the mobile node is negotiating the session key with the primary AP . For recursive visits the proactive scheme of [48] maintains the same cost T_{T_1} and all the previous

dependence and it continues rekeying the non visited AP in NG. However, our scheme has no dependence on the server and the time cost of AP rekeying is equal to zero if the cache is enabled.

Tab. 3.1: Comparison between our proactive scheme and the one in [48] during different path.

Comparison elements	One street line path		Circular recursive path	
	Proactive key distribution [48]	Authentication ticketing	Proactive key distribution [48]	Authentication ticketing
Full authentication time cost	T_f	$T_f + n * T_{DRV}$	T_f	$T_f + n * T_{DRV}$
AP keying in the roaming path	By the server	By the server	By the serve	by the primary AP
Time cost by AP	T_{T_1}	$T_{decrypt} + T_{decrypt}$	T_{T_1}	0 (If cache hit) T'_{DRV} if cache miss
Depend on node speed	Yes	No	Yes	No
Depend on network and, server load	Yes	No	Yes	No
PMK derivation	$PMK_i = F(PMK_{i-1})$	$PMK_i = F(H^{i+1}(PMK))$	$PMK_i = F(PMK_{i-1})$	$PMK_{im} = F(H^m(PMK_i))$
Wasteful rekeying	-	-	Yes	No

Table 3.2, provides the same comparison for the Token scheme. The full authentication time for the reactive scheme of [48] is faster than our token scheme, but this is an expected delay in the beginning, thus does not affect the performance of our scheme. The time cost of AP rekeying in the first scenario depends on the server and network load, so it is unambiguously higher than for the token authentication. In the circular recursive scenario, the time cost for AP rekeying as well as the dependence on the server does not change for the reactive scheme of [48]. However, the time cost of the token scheme becomes null if the cache is enabled, or is equal to $T'_{RDV} < T_{T_2}$ if the cache is not enabled. Furthermore, a small chip like a smart card with 32KB memory can store 500 tokens of 500-bit size each (ID= 64 bits, $PMK_i = 256$ -bits, $H()$ =160-bits, $Times = 8$ -bits).

3.7 Conclusion

In this work, we present two fast authentication schemes for a roaming node in the same subnetwork while keeping the same level of strong full authentication at each visited access point. We exploit the capability of dynamic mobility pattern prediction to enable the pre-computation, the pre-distribution and the update of credentials. The schemes are then compared to the proactive and reactive schemes in [48]. They show better performance in terms of latency and exhibit less dependence on the authentication server for re-authentication while roaming. The solutions proposed in this chapter

Tab. 3.2: Comparison between our hybrid scheme and the reactive one in [48] during different path.

Comparison elements	One street line path		Circular recursive path	
	Reactive key distribution [48]	Token Authentication	Reactive key distribution [48]	Token Authentication
Full authentication time cost	T_f	$T_f + n * T_{DRV}$	T_f	$T_f + n * T_{DRV}$
AP keying in the roaming path	By the server	By the server	By the serve	by the AP
Time cost by AP	T_{T_2}	$T_{decrypt}$	T_{T_2}	0 If cache hit T'_{DRV} if cache miss
dependency on node speed	Yes	No	Yes	No
Depend on network , server load	Yes	No	Yes	No
PMK derivation	$PMK_i = F(PMK_{i-1})$	$PMK_i = F(H^{i+1}(PMK))$	$PMK_i = F(PMK_{i-1})$	$PMK_{im} = F(H^m(PMK_i))$
Wasteful rekeying	-	-	No	No

are designed for authentication of mobile nodes in the same administrative wireless network. A reasonable question that may arise at this point is what happens when we have a mobile group and the mobility is across networks. In the next chapter we discuss the consequences of this case, and then we propose an adapted solution.

4. GROUP AUTHENTICATION PROTOCOL FOR MOBILE NETWORKS

4.1 Introduction

The main contribution of this chapter is proposing a new architecture for authentication management and an associated authentication protocol for mobile groups and individual nodes over heterogeneous domains. Our solution enables mobile groups as well as individual nodes to perform fast authentication. The proposed authentication protocol, mGAP, uses nodes' roaming profile to authenticate to a set of visited domains listed in the mobile node's roaming profile. Moreover, mGAP reduces the computational burden of the authentication process on the mobile nodes. To the best of our knowledge there is no scheme in the current literature that handles group authentication in mobile networks.

The balance of this chapter is organized as follows. In Section 4.2 we describe the network model. In Section 4.3 and Section 4.4, we present the details of our proposed authentication management architecture and the mobile group authentication protocol. In Section 4.6, we discuss the related work and finally we present our conclusions in section 4.7.

4.2 The Network Model

The network model consists of different domains owned by different administration authorities. Each administration authority has its own clients (subscribers), and each individual subscriber or group of subscribers have a set of allocated services. The different domains have roaming agreements with each other and can communicate in a secure way using certified public keys and signatures. The different domains can be homogeneous or heterogeneous (WLAN to WLAN, WLAN to GSM, WLAN to Ad-hoc). The main components of each domain are access points (*APs*), base station, mobile nodes (*MNs*) and mobile groups (*MGs*), and authentication servers (*ASs*). The *APs* and *ASs* can be mobile or fixed and are often provided with several network interfaces to support the mobility across domains of different

wireless technologies.

The mobile nodes can move individually or in groups of mobile nodes. The mobile group is a set of mobile units acting as a team to perform some tasks in a collaborative way. Examples of mobile groups include:

- Digital battlefield communications (navy group, marines groups, etc);
- Wireless video game between mobile teams;
- Search and rescue situations where different services collaborate (national guard, police, paramedics and fire fighters) to perform specified tasks.

The mobile nodes communicate with each other in “ad hoc mode” and in “infrastructure mode” mode with the AP or base station.

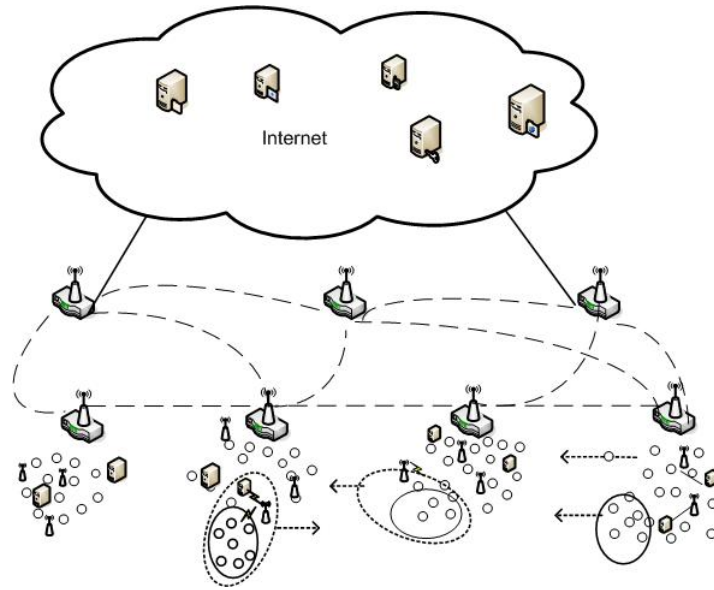


Fig. 4.1: Mobile network model

There are two kinds of mobile groups, *open* and *closed*. The closed group is easier to manage because the number of the members of the group stay constant, so the mobile access point ensures access to the network resources for the same group members in the home domain as well as in the visited domain.

The open group is more complicated to manage because the number of the group members changes according to the group and individual node operations as well as the network conditions. The main operations within the

open mobile group are leaving and/or joining of nodes, and group merging and splitting. Members of a mobile group can leave because of mobility, reachability problems, or access revocation.

4.3 Authentication Management Architecture

Our architecture, as presented in Figure 4.2 and 4.3, is based on the capacity of the home domain to analyze the mobile entity's behavior and information registered in the domain database, and to predict the authentication needs according to the home domain (*HD*) policy. In this section we describe the modules that constitute the architecture (see Figure 4.2). The idea of introducing profile management aims to provide mobile entities with appropriate *single* credentials for authentication in a set of visited domains.

4.3.1 Architecture Components

The components of the authentication architecture are presented in Figure 4.2 and are detailed as follows

1) **Mobility database module** contains information about the mobile entities (nodes, groups, mobile APs and ASs). Information such as entity characteristics, allowed services, subscription details, billing, personal information, and roles...etc, can be represented in terms of quantities in the registered subscription contract $SUB = \{x_1, x_2, x_3, \dots, x_n\}$ of each mobile entity, where n is the number of information items. The mobility database includes also an initial list of domains for which all the registered mobile entities can roam (as initial service) before the end of the profiling phase. This part of initial service can be included in the domain policy.

2) **Profiling module** is divided in two parts: behavior data collector and behavior analyzer.

a. Behavior data collector observes the behavior of mobile entities to define an accurate profile. After the first subscription at the home domain (the first connection to the network), the mobile entity's behavior data are collected for a predefined period of learning to construct the profile data. The collected information includes the frequent handoff destinations and rates, the frequent geographic mobility areas of action, the client requirements in terms of bandwidth, QoS, monetary cost preferences and last but not least the security requirements.

Example: Let A_{HD} be the home domain and A the set of domains that have roaming agreement with A_{HD} where A is expressed as $A = \{A_0, A_1, \dots, A_n\}$. Let M be the mobile node that has A_{HD} as home domain. While M roams

across networks, the behavior data collector collects the following information about M roaming behavior, that is:

M visits the set of domain $A_V \subseteq A$. All the domains $A_i \in A_V$ share a specific geographic area. All the domains $A_i \in A_V$ are able to provide M with the services for which M is registered at the home domain A_{HD} . The monetary cost of the services requested by M at the visited domains must coincide with the monetary preferences of M . All the domains $A_i \in A_V$ ensure QoS equal to or greater than that requested by M .

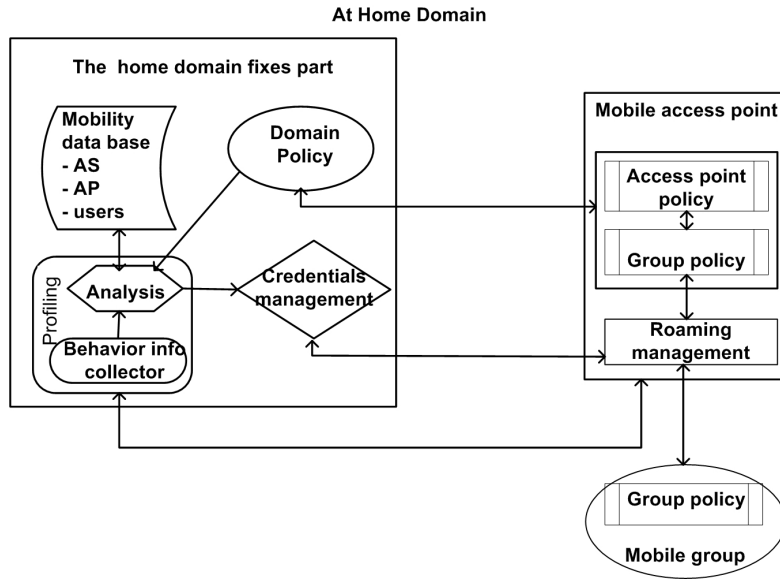


Fig. 4.2: Elements of authentication architecture with mobile access point

b. Behavior analyzer module combines information from the mobility database, the behavior data collector and the domain policy module (we detail in the next session) to define the profile and classify the mobile nodes in specific categories corresponding to their specific roaming authentication needs (by default their capacity is limited). The classification is necessary given the large number of other domains which have roaming agreements with the home domain. In some cases, where the roles and the tasks as well as the roaming right are known a priori, (e.g. national guard hierarchy), the categories are less dynamic and the profiling is easier.

Example: Let M be mobile node registered at mobility database of its home domain for the services: Web browsing, phone calling, SMS, MMS and initial

minimum roaming destination. During the learning phase, the data collector collects the information about M requests of services, geographic area of action, the QoS requirements, bandwidth requirements, security requirements, etc. In the home domain policy we can find for example the two rules: the domains without roaming agreement with the home domain can not be a roaming destination. All domains that use security protocol supporting only 256 bit key size cannot be a destination of roaming. Based on this information, the analyzer generates an authentication profile not necessarily limited to the geographic area of the learning phase (for more roaming freedom). The profile contains a list of allowed roaming destination larger than the set discovered in the learning phase (enlarged A_V), which matches also with all the requirements of M and respects the home domain policy conditions. This profile when received by the credential manager will be the basis for the creation of roaming credentials. Finally the output of the profiling module processes will be the largest possible list of domains that matches with data coming from the other modules and operations. When communicated to the credential manager, this later will create specific roaming credentials.

3) **The policy module** of each domain manages memberships, services allocation, roaming right. The policy, determined by this module, follows the market conditions and the commercial agreements and priorities. The home domain (HD) policy is the set of rules adopted by each domain to regulate for example the conditions on roaming right, the level of security tolerated, the thresholds of service allowed to the registered entities while roaming etc. This can be expressed as follows:

Let P_{HD} the home domain policy expressed in this way

$$P_{HD} = \{p_1, p_2, p_3 \dots p_n\}$$

where $p_i \in P_{HD}$ is a domain elements of policy. If A_V is the list of expected future roaming destination, so each $A_i \subseteq A_V$ must have requirements, conditions and characteristics of services that match not only with M requirements and needs but also with the rules defined in the home domain policy P_{HD} .

4) **The credential management module** uses information received from the profiling module, to create the authentication credential corresponding to the entity profile, in such a way that each credential K_M delivered by the HD allows authenticated roaming across the domains of the profile.

Example: Assume a mobile entity M have a roaming right across the set of domains A_V ; so each $A_i \in A_V$ can authenticate M , with the transportable credential K_M obtained from the credential manager.

5) **The mobile access point policy (P_{AP}) module** is part of the home domain policy, thus it is obvious that AP respects its rules and conditions. However, the AP needs its local subpolicy P_{AP} which must be in accordance with HD policy P_{HD} . For example, this policy has to manage individual mobile node as well as static and dynamic groups.

One example of P_{AP} elements is a condition on the bandwidth where the AP limits the roaming right of its mobile nodes and groups toward visited domains. Thus, if the conditional bandwidth of an AP is 500Kb/s, the AP prohibit to its groups or nodes roaming toward domains with lower bandwidth. Such policy ensures load balancing over the network and enhances QoS.

6) **The roaming management module** stores and manages in the mobile entities the roaming credential received from the credential manager. This module is in charge of the re-authentication process during a visit in a new domain. Importantly, the roaming credential is updated for each re-authentication either by the home domain or by the last visited domain. When the mobile access point supports a static mobile group, they are bind together, therefor the authentication becomes transparent to the mobile group meaning that only the mobile AP re-authenticates.

When the mobile group is dynamic, because of departure and joining of members, the group can be partitioned to subgroups, or a whole group can merge with other local groups. Our architecture and protocol addresses exactly the problem of re-authentication of these inter or intra movements. How this is performed, is the subject of section 4.4.3. Now we provide an example of the architecture flow.

4.3.2 Example of the Architecture

Let the AP_m be a mobile node, registered at the home domain HD , subjected to the policy and conditions of this HD . The AP_m has, like all registered nodes in HD , a subscription contract that specifies its obligations and rights like its services, specific information about its identity, address, technical characteristics, the initial list of the allowed roaming destination and the group served by the AP_m , if there is any etc. We can express the AP_m subscription contract as follows: {services: (SMS, WEB browsing, phone calling, the group served by AP_m), ID = MAC-address, number of radio interfaces, geographic address, HD identity, the initial allowed list of roaming, signed public key of AP_m ...}. The steps to get the credentials for this AP_m are:

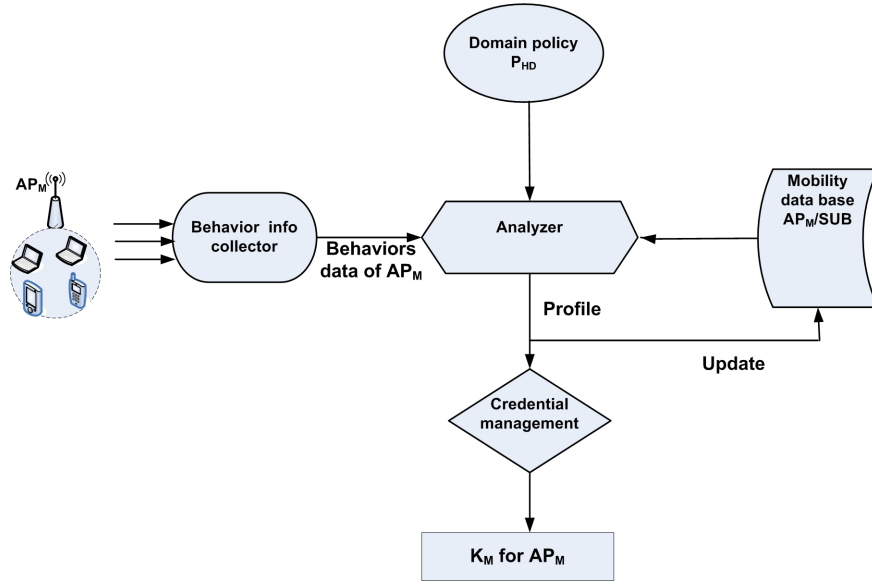


Fig. 4.3: Input and output of the home domain architecture elements

1. All the information is stored in the *HD* at the Mobility Database module.
2. During the learning phase, the behavior information collector observes:
 - (a) The frequent requests of the AP_m like services, handoff, mobility destinations,
 - (b) The list of frequent domains visited by AP_m ,
 - (c) The collected common characteristics of the domains that interested the access point AP_m , like {all domains with: Security 64 bits $\geq$ key $\leq$ 128 bits, 256 bits/s $< BW \leq 2Mb/s$, $QoS > 0.5$, cost $\leq$ average (fixed by *HD* or AP_m)} will feed the analyzer with other data to establish the profile.
3. The *HD* policy is a set of rules regulating the roles and rights of the registered entity and the relation with others domains, the inter-domains billing system, inter-domains security and so on. The panel of services and rights that *HD* can allocate are very large but they are not allocated to all registered entities. As noticed before, the services allocated to each entity are tailored depending on their needs, registration (contract) type, and mobile entities capacities. The mobile

entity AP_m policy is a subgroup of the HD general policy. Concretely, the HD policy may include p_1, p_2, p_3, p_4 policy elements that are defined as follows: p_1 allows roaming only to the domains certified by government, p_2 allows roaming only to domains with equivalent access control; p_3 allows roaming to domains with height Bandwidth until 54 MB; p_4 allows roaming to domains with different tariffs of services. Then the AP_m subpolicy can include p_1, p_2 , but neither p_3 nor p_4 because of the limitation of its capacity.

4. Subsequently, the information about AP_m behavior is transferred to the analyzer module in the architecture, which combines it with the information stored in the mobility data base module and the HD policy in order to address an accurate profile.
5. At that point, the credential manager receives the profile from the analyzer and tailors the roaming credential for the AP_m according to its profile.

Consider now a practical numerical example that contains the data combination and accumulation in each of the above steps with the purpose of establishing the profile of the mobile entity:

1. subscription : services (SMS,WEB, phone calling, the group served by AP_m), ID= MAC-address: 00-06-5D-73-3B-6D, N8 of radio interfaces:2, (WLAN 802.11g, GSM CDMA), Geographic address= GPS:(X:241200, m Y: 5435700 m), HD Identity = Belgacom, Signed public key of $AP_m = PK_{AP_m}, s_{Belgacom}(PK_{AP_m}, HD \text{ Identity}, AP_m \text{ Identity})$
2. Behavior information collector gathers the following information margin of security $64bits < key < 128bits$, Bandwidth requested $256bit/s < BW < 2Mb/s$, $QoS > 0.5$, $cost < average$ (fixed by HD or AP_m).
3. Only the domains with p_1, p_2 as elements of their policy represents an interest for AP_m , because the others (p_3 and p_4) are very powerful. Even if available at the visited domain they are not a base of selection, because AP_m has no capacity to use them.

These steps lead to the identification of the set of domains that this entity has the right to visit. Hence, the result is the profile defined as the list of domains, A_V , satisfying all the above conditions dictated by the set of information in possess of the analyzer. A_V can be the following list A_V =the initial allowed list of domains, Singular, VT, T-mobile, Orange.

Once the profile is established, it will be stored in the mobility database for further uses, but it can also be updated by the analyzer in case of a contract and/or a behavior change, or in case a new domain enters the A_V . At the expiration date, the credential can be renewed upon the stored profile without restarting all the profiling process.

The credential manager will combine the information in (1), (2) and (3) to tailor a credential that allows AP_m to roam to all the list $A_V = \{\text{the initial allowed list of domains, Singular, VT, T-mobile, Orange}\}$.

4.4 The Authentication Protocol

The protocol mGAP treats the mobile access point both as a support for the mobile group and as an individual node. The support of the mobile access point leads to faster authentication to be implemented, namely, the mobile access point re-authenticated at the visited domain will avoid individual re-authentication of each node in the roaming group. The profiling step would allow the credential manager to provide the mobile AP with credentials that allow it to be authenticated with any domain listed in the profile. On the other hand, if any node or the entire group moves to the new access point in the visited domain, the authentication is performed by proxy to the mobile access point. The same process of proxy authentication is valid for a local individual mobile node or group in order to use a mobile access point.

The roaming credential design is based on the cryptographic primitive described in the extension of [37] which allows any member of targeted t-member subset of an ad hoc group of n receivers to recover the message. The notation of this scheme is $(1, t, n)$ where 1 means that one receiver decrypts the message, 't' represents the number of targeted members from n , where n is the total number of group members formed by the sender (see subsection below).

4.4.1 Secure Custodian Hiding Verifiable Encryption Scheme

Custodian-hiding verifiable encryption paradigm described in [37] allows a sender to verifiably encrypt a message to a group of receivers in a way that only one of them (optionally; any of them, a set of them) is able to decrypt it. In addition, any public verifier can ascertain this fact while he knows nothing about the plain text and can not compute the identity of the actual decryptor.

Let (E_i, D_i) , $1 \leq i \leq n$, be secure public-key encryption and decryption functions generated by G . Let π index the targeted receiver. Let p, q be large

primes, $q \mid p - 1$, and $g \in F_p$, order $(g) = q$. Let the security parameter λ_s be as large as $|q|$. Let f be defined by $x \rightarrow g^x$ which is an instantiation of one way group homomorphism from Z_q to $\langle g \rangle$. Let $m \in Z_q$ be a message. Let N be the number of cut-and-choose rounds. Let $H_1 : \{0, 1\}^{\lambda_s}$ and $H_2 : \{0, 1\}^* \rightarrow Z_q$ be some statically independent and cryptographically strong hash functions.

When the prover P computes any probabilistic public-key encryption function, P needs to send the corresponding coin flip sequence to the verifier V and the sequence is to be carried on wherever the original message goes. The scheme does not explicitly specify such in the following.

$Sym(n)$ denotes the symmetric group of order n . It consists of all permutations on n objects. The scheme instantiates a one-way homomorphic mapping of m , typically used in verifiable encryption literature, by discrete exponentiation g^m , and assumes the discrete logarithm problem to be secure.

The protocol between P and V (encryption)

1. P computes $d = g^m \bmod p$ and send d to V .
2. repeat the following steps N times in parallel.
 - (a) (commitment) P randomly picks $s \in_R Z_q, r_i \in_R \{0, 1\}^{\lambda_s}$ for $1 \leq i \leq n, \phi \in_R Sym\{n\}$. P computes

$$\lambda = E_1(r_1) \parallel \dots \parallel E_n(r_n)$$

$$\gamma = (g^{H_2(r_{\phi(1)})} \bmod p, \dots, g^{H_n(r_{\phi(n)})} \bmod p)$$

$$\alpha' = E_1(s) \parallel \dots \parallel E_n(s)$$

$$\alpha = H_1(\alpha')$$

$$\beta = g^{H_2(r_\pi)s} \bmod p$$

$$\theta = H_1(\lambda \parallel \gamma \parallel \alpha \parallel \beta)$$
 P sends θ to V .
 - (b) (Challenge) V picks $b \in_R \{1, 2, 3\}$ and sends b to P .
 - (c) (Response)
 - Case $b = 1$, P sends $r_1, \dots, r_n, \gamma, \alpha$, and β to V
 - Case $b = 2$, P sends λ, γ, α , and s to V .
 - Case $b = 3$, P sends λ, γ , and α' , and $s' = H_2(r_\pi)s + m \bmod q$ to V .
 - (d) (Verification by V)
 - Case $b = 1$:
 - Verify that $r_1, \dots, r_n$ are distinct.

- Verify that there exists a unique permutation $\delta \in \text{Sym}(n)$ such that $\gamma = (g^{H_2(r_{\delta(1)})} \bmod p, \dots, g^{H_2(r_{\delta(n)})} \bmod p)$
 - Verify that $\theta = H_1(\hat{\lambda} \parallel \gamma \parallel \alpha \parallel \beta)$ where $\hat{\lambda} = E_1(r_1) \parallel \dots \parallel E_n(r_n)$. Continue only if all verifications succeed.
 - Case $b = 2$:
 - Denote $\gamma = (\gamma_1, \dots, \gamma_n)$.
 - Compute $\tilde{\alpha} = H_1(E_1(s) \parallel \dots \parallel E_n(s))$ and $\beta_i = \gamma_i^s \bmod p$, for $i = 1, \dots, n$.
 - Verify that $\theta = H_1(\lambda \parallel \gamma \parallel \tilde{\alpha} \parallel \beta_i)$ for exactly one index $i \in \{1, \dots, n\}$. Continue only if the verification succeeds.
 - Case $b = 3$
 - Compute $\beta' = g^{s'} / d \bmod p$
 - Verify that $\theta = H_1(\lambda \parallel \gamma \parallel H_1(\alpha') \parallel \beta')$. Continue only if the verification succeeds.
3. (Output) V terminates if any verification fails in any of the N cut-and-choose Rounds. Otherwise, it outputs d and the four-tuple sequences $(\alpha', \lambda, \beta', s')$ for all case- $(b = 3)$ rounds to all n receivers as the ciphertext, C .

Recovery Algorithm R. Denote $\bar{\lambda}_1 \parallel \dots \parallel \bar{\lambda}_n = \lambda$ and $\bar{\alpha}'_1 \parallel \dots \parallel \bar{\alpha}'_n = \alpha'$. For d and Each four-tuple sequence $(\alpha', \lambda, \beta', s')$, each receiver i , $1 \leq i \leq n$, independently performs the following steps.

1. Compute $r_i = E_i^{-1}(\bar{\lambda}_i)$ and $s = \bar{\alpha}'_i$.
2. Compute $m' = s' - H_2(r_i)s \bmod q$.
3. Verify that $g^{s'} = g^{m'} \beta' \bmod p$. If the verification succeeds, then receiver i is the targeted decipher and it outputs the decrypted message m' and halts. Otherwise, the receiver repeats the steps for another four-tuple sequence.

The $(1, t, n)$ scheme extension:

The $(1, t, n)$ scheme allows any one in a targeted set of t receivers to recover the encrypted message, and then the receivers are anonymized.

Encryption. Let $\pi_1, \dots, \pi_t$ be the index of t targeted receivers. the encryption algorithm is similar to before, but with small modification:

1. P also send t to V before commitment.
2. (commitment) Same as before except

$$\beta = g^{H_2(r_{\pi_1})s} \parallel \dots \parallel g^{H_2(r_{\pi_t})s} \bmod p$$

3. (Response) Same as before except in case $b = 3$, replace the original s' with $s'_i = H_2(r_{\pi_i})s + m \bmod q$ for $i = 1, \dots, t$.

4.(verification) Same as before except in

(a) Case $b = 2$, verify that

$$\theta = H_1(\lambda \parallel \gamma \parallel \tilde{\alpha} \parallel (\beta_{i_1} \parallel \dots \parallel \beta_{i_t}))$$

For a unique t -member ordered tuples $\{i_1, \dots, i_t\} \subset \{1, \dots, n\}$.

$$\beta' = g^{s'_1/d} \bmod p \parallel \dots \parallel g^{s'_t} \bmod p$$

5. (Output) Same as before except replacing the original s' with $s'_1, \dots, s'_t$

Decryption - Denote $\bar{\beta}'_1 \parallel \dots \parallel \bar{\beta}'_t = \beta'$.

- Step 2 is modified as: receiver i computes $m'_{i,j} = s'_j - H_2(r_i)s \bmod q$ for $j = 1, \dots, t$.

- Step 3 is modified as: receiver i checks if $g^{s'_j} \stackrel{?}{=} g^{m'_{i,j}} \bar{\beta}'_j \bmod p$ for $j = 1, \dots, t$. If one of them equal, then receiver i is one of the targeted decipherers.

Remark: It was proved that if for encryption length of 1024 bits, the communication bandwidth is about 4096 bits for each round. If there are 100 rounds, the total bandwidth is about $50n$ kbytes and the size of cipher text is about $17n$ kbytes, where n is the number of custodians.

4.4.2 The group authentication protocol

Our protocol may not use all characteristics of the previous scheme like the verifier role (it can be done by the home network). The only the final encrypted message for the group is used, however the mobile entity could play the role of verifier if its the computation capabilities allows such burden.

Our protocol is inspired by basic Kerberos [74], however we provide the mobile entity with transportable credentials that are recognizable by all the visited domains listed in the profile, and used by them for mutual authentication with the mobile entity. This credential allows firstly the establishment of authentic session keys, and secondly the authentication of the home domain by the visited one, and finally it avoids expensive storage and computations to the mobile entity. The protocol exchanges and the protocol notation are explained below (see also Figure 4.4).

1. $U \longrightarrow HD : U, N_A$
2. $HD \longrightarrow U : Ticket, E_{K_{HD-U}}(K, N_A, SUB)$

3. $V \longrightarrow U : N_V$
4. $U \longrightarrow V : Ticket, authenticator, s_U(H^n(K_{Ro}, N_V), V)$
5. $V \longrightarrow U : E_K(N_U)$

Tab. 4.1: The protocol notation

U, HD, V	mobile entity, home domain, visited domain identities
SUB	subscription contract excludes: validity time, authorized services, (U) , (HD) , S_U
S_U, s_U	mobile entity verification and signature public key pair
S_{HD} and s_{HD}	home domain signature/verification public keys pair
N_A, N_U, N_V	user nonce's, visited domain nonce's
K_{HD-U}	shared key between HD and mobile entity
K	roaming authentication key delivered by HD
K_{Ro}	session key to use by the visited domain and mobile entity
H	a hash function
$PKG(1, t, n)$	public key to encrypt messages group of domains t from n
Ticket	$E_{PKG}(K, SUB, s_{HD}(K, SUB))$
The authenticator	$E_K(SUB, K_{Ro}, N_U, N_V)$

As explained before, the mobile entity U at the registration time is initialized by the home domain HD with both the shared symmetric key K_{HD-U} and the contracted subscription SUB . We describe here the mGAP authentication protocol in terms of the generic authentication model presented in previous chapters [53]. A presentation of the protocol messages exchanged can be found in Figure 4.4.

Bootstrapping: To get bootstrapped with roaming credential, the mobile entity U generates a nonce N_A and send its identity U with N_A to the home domain HD in message (1). When HD receives message (1), generates a roaming authentication key K , uses (K_{HD-U}) to encrypt K , the received nonce N_A and the subscription contract SUB of the mobile entity U . HD also creates a ticket those content is encrypted with the encryption public key $PKG(1, t, n)$. The elements contained in the ticket are K ,

the mobile entity subscription contract SUB , the signature of the home domain $s_{HD}(SUB, K)$ over both the contract SUB and the key K . The bootstrapping step is finished when the U receives message (2) from HD . Thus U receives securely the authentication credential for secure across domains roaming. U decrypts the second element of message (2) using the shared key K_{HD-U} then recovers K , N_A , and SUB . U verifies that the N_A matches with the one sent in (1).

Pre-authentication: starts when the mobile entity U arrives at the visited domain V , and receives for the latter the nonce N_V and challenges U with message (3). Consequently, U generates a session key K_{Ro} and a nonce N_U . U uses K to encrypt K_{Ro} , N_U and SUB to create what we call the authenticator. U computes also the signature $s_U(H^n(K_{Ro}, N_V), V)$ and sends the latter with the authenticator as well as the ticket received in message (2). The pre-authentication step finishes when the visited domain V receives the message (4).

Credential establishment: starts when the visited domain V treats the message (4), where it decrypts the ticket, recovers K , decrypts the authenticator, recovers N_U , N_V , SUB and K_{Ro} . V verifies also that the N_V matches the one it has sent in message (2); then it establishes K_{Ro} as a session key. Therefore, K is established as authentication key and K_{Ro} as session key. This finishes the credential establishment and the one way authentication U to V . Later, the session key K_{Ro} can be used among other elements to encrypt a K' the future authentication roaming key for U to allow this latter to authenticate to another domains in the profile.

Authentication: When the mobile entity U receives message (5), she knows that she is authenticated to V and can authenticates V . Message (5) confirms to U the authentication key K .

Monitor behavior: The visited domain V monitor the validity of the mobile entity credential and subscription contract SUB . If the credential and the contract are still valid the authenticated state will be kept. However, if the credential and the contract is expiring soon, a new authentication process is started from the credential establishment step. On the other hand, if the visited domain V uses a misbehavior detection scheme like the watch dog, the misbehaved mobile entity can be revoked and isolated in a black list.

4.4.3 Mobile Group Operations

One of our aims is the introduction of the mobile access point as a support for mobile groups operations. This leads to less costly and faster authenti-

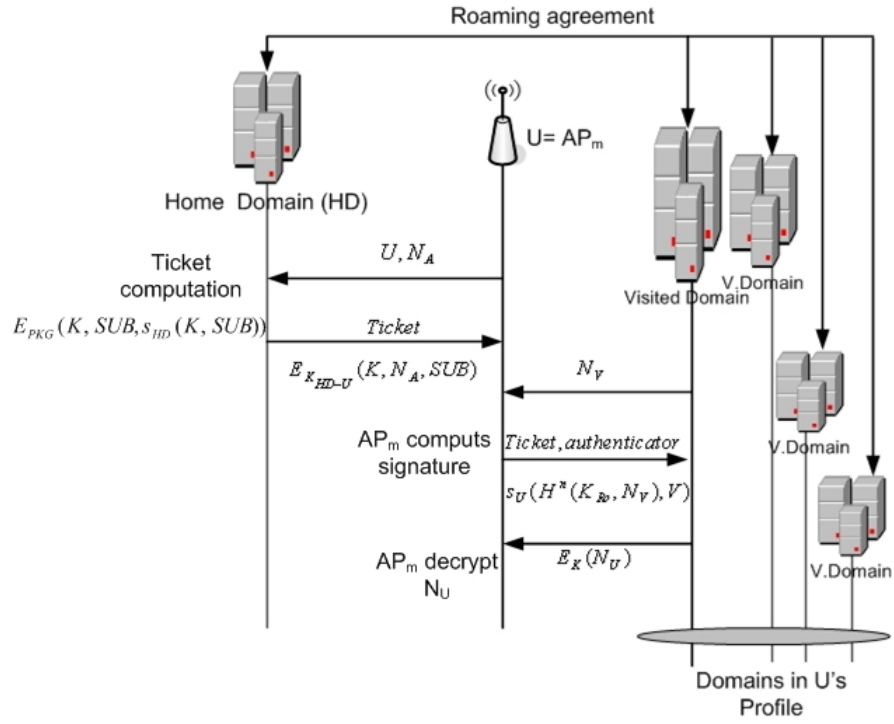


Fig. 4.4: The protocol exchanged messages

cation that relies on specific credential supplied by our protocol (see next section). In the general case, a mobile access point cannot roam to a new domain that is not allowed by the home domain, because of the lack of roaming agreements or because of a mismatch in service qualities. The authentication to a non tolerated domain is going to fail because the credential will not be recognized. By contrast, the authentication to an *allowed* domain, which is not in the profile of the mobile entity, is possible through a proxy authentication at the first visit. Afterward, the profile will be updated to incorporate the new visited domain and to allow local future authentication.

As already introduced, the mobile groups can perform and undergo many operations while roaming across domains. The most frequent operations are node leaving and joining the group, group splitting and group merging. Our architecture manages these operations in an elegant way:

Node joining/leaving the group. Each mobile node, member of a group, shares security association with its access point. In the eventuality that this mobile node leaves its mobile access point toward a different one, with which it does not share security association, its authentication is performed by proxy to its original mobile access point. But this happens locally at the visited domain. In this way, the authentication can be performed locally through the communication between the access points which make the process less costly in term of latency and computation, unlike other architectures where the authentication request by proxy travels across domains and undergoes a series of encryption-decryption. Analogously, local authentication through visiting mobile access point is allowed for mobile nodes coming from local access points.

Mobile group splitting/merging: The group can split for several reasons, like operation needs in the mobile network. In this case, the subgroup can access the network services through another local access point. The authentication of the subgroup can be performed in the same way as for the individual mobile node, that is by proxy to the mobile access point roaming with the subgroup to the visited domain. The authentication of the members of the subgroup is performed one by one, by proxy to its mobile access point. The same authentication process is used for local access points to the mobile access point.

Hence, when the mobile group, its subgroups or one of its affiliated nodes, roam to a local access point in the visited domain, the authentication can be performed by proxy to the mobile access point.

4.5 mGAP Security and Performance Analysis

To the best of our knowledge, this work is the first proposing a group encryption scheme for transportable authentication credential creation, where the mobile AP provides support for group mobility in the context of mobile networks. This protocol can also be used for individual mobile nodes as well as for other mobile entities such as mobile access points.

In this section we provide the analysis of the security and performance of mGAP.

4.5.1 Security Analysis

Consider a threat model where an attacker can be active or passive. Thus it can attempt to eavesdrop on exchanges, to inject messages, to alter the exchanges, or to replay old messages. We assume that all domains have or can easily obtain the authentic public key of the different domains. This is a reasonable assumption since the number of domains is much smaller than the number of mobile entities. In what follows, we present the security aspect of each message and message components.

- mGAP provides the mobile entity (AP , mobile node) with a fresh roaming authentication key ' K ' via a secure message (encrypted using the long term shared key K_{HD-U}) in message (2). An attacker can neither recover K without knowing K_{HD-U} , nor it can use an old key since the roaming authentication key K is encrypted and sent with the nonce N_A received by HD in message (1). The presence of N_A in the non ticket part of message (2) proves the freshness of K for the mobile entity U .
- Message (2) also includes a ticket, which is the transportable credential to authenticate the mobile entity to the visited domain and vice versa. This ticket is encrypted using the group's public key based encryption algorithm. The ticket also includes K and SUB as well as their signature by HD ; therefore an attacker (non member) of the group cannot recover K .
- The signature of HD over (K, SUB) provides V with an authenticated key. The signature of HD ensures the non-repudiation of roaming credential by HD to V ; moreover the signature of the SUB by HD certifies the validity time, the allowed services, the roaming entity identity, and U signature verification key.
- Once message (4) is received, V decrypts the ticket to recover K and decrypts the authenticator. Only a member of the group can decrypt the ticket and the authenticator. The presence of N_V sent to U in message (3) within the authenticator proves the freshness of the authenticator and

proves that U knows the key K ; therefore an attacker cannot impersonate U to V , or even replay an old authenticator.

- The signature part in message (4) proves again the identity of U and that U requests the access to the nominated domain in the signature. An attacker cannot impersonate U to create a fresh signature.
- The freshness of the signature is ensured by the presence of N_V . The fresh signature of U includes the chosen domain's identity. This shows from which domain the service was requested and allows for protection against impersonation and rogue domain. However, the domains of the profile that have not been chosen for the present transaction are trusted to not impersonate each other to the mobile entity. The n^{th} hash included in the signature allows the mobile entity to request different services or different parts of services. This by giving an earlier signed hash value from the given hash chain instead of re-authentication. This practice is often used in micro-payment schemes [13].
- Message (4) confirms the key K and authenticates V to U . For future roaming, V and U repeat exchanges (1) and (2), with V using the new nonce N'_A and new ticket. V also uses K_{Ro} to provide U with a new authentication roaming key K' . Note that K , K_{Ro} as well as K_{HD-U} are used for limited time and specific operations; so their corruption by an attacker is impossible in practice, since the size of each of them is at least 128 bits.
- Our protocol ensures forward secrecy because if one of the roaming authentication keys is compromised, the previous sessions are not compromised.

4.5.2 Performance Analysis

Our protocol allows the mobile entity and mobile group to roam freely across different administrative domains using one credential for mutual authentication and key establishment. This protocol does not require high capacity for storage or pre-computation before roaming to a new destination. The mobile entities do not need to decide in advance the destination of roaming as long as they roam within their roaming profile and tolerate extra-profile roaming. Our protocol shifts most of the computation burden on the domain side. The mobile entity performs one symmetric encryption, one symmetric decryption, n hashes for each roaming destination, and finally one low cost signature [32]. Our protocol uses three message exchanges for the mutual authentication process and key establishment (messages 3, 4 and 5).

The assumption of trust among the domains of the profile is reasonable since each domain is home domain for certain nodes and visited domain for others. If we do not trust the domains in the profile to not impersonate

each other, the protocol can be easily enhanced at the cost of one signature verification at the mobile entity side, one signature more from the visited domain and public keys storage at the mobile entity side of each domain in the profile.

The storage space requested in mGAP for the ticket is reasonable. If we consider a subscription (SUB) size equal to an X.509 certificate (1 Kbits); a symmetric key (K) of size 128 bit, and a bloc signature of size 192 bits (using elliptic curve scheme) then the ticket size will be less than 2 Kbits.

For more details on the performance analysis, see the last chapter of the thesis, where I provide some quantitative performance analysis of mGAP using the lower power device network simulator called TOSSIM.

4.6 Related Work

Several architectures exist to manage authentication across networks, but to the best of our knowledge none of them was built with a specific support for mobile groups.

In cellular networks the authentication is based on symmetric shared key between mobile node (MN) and the authentication center (AUC) plus a challenge response scheme [69] [67] [3]. When MN roam to a new location, the visited subnetwork or network request the authentication vectors from the MN's old location for the re-authentication process.

The IETF developed another architecture called mobile internet protocol (MIP) for security and mobility management. The MIP authentication was not efficient for across network authentication because of either heavy computation [58] [31] or multi proxy operation [73] [41]. Recently MIP was enhanced using brokers at the networks edges [43] [36]. Thus the MN continues to be authenticated by proxy to its home network.

Other architectures use expensive key agreement protocols, where the parties contribute to the creation of authentic session key using values certified by common trusted authority. This obliges the MN to perform many signature and signature verification which causes expensive exponentiations for each session and new coming node. Those protocols store a pre-computed modular exponentiation for each new authentication, and combine them with signature verification such as in the Bresson et al protocol for low power devices [16] and the protocols in [15]. The multi parties protocols suffer from lack of efficiency because of the required sequential role of the participants, the non granted reachability in wireless environment [8] [17] and public key storage in MN.

Tab. 4.2: Comparison of authentication protocols for low power devices.

Protocols	Messages	Computation	Authentication
Our protocol	3	n*hash + 2 sym encrypt 1 sig	Mutual/ Local
MIP/challenge response	3	+/- 3 keyed hash	One way/ By proxy
MIP/SSA	3	3 sig verify 1 sig	Mutual/ Local
MIP/AAA	3	3 sym encrypt	Mutual By proxy
GSM/GPRS	3	1 sig	One way/ By proxy
G3/GSM	3	1 sig	Mutual/ By proxy
Yacobi Shmueli [81]	2	3 exp	Mutual/ Local
ASPeCT [30]	3	1 hash 2 expo 1 sig 1 sym encrypt 1 sym decrypt need PKI	Mutual/ Local
Pointcheval et al. [16]	3	3 exp 3hash	Mutual/ Local
Wong and Chan [80]	3	1 exp 1 asym encrypt 1 sym decrypt	Mutual/ Local
Bresson et al. [17]	2	2 exp 1 sig	Mutual/ Local

Acronyms: sym: symmetric, sig: signature, encrypt: encryption, exp: modular exponentiation, hash: hash function operation, asym: asymmetric.

Table 4.2 compares our protocol with different protocols in the literature using the communications and computation cost as criteria. It explains the way the authentication is performed and ignores the known attacks on some of them. To the best of our knowledge, our protocol is the only one providing authentication credentials not dependent on specific visited domain and does not assume any pre-computation.

4.7 Conclusion

In this chapter, we presented a new authentication architecture and protocol, mGAP, for group authentication in mobile networks. Our solution uses mobile access points and aims at predicting the authentication profile of the mobile entities. The authentication profile can be updated in case of unexpected changes, while keeping the possibility of authentication by proxy to the home domain, whenever needed. The mGAP is based on a new cryptographic primitive (group encryption) that is used for the first time in mobile networks. This primitive allows for a flexible and scalable authentication process over different administrative domains.

Further research can explore nominative signature schemes (where the signature is verifiable only by the nominee) to generate credentials according to the resulting profile. It is also possible to explore how to integrate authentication delegation with mGAP in a common framework, because no single authentication protocol can satisfy the diverse needs of authentication in mobile networks. The next chapter provides an Imbricated protocol suite which combines mGAP with authentication ticketing protocol of Chapter 3. The Imbricated protocol allows roaming entities to create a secure session within the visited network. This session is secure even against the other domains in the mobility profile.

5. AN AUTHENTICATION PROTOCOL SUITE FOR ROAMING USERS AND GROUPS

5.1 Introduction

In this chapter, we present an Imbricated protocol for authentication issues, implementable in different mobility contexts. Differently from the existing literature, this protocol is suitable for the authentication of intra and across networks mobility. It is based on our previous works concerning fast roaming in WLANs and mobile group authentication mGAP.

The goal of this protocol is to manage the sudden changes of the environment regardless of the mode of operation. Furthermore, (i) it allows authentication in both ad hoc and infrastructure mode; (ii) it deals with entities' authentication in intra-network as well as in extra-network; and (iii) it manages group and mobile individual node authentication for several flexible mobility models as for instance mobility aware, neighbor graph, predefined path, etc.

More importantly, the protocol (i) should not require expensive additional infrastructure deployment and (ii) should perform seamless secure mobility, which involves reducing the latency of the authentication process.

Most of the existing protocols provide efficiency solutions for authentication either when mobility is intra network or it occurs across networks. For instance, Mishra et al. [48], Park et al. protocols [55] [56], and ourselves in Ticketing and Token authentication protocols [52], propose to enhance the efficiency of the authentication process only when mobility is predicted to be intra network. On the other hand, MIP protocol [58] [73] [43] and our mGAP protocol [4] provide efficient authentication for mobility across networks. Consequently, a comprehensive solution of either scenario becomes mature in our consideration of authentication in wireless networks. Indeed, we propose the Imbricated authentication protocol on basis of our previous experiences and knowledge of the environment. We attempt to design an authentication protocol that achieves the previously listed goals and satisfies the requirements described before. Namely, it deals with the authentication of groups and nodes under intra and across network mobility preserving

efficiency requirements. We propose an Imbricated protocol which is built by upgrading and combining both our authenticating ticketing protocol and mGAP protocol in one authentication protocol.

Next section provides an overview of our protocol and a fast recall of mGAP and authentication ticketing protocols. Section 5.3 details the Imbricated protocol as well as the conditions and the management architecture behind this construction. Section 5.4 analysis this construction while section 5.5 gives implementation results. The section 5.6 recall the related work and finally section 5.7 concludes this chapter.

5.2 Protocol Constructs

This section describes the basic constructs of the protocol suite while the next section details the construction of the Imbricated protocol suite that extends our work in [4] (group authentication protocol, mGAP) and [52] (intra-network authentication ticketing protocol).

Authentication model The proposed Imbricated protocol suite is based on the generic authentication model we developed in [53]. This model divides the authentication process into several generic steps: bootstrapping, pre-authentication, credential establishment, authentication, and finally expiration or revocation of credentials. An authenticating node can go back to credential establishment using credential renewing. Generally, authentication protocols follow these steps, albeit, to varying degrees.

Mobility profile Mobility prediction is a primordial component of the Imbricated protocol. It provides a mobile entity (node or group of nodes) with authentication credentials for a large set of networks before the entity leaves the home network. Mobility prediction is based on entity profiling as detailed for our mGAP protocol. A mobility profile is initially constructed at the home network and it consists of a list of networks where the entity most likely may roam. The profile is established according to the contracted services, the set of preferences, geographical location, network policy and historical behavior. The profile contains also a default list of networks for the roaming entity. Thus, the authentication credentials are tailored with respect to the resulting profile. Moreover, the roaming profile as well as the authentication credentials can be updated by the visited network.

The proposed protocol suite assumes that each visited network possesses its own mobility prediction and management scheme. These schemes may use dynamic neighbor graph, topology, pre-configured path, static frequent handover region, etc.

Authentication credentials The Imbricated protocol suite provides the mobile entity (node or group of nodes) with a mobile credential that is created specifically for the mobility profile of each entity. The credential is based on our group authentication protocol mGAP [4] and a cryptographic primitive described in [37]. In a group of n members, the primitive allows any member of the group to select a subset t of n to which to target an encrypted message. For the latter to be decrypted, it is sufficient to have only one member of the subset t . The message can be decrypted by nobody out of the subset t .

Thus, after the initial authentication by the visited network, the latter creates and then distributes authentication credentials to the predicted internal roaming destinations according to its internal mobility prediction scheme. The distribution and the creation of these credentials use the concepts of our authentication ticketing scheme in [52]. Indeed, the visitor entity and the visited network use the master secret resulting from the first authentication to derive the roaming credentials. Moreover, when the roaming destinations receive the authentication credential of the roaming entity, they cannot only authenticate the roaming entity, but also renew independently the authentication credential for recurrent visits.

5.3 Imbricated Protocol Suite

The Imbricated protocol suite considers the mobile entity either as a support for a mobile group or as an individual mobile node. In the home network, the mobile entity can be either a member of an ad hoc network or an access point offering access to a set of mobile nodes. In both cases, the mobile entity uses the Imbricated protocol suite to authenticate itself or to authenticate on behalf of its attached nodes.

After the profiling process at the home network, the mobile node is provided with the adequate roaming credential for a list of networks in its profile. The roaming credential design is based on the cryptographic primitive described in [37], which allows any member of the targeted subset t of an ad hoc group of n receivers to recover the message. The notation of this scheme is $(1, t, n)$ where 1 represents the single sufficient receiver who can decrypt the message, and who is member of subset t ; t represents the number of targeted members; and n is the total number of group members formed by the sender (home domain HD). Importantly, the encrypted message recovery is accessible only by the members of the subset t . For the protocol notations see table 5.1.

Tab. 5.1: The protocol notations

U, HD, V	mobile entity, home domain, visited domain identities
SUB	subscription contract includes: validity time, authorized services, (U) , (HD) , S_U
S_U, s_U	mobile entity verification and signature public key pair
S_{HD} and s_{HD}	home domain signature/verification public keys pair
N_A, N_U, N_V	user nonce's, visited domain nonce's
K_{HD-U}	shared key between HD and mobile entity
K	roaming authentication key delivered by HD
H	a hash function
$PKG(1, t, n)$	public key to encrypt messages group of domains t from n
Ticket	$E_{PKG}(K, SUB, s_{HD}(K, SUB))$
The authenticator	$E_K(SUB, N_U, N_V)$
$E_V(N_U, N'_U)$	Encryption of the nonces with the public key of V
MK	A master key resulting from across network authentication
PMK_i	perwise master key shared for authentication with AP_i
$Cr d_i$	$(ID_{AP_i}, E_{K_i}(PMK_i, ID_U, Times))$
K_i	secret key shared between the server of the visited domain and APs in visitor mobility pattern
n	is number of roaming destination in the mobility pattern

Differently from mGAP, the execution of the Imbricated protocol produces a pairwise secret between the visited network and the mobile entity. Moreover, it creates and distributes proactively the authentication credentials for roaming within the visited network. These credentials are based on the upgraded concepts and functions of our authentication ticketing scheme [52]. The protocol exchanges are processed as follows:

1. $U \longrightarrow HD : U, N_A$
2. $HD \longrightarrow U : Ticket, E_{K_{HD-U}}(K, N_A, SUB)$
3. $V \longrightarrow U : N_V, Cert(V)$
4. $U \longrightarrow V : Ticket, Authenticator, E_V(N_U, N'_U), s_U(H^n(N_V, N'_U), V)$

5. $V \longrightarrow U: E_K(N_U, N'_V)$
6. $V \longrightarrow AP_0: Crd_i, 0 \leq i \leq n$
7. $AP_0 \longrightarrow AP_i: Crd_i, 0 < i \leq n$

We describe the steps of the Imbricated protocol according to the authentication model in section 5.2.

- *Bootstrapping*: the home network provides the mobile entity with a shared symmetric secret (K_{HD-U}), a contract that contains the mobile entity's public key certificate, validity time, authorized services, the identity of the mobile entity and the identity of the home network.

The home network also provides the mobile node with a ticket, namely, the roaming authentication credential according to its mobility profile. The mobile node as well as the authentication servers of the potential visited networks are bootstrapped with equation (1). This is used for the proactive derivation of the authentication credential for roaming within the visited network.

$$PMK_i = TLS_PRF(MK, H(MK \| U_MAC \| AP_i_MAC)) \quad (5.1)$$

The pairwise key PMK_i is the master key distributed by the authentication server of the visited network proactively to the next potential roaming destination AP_i of the mobile node U . MK is a master secret resulting from the across network authentication with the Imbricated protocol suite. AP_i_MAC and U_MAC is respectively the access point and the mobile node MAC addresses, i is the index of the AP ($0 \leq i \leq n$), where n is the number of AP s in the mobility pattern of the visited network and H is a secure one way hash function.

The AP s, as well as the servers of the visited network, are bootstrapped with the mobility patterns and a shared key k_i between the server and each AP .

Finally, they are bootstrapped with equation (2) for the computation of the credential for recurrent visits.

$$PMK_{im} = TLS_PRF(H^m(PMK_i), Counter \| STA_MAC \| AP_i_MAC) \quad (5.2)$$

where PMK_{im} is the pairwise master key proactively derived by the AP_i and mobile node U for authentication at the m^{th} visit.

- *Pre-authentication*: starts when the mobile node U arrives at the visited domain V , and receives from the latter the nonce N_V and the certificate

$Cert(V)$ in message (3). Consequently, U generates two nonces N_U and N'_U . U creates the authenticator using K . U uses the visited network public key to encrypt both nonce N_U and N'_U for V . U also computes the signature $s_U(H^n(N_V, N'_U), V)$.

Finally (U) sends the previous values as well as the ticket received in message (2) to the visited network. The pre-authentication step finishes when the visited domain V receives message (4).

- *Credential establishment:* starts when the visited domain V treats the message (4). It decrypts the ticket, recovers K , decrypts the authenticator, recovers N_U , N_V and SUB . V also verifies that the N_V matches the one it was sent in message(2), then establishes K as authentication key. V decrypts $E_V(N_U, N'_U)$. V creates a new nonce N'_V and computes the master secret $MK = H(N'_U, N'_V)$ between V and U . This completes the credential establishment and the one way authentication U to V .

The authentication server of the visited network computes according to its roaming pattern a set of authentication credentials using equation (1) for local roaming. This completes the credential establishment and the one way authentication U to V . The session key between the primary AP and the mobile node will be derived from the PMK_0 .

- *Authentication:* When the mobile node (U) receives message (5), it knows that it is authenticated to V and can authenticate V . Message (5) confirms to U the authentication key K . This also means that the mobile node is pre-authenticated to the AP s in the mobility pattern since the primary AP has already received roaming credential in message (6) and distributes these local authentication credentials to those AP s ahead of mobile node motion in message (7). Thus, the mobile node and the primary AP can start a challenge response based on a four way handshake [2] session key derivation using PMK_0 .

For recurrent visits the AP s compute and cache an authentication key for the next expected visit of the mobile node after the latter leaves. Equation (2) shows how the authentication material of recurrent visits is computed.

- *Monitor behavior:* The visited domain V monitors the validity of the mobile node credential and subscription contract SUB . If the credential is expiring, a new authentication process starts from the credential establishment step. Notice that the authentication material for within network authentication in the visited networks coincides with the validity duration of the original contract with the home network. However, if the roaming agreement between home network and visited network allows contract renewing at the visited one, the credential can be created for a duration that goes over the contract deadline. The storage duration of pre-authentication

credential and recurrent visits authentication is set according to the visited domain policy.

5.4 Analysis

5.4.1 Security analysis

We consider a threat model where an attacker can be active or passive, thus he can attempt to eavesdrop on exchanges, inject messages, alter the exchanges, or replay old messages. We assume that each network authority has or can easily obtain the authentic public key of the different network authorities. Since there is roaming agreements, each network authority's public key is certified by the other networks' authorities party to the roaming agreement. This is a reasonable assumption since the number of networks is much smaller than the number of mobile entities. Given this threat model and the above assumption, the reaction of the Imbricated protocol suite can be summarized as follows:

- *Freshness*: The protocol provides the mobile node with a fresh roaming authentication key ' K ' via a secure message (encrypted using the long term shared key K_{HD-U}) in message (2). An attacker can neither recover K without knowing K_{HD-U} , nor can he use an old key since the roaming authentication key K is encrypted and sent with the nonce N_A received by HD in message (1). The presence of N_A in the non ticket part of message (2) proves the freshness of K for the mobile entity U .
- *Confidentiality*: Message (2) also includes a ticket, which is the transportable credential to authenticate the mobile node to the visited network and vice versa. This ticket is encrypted using the group's public key encryption algorithm. The ticket also includes K and SUB as well as their signature by HD ; therefore an attacker (non member) of the group cannot recover K .
- *Integrity*: The signature of HD over (K, SUB) provides V with an authenticated key. The signature of HD ensures the non- repudiation of roaming credential by HD to V ; moreover, the signature of the SUB by HD certifies the validity time, the allowed services, the roaming entity identity, and U 's signature verification key.
- *Impersonation*: Message (3) of the visited network to the mobile node, provides the latter with a certified public key of the visited network that is easy to verify since it is signed by the home network of the mobile node. Message (3) also contains a nonce that will be used by the mobile node in message (4) to prove the freshness of the latter.

- *One way authentication:* Once message (4) is received, V decrypts the ticket to recover K and decrypts the authenticator. Only a member of the group can decrypt the ticket and the authenticator. The presence of N_V sent to U in message (3) within the authenticator proves the freshness of the authenticator and proves that U knows the key K ; therefore an attacker can not impersonate U to V , or even replay an old authenticator. Thus the one way authentication U to V is established.

- *Non forgery:* The signature part in message (4) again proves the identity of U and that U requests the access to the nominated domain in the signature. An attacker can not impersonate U to create a fresh signature. The signature also contains N'_U which provide V with authentic contribution in the master key creation. The value $E_V(N_U, N'_U)$ provides V with N'_U in a secure way, such that as only V can recover N'_U since it is the only one possessing the private key to decrypt $E_V(N_U, N'_U)$.

The freshness of the signature is ensured by the presence of N_V . The fresh signature of U includes the chosen domain's identity. This shows from which domain the service was requested and allows for protection against impersonation and rogue domain. However, the domains of the profile that have not been chosen for the present transaction are trusted to not impersonate each other to the mobile entity. The n^{th} hash included in the signature allows the mobile entity to request different services or different parts of services. This is true by giving an earlier signed hash value from the given hash chain instead of re-authentication; this practice is often used in micro-payment schemes [13].

- *Authenticated key establishment:* When V treats the different part of message (4) it creates the nonce N'_V . So that it can compute the pairwise master keys (PMK_i) for the primary access point AP_0 and the AP s in the local mobility pattern. The keys PMK_i will be the re-authentication basis within the network for the mobile node. Moreover the PMK_i will be the basis of session key derivation between the mobile node and the AP s in the local mobility pattern. Thus, the transaction inside the visited network will be secure even against the other networks in the profile of the mobile node.

- *Mutual authentication:* When the mobile node receives message (5) the mutual authentication across network is accomplished. This message contains visited domain contribution N'_V in the creation of the master secret MK . Thus the mobile node can compute PMK_i for authentication while roaming within the network .

For future across network roaming, V and U repeat exchanges (1) and (2), where V will use a new nonce N'_A and create a new ticket. V also uses the session key previously derived from PMK_i s to provide U with a

new authentication roaming key K' . Note that K , the session keys as well as K_{HD-U} are at least AES-128 keys, used for limited time and specific operations. Their corruption by an attacker is impossible in practice.

5.4.2 Performance Analysis

The Imbricated protocol suite supports authentication of roaming entities across networks in both infrastructure and ad hoc modes. Moreover, it supports the authentication of the mobile entity accompanied by a group of mobile nodes and authenticating on behalf of them.

- *Computation overhead:* This protocol shifts most of the computation burden on the domain side. The mobile entity performs one symmetric encryption, one symmetric decryption, n hashes for each roaming destination, one low cost signature [32] and finally one asymmetric elliptic curve encryption.

- *Storage overhead:* Our protocol does not require high capacity for storage or pre-computation before roaming to a new destination. The mobile node does not need to decide in advance the precise destination of our protocol tolerates extra-profile roaming using proxy authentication. The profile can always be updated according to changing needs. On other hand the storage space requested in the Imbricated protocol at the mobile node side for the ticket is reasonable, if we consider the subscription (SUB) size equal to X.509; the certificate size (1 Kbits), the symmetric key K size equal to 128 bits and the bloc signature key size is 192 bits (using elliptic curve scheme). This yields a ticket size less then 2 Kbits. On the other hand, for the within network authentication, the AP s in the mobility pattern store a PMK s of 256 bits. And the mobile node computes the PMK_i for each visited destination instead of storing them.

- *Communication overhead:* From communication point of view, the Imbricated protocol suite uses only 3 message exchanges (messages 3, 4 and 5) to perform a mutual authentication and key establishment between U and V .

5.5 Implementation and Results

The implementation of the protocol shows the performance from the communication point of view. This section is dedicated to an example of implementation of the Imbricated protocol. Recall that the efficiency of the intra and across network authentication is easily performed higher is the resources of the nodes and of network where these authentications take place.

Our strategy to test the protocol is to choose the "poorest" network and corresponding nodes. If in this case implementation not only succeeds, but it also overcomes existing protocols in terms of authentication's efficiency, our example is the proof that the Imbricated protocol provides the tool to reach efficient authentication, in intra and across modes, in any wireless network endowed with higher level of resources. In such network, there are many applications using mobile nodes (e.g. mobile network of low power devices), where an authentication protocol for several roaming scenario is needed. We implement our Imbricated protocol suite on Wireless Platform for Low-Power device Networks (8-bit microcontroller Atmega 128L) called Micaz. The network uses the frequency 2.4 GHz and it is IEEE 802.15.4 compliant. The platform provides up to 250 kbps data rate. The nodes in this kind of network shares the radio channel, thus, to avoid collisions they listen to the channel and they transmit only if it is free. It follows that our experiences cannot take in account the impact of traffic load on the network while measuring performance parameters.

In the experiments conducted, we have run the Imbricated protocol between the mobile node U and the node V in the visited network. The node V is networked with V_a , V_b , V_c and V_d in ad hoc mode. Node V can also be either directly connected to a distant node HD from U 's home network or via intermediary relay, R_1 , $R_1, \dots, R_n$ to perform proxy authentication of U . Moreover, after U is authenticated by V , it can roam to any node of the group V_a , V_b , V_c , V_d using the Imbricated protocol.

In our experiments, the visitor U authenticates to V using the Imbricated protocol. U performs this authentication when V is under different computation load. This for first authentication, then, U authenticates to one of the nodes networked with V , such as V_a , V_b , V_c and V_d , using the feature of within network authentication of the Imbricated protocol (see Figure 5.1). Within network authentication is also performed for different loads. The load on the authenticator node is specified as the time after which the node is able to process a new coming request. Thus we can measure the latency of the authentication process to compare the performance of different authentication. For the proxy authentication, in addition to vary the load on HD , we have varied the number of intermediary relay between V and HD .

As we are interested in performance based on latency variation, for simplicity sake, the latency of cryptographic operation was replaced in our implementation by a dummy one. The used values were extracted from a literature focused on optimized implementation of cryptographic operation in sensor networks [7], [77]. The used values are

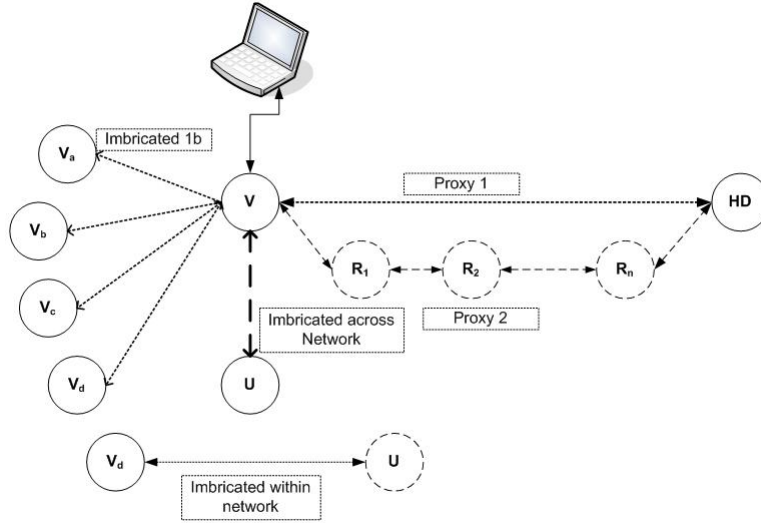


Fig. 5.1: Network model authentication possibilities

- 160-bit elliptic curve digital signature algorithm ECDSA sign:1.925s; 160-bit elliptic curve digital signature algorithm ECDSA verify: 2.433s.
- 128-bit AES encrypt/decrypt for 16 bytes: 3.75 ms. AES is also used to compute hash values.
- The table below presents the authentication latency when using different authentication protocols on our Micaz nodes.

Tab. 5.2: The protocols latencies

Protocol	Latency
Imbricated across networks	6448 ms
Imbricated within network	115 ms
symmetric proxy	122 ms
asymmetric proxy	6795 ms
EAP-TLS	8800 ms

These values are computed without any load on V or on HD . The Imbricated protocol suite is executed between U and V . The execution of such protocol in the within network authentication is reduced to a session

key derivation, as described by the four way handshakes protocol of the security standard IEEE 802.11i. This is true because the authentication credential was distributed ahead of the mobile node U to V_a , V_b , V_c and V_d . However, the authentication protocol by proxy is a protocol using symmetric cryptography to authenticate a mobile node U by transferring these duties to the node HD in the home network of the mobile node U (see Table 5.2). For instance, the symmetric proxy authentication is widely used in GSM network and wired computer networks. The proxy asymmetric protocol performs the authentication of mobile nodes in the same way but uses asymmetric cryptography. The EAP-TLS is the most used authentication protocol in WLAN for within the same network authentication.

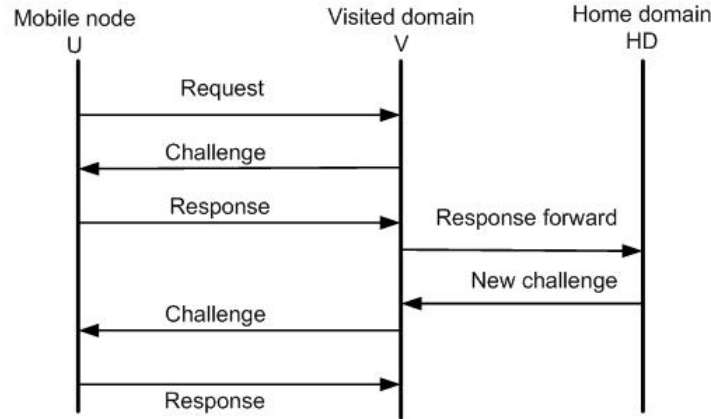


Fig. 5.2: Generic proxy authentication exchange

When there is no load on the network, the protocols have the nominal latencies shown in Table ???. As the Imbricated protocol suite for across and within network authentication is always faster than proxy asymmetric protocols. We do not analyze the proxy asymmetric authentication protocol any further.

The authentication latency of a symmetric mutual authentication protocol depends on the number of the intermediary relays in the path between V and HD . We notice from our implementation that each relay adds approximately 16.75 ms, if we consider no specific load on the relays of the path

V to HD . Figure 5.3 shows the latency of the different protocols when there is equal load on V and HD and no relays between them. The result shows a net advantage for proxy authentication but this process must be repeated each re-authentication. However, with Imbricated protocol only the first authentication is costly while the re-authentications for roaming in the same network will use the generated credential for within network authentication. Hence, the Imbricated protocol becomes faster than proxy authentication. Moreover, the network condition can change at any moment and the proxy authentication can fail or slow down. Furthermore, in real conditions, often the proxy authentication passes through several relays incurring many encryption/decryption and path related delays.

Figure 5.3 shows the latency impacts of an increasing load applied the both V and HD . We can see that the latency of Imbricated protocol for across network authentication is higher than the latency of symmetric proxy authentication, but the authentication latency of the Imbricated protocol for within network roaming continue the be lower than the proxy protocol. When the proxy authentication request has to cross several relays in the

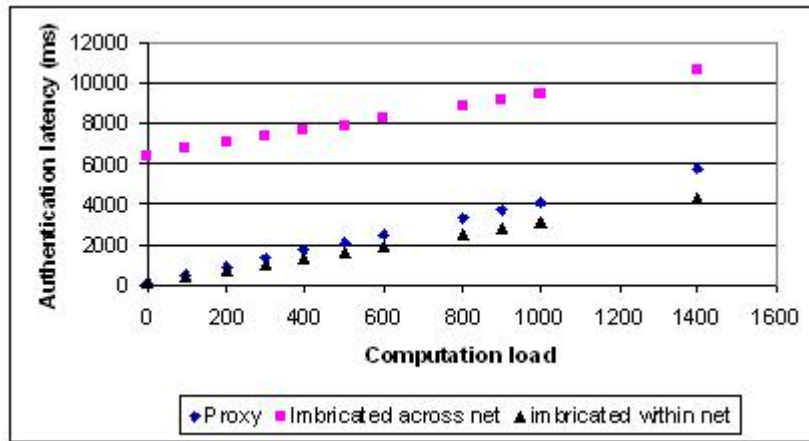


Fig. 5.3: Latency variation of Imbricated protocol for within and across network authentication VS symmetric proxy authentication. The same load is applied on V and HD and no relay between V and HD

path between V and HD , the authentication latency increases dramatically. It covers the communication latency introduced by each relays and the latency due to the computation load on V , on the relays R_i , and on HD . However, the latency introduced by the relays R_i and HD does not impact

the Imbricated protocol, because the latter treats locally both across and within network authentication.

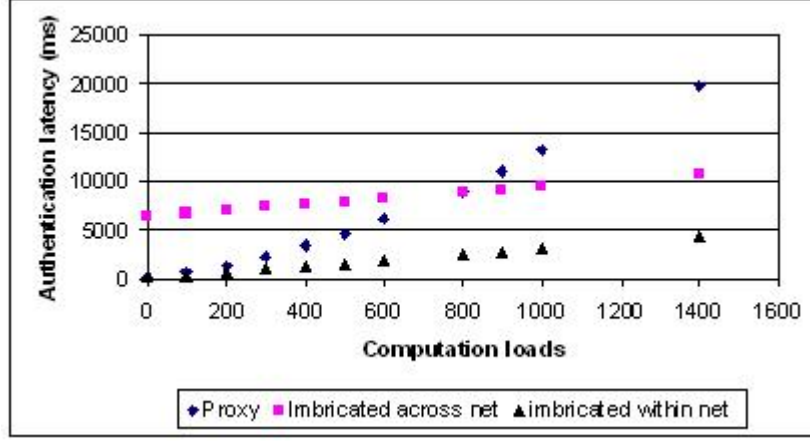


Fig. 5.4: Latency variation with increasing number of relay between V and HD and increasing load on the relay .

From figure 5.4 we can conclude that first authentication with Imbricated protocol is faster than proxy authentication starting from the point where there are seven relays, with moderated load in the path V to HD . This shows clearly the advantage of the Imbricated protocol towards the symmetric proxy authentication in real network conditions.

The consideration of other kinds of network should modify these results in favor of Imbricated protocol. Wireless Sensor Networks (WSN) are indeed not suitable for heavy public key cryptography (PKC) computations as opposed to WLANs. In WLANs, PKC computations can be 700 times faster than in WSN (see [65], with an AMD Athlon 1467 MHz) while communications have a throughput roughly 150 times bigger (practical throughput of 19 Mbps in 802.11g). A quick estimation for a WLAN in a similar situation as described before (loads scaled down by 700) gives the following latencies: 8 ms for across network authentication with Imbricated protocol, 2 ms for within network. However, proxy symmetric authentication costs 3.3 ms. Consequently, the number of relays, from when the Imbricated protocol becomes faster than proxy scheme, is lowered to the value of 4. This is because the latency of Imbricated protocol is mainly composed of computations delay, while the latency sources of the proxy symmetric protocol are more diversified. Simulations should be carried out in order to provide

more accurate results.

5.6 *Related work*

To the best of our knowledge, our work is the first to present a comprehensive solution adaptable to both intra-network and inter-network authentication. In cellular networks, the authentication is based on a symmetric shared key between a mobile node (MN) and the authentication center (AUC) plus a challenge response scheme [69] [67] [3]. When the mobile node roams to a new location, the visited sub-network/network requires the authentication vectors from the old location of the mobile node for the re-authentication process. The ITF developed the mobile Internet protocol (MIP) for security and mobility management across networks. The MIP authentication was not efficient because of latency due to either heavy computation [58] [31] or multi proxy operation [73] [41]. Recently, MIP was modified to reduce the authentication latency using brokers' entities at the networks edges [43] [35]. Thus, the mobile node continues to be authenticated by proxy to its home network. Other architectures use expensive key agreement protocols [16] [15], where the parties contribute in authentic session key derivation using expensive exponentiations for each session and new coming node. Those multi party protocols suffer from lack of efficiency because of the required sequential role of the participants, the non granted reachability in wireless environment [8] [17] and the need for public key storage in mobile nodes.

Other works have focused on reducing the authentication latency for a roaming node within the same network, for instance, the IEEE 802.11i standard roaming authentication scheme [2] uses a pre-authentication to proactively authenticate the mobile node. The predictive authentication scheme [55] [56] predicts the mobility pattern using central information about the network configuration and the node's behavior to proactively authenticate the roaming nodes in static Frequent Handoff Region. Recently, a proactive key distribution scheme [48] uses a centralized (at the server) dynamic data structure of the mobility pattern to proactively distribute the authentication material to the predicted roaming destination, at the cost of permanent dependence on server. The standard IEEE 802.11i also proposes a reactive scheme where heavy full authentication (e.g. EAP-TLS) is repeated at each re-authentication. Finally, we proposed in [52] a proactive and hybrid scheme for authentication within the same network.

5.7 Conclusion

Mobility and permanent connectivity are the target of future networks and applications. To provide secure connectivity in such environment, we propose the Imbricated protocol. Our protocol guarantees the authentication of the mobile node (whatever this node is) across networks and within the visited networks. Our solution can adapt to the mobility schemes of the visited networks without further configurations. The Imbricated protocol gives wide mobility freedom across networks and within visited networks, because it exploits the profiling mechanism of the home network and the internal mobility scheme of the visited networks. The implementation proved the advantages of the local authentication and re-authentication provided by the Imbricated protocol for across and within network roaming against others methods based on proxy authentication.

6. TWO LEVEL SIGNATURE DELEGATIONS FOR MOBILE FIRST RESPONDER AUTHENTICATION

6.1 *Introduction*

The increasing deployment of wireless networks covers both civilian and military domains. One of the most important networks that recently adopted the wireless technologies to exploit its mobility is first responder networks. The nature of the first responder work requires collaboration among heterogeneous and mobile groups, which creates many security challenges like user authentication, information flow and wireless media management across heterogeneous domains.

Frequently, during disasters, when wired networks get damaged or become inaccessible, wireless networks may be the only communication services available for first responder agencies. Wireless networks in infrastructure, ad hoc and mixed mode could enhance these agencies work performance as well as information and resource sharing between their mobile units. For more effective disaster intervention, there is need for developing wireless network technologies that would enable the secured cooperation and collaboration among different first responder agencies. Current authentication mechanisms employed may block not only attackers but also potential collaborators from different agencies. Indeed, authentication of mobile users across different administrative networks is a challenging proposition. A primary challenge is to create an authentication solution that maximally decentralizes the authentication process to aid mobile individuals and groups.

The authentication solution proposed in this chapter aims to endow a mobile entity with a unique credential used across different administrative networks. This unique credential should allow a mobile entity to roam and authenticate in new visited networks. The solution should be applicable to both infrastructure and ad hoc networks. It should also take into account the limited computation, storage and power resources of mobile entities. Many schemes in the literature studied the authentication issues in ad hoc network. However, based on our knowledge, none provides a satisfactory solution that can meet the authentication needs of first responders (see [5] [83]). Most

existing authentication schemes do not support the smooth authentication of mobile entities across different administrative networks. They often either require from the new coming user a physical contact with an offline or online central authority of the visited network to be provided with new credential, or the local authentication authority authenticates the visiting user by proxy through the contact with mobile entity's home network, regardless of the length of the path and the cost of the process.

This chapter proposes a decentralized authentication scheme for roaming nodes across networks. Our scheme permits secure resource sharing between nodes from different first responder networks. To achieve this objective we leverage the proxy signature technique introduced by Mambo et al [44] to build a scheme that guarantees authentication of mobile groups and individual nodes across different networks.

In general, the signature delegation schemes, also called proxy signature schemes, consists of three entities; the original signer (designator), the proxy signer and the verifier. Proxy signature schemes generally fall in one of the following categories:

- **Full delegation:** The designator gives its secret key to the proxy signer that can then use it to sign the messages. In this case, the proxy signature is just a standard signature and can be verified in the usual way. This kind of delegation has however many limitations, the most important being that its security relies on the honesty of the proxy signer. Moreover, it requires a secure channel between the original and the proxy signer to transfer the secret key. This kind of delegation makes the signatures of the original and the proxy signer undistinguishable.
- **Delegation by certificate (or delegation by warrant):** The designator uses its secret key with a signing algorithm to sign the warrant. The latter may contain the proxy signer's public key, a period of validity and the restriction of the class of messages for which the warrant is valid. The signature of the originator on the warrant is called the certificate and is sent to the proxy signer with the warrant. The proxy signer then uses its own secret key to sign on behalf of the designator, and sends the proxy signature with the warrant and the certificate to the verifier. The verifier needs to ensure that the certificate is valid with respect to the public key of the designator; to verify the second signature using the public key of the proxy signer specified in the warrant; and to check the conformity of the proxy signature to the restrictions in the warrant.
- **Partial delegation:** The designator generates a proxy signature key from its private key and gives it to the proxy signer, such that the proxy can sign on behalf of the designator using the proxy key. The proxy signature does

not allow the proxy to keep its identity secret. Moreover, the designator can impersonate the proxy signer.

Our scheme assumes that each entity has its own authentic public-private key pair. The scheme divides the certifying authority into three levels. The first one represents the highest authority that may be a government or a national disaster management authority (DMA). This authority is trusted by each entity involved in the scheme. DMA certifies the public key of each first responder agency, and delegates to the latter the capacity to issue certificate on its behalf. The delegation is regulated with some conditions.

The second level of authority is the first responder agency. This agency issues certificates for its members (subscribers) on behalf of the DMA. Thus, the delegation operation avoids the bottleneck at a centralized certificate authority.

The third level of authority is composed of the first responder agency's subscribers. Because of the reachability and mobility problems in a mobile network, the agency authority cannot always verify and renew the certificate of each member and each visitor. This is why the agency distributes these capacities to its subscribers in such a way that a threshold of subscribers can perform these duties on behalf of the agency. This delegation is limited and it allows that only a set of subscriber can verify, renew, and authenticate a certificate of a member or a visitor. The threshold ensures strong security against colluding members and prevents problems due to the unreachability of the central certification authority.

Our two-level delegation construction allows for mobile node authentication, certificate renewal and revocation whenever needed. The scheme construction does not require all users to know and to store the public key of other entities in other first responder networks. Furthermore, each first responder network can choose its own level of security by setting its local threshold delegation without harming the cooperation. The scheme proposed in this chapter provides the security services in case of leaving and joining nodes, or radio channel errors and network partitioning.

Important to note is that the combination of two hierarchical levels of signature delegation (partial proxy signature with warrant and threshold proxy signature with warrant) is *not* a straightforward extension of the proxy signature. To the best of our knowledge, this construction is the first to guarantee strong authentication across different mobile networks in both infrastructure and ad hoc modes.

Our authentication scheme uses a high authority as a reference of trust without involving it in subsequent operations. The scheme guarantees free roaming and secured resource sharing for any mobile node certified by one

of the following authorities: (i) high authority, (ii) first responder agency or (iii) a threshold of subscribers from a first responder agency.

The remainder of the chapter follows with the description of our network model and with an overview of the proposed schemes in Section 6.2. Section 6.3 details the background of the proxy signature construction while Section 6.4 describes the two-level delegation construction. Section 6.5 provides the scheme security and performance evaluation. Finally Section 6.6 gives an overview of the related work. The chapter concludes in Section 6.7.

6.2 Two-Level Delegation Authentication Scheme

Our solution based on signature delegation perfectly fits the first responder networks model. We first describe the network model, followed by a detailed description of our proposed first responder authentication scheme.

6.2.1 Network Model

The network model presumes a *high authority* (i.e. a service provider, the government, a national Disaster Management Authority (*DMA*), etc.). This high authority is supposed to provide services to several first responder agencies and has a strong trust relationship with them. The first responder agencies in turn are the brokers of the services to their local subscribed users/members (see Figure 6.1). This model assumes that the high authority, the first responder agencies as well as their subscribed members are public key capable. This assumption on the nodes is reasonable because of the constant growth of the number of computation capability devices and the spread of saving power consumption technologies. The recent development of the elliptic curve cryptography allows public key implementation in limited resource devices like smart cards [38] or/and sensor networks [64]. Even our construction framework does not use ECC, but it can be adapted to its use. This allows the application of this scheme also for low capable devices.

Moreover, each first responder agency, as well as its subscribers, possess an authentic public key of the high authority. The subscribers in their home network communicate in both infrastructure and ad hoc mode depending on the network condition. Since the first responder networks are mobile networks owned by different agencies like police, army, fire fighters, health care, etc., their collaboration and resource sharing in disaster situations faces inter-network authentication problems. Our authentication scheme is designed for inter first responder networks authentication even in the ab-

sence of previous security association between the visited network and the visitor nodes.

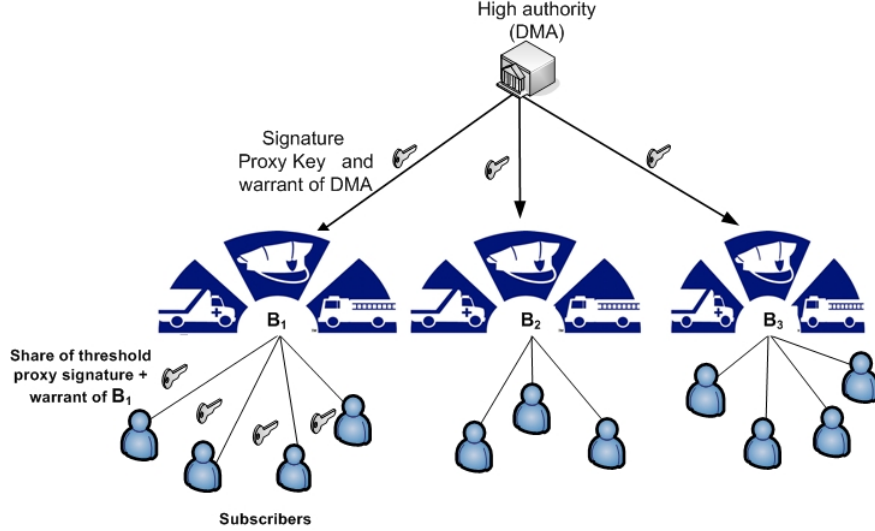


Fig. 6.1: The network model for two level delegation scheme

6.2.2 The Proposed Scheme

The idea behind our scheme is to delegate the signing (certifying) authority into two levels. First, the *DMA* delegates to the first responders agencies its signing capacity. Consequently, these agencies are enabled to issue, renew and revoke the certificate of their subscribers on behalf of the *DMA* respecting the delegation conditions specified by the *DMA*.

For simplicity, let A represent a *DMA*, B_i a first responders agency i , and C_{ij} a subscriber j in the network of agency i , where $0 < i < n$ and $0 < j < m$; n is the total number of agencies and m is the total number of subscribers to the network of agency i (see Figure 6.2). The protocol proceeds as follows.

The disaster management authority A provides each agency B_i with a signed certificate ($Cert(A, B_i)$) and the delegation condition (the warrant m_w).

The warrant contains the identity and the public key of A , the identity and public key of B_i , and the type of messages allowed to be signed by

B_i on behalf of A . The warrant can also include a validity period for the delegation, which represents an efficient revocation technique. As a result, the delegation must be renewed before the expiration date.

Notice that the delegation warrant is different for each agency B_i . When B_i receives the certificate and the delegation warrant, it can create its proxy pair of private-public keys. Using its private (proxy) key, B_i signs the certificate of its subscribers on behalf of A . In order to be authenticated to their home network or at any visited one, the subscribers present their certificate and the delegation warrant to their authenticators. Taking for granted that the public key of A is known to every involved entity, the authentication turns to the verification of the warrant and of the certificate.

The nature of first responder networks and subscriber's mobility makes central authentication authority not adequate to secure this type of network. Thus, a fully distributed authentication authority across the network is more adapted. For this purpose, at the second level of delegation, B_i delegates its signing (certifying) authority to its subscribers in such a way that a threshold of subscribers can perform authentication as well as the certificate renewing duties on behalf of B_i . Each agency can fix its one threshold size according to its security policy.

B_i specifies a new condition for the second level of delegation warrant m_w^* and signs it using its proxy private key. The resulting signature will be used as a new key, and will be divided into shares. The latter are distributed to the subscribers C_{ij} using (t, n) Shamir's secret sharing scheme. Therefore, any threshold of t subscriber can collectively authenticate and renew the certificate of an other subscriber or visitor on behalf of B_i . Hence, in the scenarios where C_{12} (the subscriber 2 from agency B_1) moves to agency B_2 network's, C_{12} broadcasts to its neighbors its certificate signed by B_1 on behalf of A . The threshold verifies the certificate of C_{12} and the correctness of the delegation A to B_1 . When the previous verification succeeds, the subscribers of B_2 certify the visitor. Thus, each member of the threshold responds to the authentication (certificate renewing) request by a partial signature computed using its share. Therefore C_{12} interpolates the partial signature to obtain its new full certificate. C_{12} verifies the new certificate and the correctness of both delegation levels A to B_2 and B_1 to C_{2j} (see Figure 6.3). Hence, our scheme guarantees free roaming and resource sharing between different first responder networks using two-level delegation of certifying authorities. In the next section, we present a concrete example of our scheme construction based on Schnorr signature scheme. We first introduce Schnorr signature scheme, and the proxy signature based on Schnorr, and then the second level of delegation based on Shamir secret sharing.

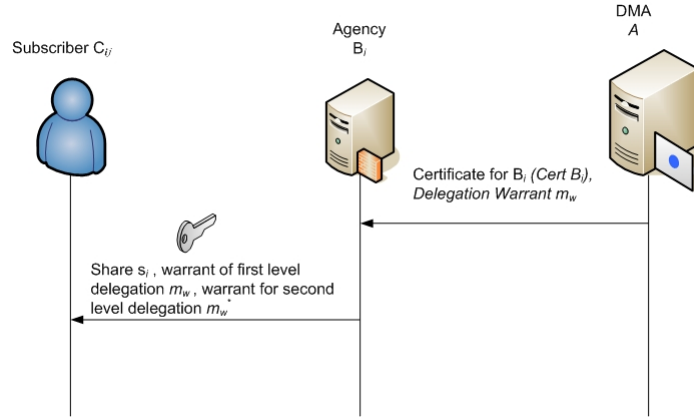


Fig. 6.2: Certificates and shares distribution

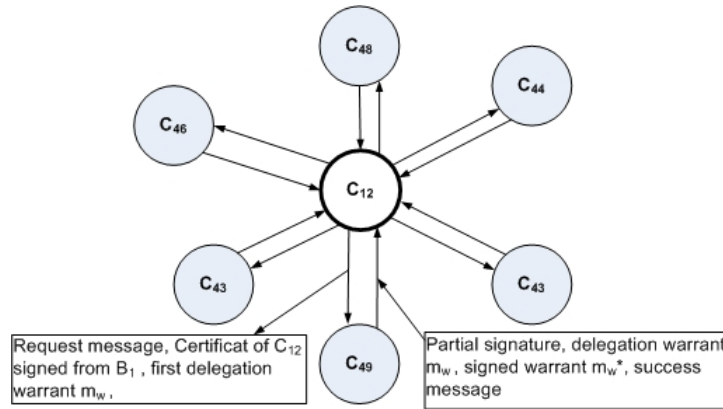


Fig. 6.3: Threshold authentication

6.3 Background

6.3.1 Schnorr Signature

The Schnorr signature [68] scheme is based on discrete logarithms and it is considered as one of the simplest schemes proved to be secure in the random oracle model [21].

All the users of Schnorr signature scheme agree on the Schnorr group G with generator g of prime order q in which the discrete log problem is hard. All users also agree on a secure cryptographic hash function H . Each user

is endowed by a private key x such that $0 < x < q$ and the public key is y where $y = g^x$. To sign a message M , the user chooses a random k such that $0 < k < q$ and computes $r = g^k$, $e = H(M, r)$ and $s = k + ex \bmod q$. The signature is the pair (e, s) or (r, s) where $0 \leq e < q$ and $0 \leq s < q$. The signature validity requires the verification $g^s = ry^e$. If this is true, the signature is valid. Note that (G, g, q, y, s, e, r) are public elements while k and x are private. In [76], Pointcheval and Stern proved that, in a random oracle model, existential forgery under adaptive chosen message attack of Schnorr's is equivalent to solving the discrete logarithm problem in the group generated by the element g .

6.3.2 Schnorr Based Proxy Signature

Many of the proposed proxy signatures have been based on Schnorr's signature scheme. In this context, the highest authority A delegates its signing capacity to a set of sub-authorities B_i , $0 \leq i \leq n$, where n is a finite integer. To build our network architecture we use the Schnorr based proxy signature scheme as applied in [42] [14] in a protected partial delegation with warrant. This combines the benefits of the partial delegation and the delegation with warrant.

1. Let A be the highest authority, which delegates its signing capability to a sub-authority B_i with some restrictions. A chooses a random number $k_A \in_R Z_p^*$ and computes the signature parameter $r_A = g^{k_A}$, $s_A = x_A h(m_w, r_A) + k_A$. The tuple (r_A, s_A) is the highest authority (A signs the warrant m_w ; the warrant should include the information on the delegation like the originator identity ID_A , the proxy signer identity (or public key) and validity period.)
2. A provides in secure way each proxy signer B_i with (m_w, r_A, s_A) .
3. Each proxy signer B_i verifies $g^{s_A} = r_A \cdot y_A^{h(m_w, r_A)}$.
4. If this holds, B_i uses its private, public key pair x_{B_i}, y_{B_i} to compute its proxy keys:

$$\begin{aligned} x_{p_i} &= s_A + h(m_w, r_A) \cdot x_{B_i} \\ y_{p_i} &= g^{x_{p_i}} = r_A (y_A \cdot y_{B_i})^{h(m_w, r_A)} \end{aligned}$$

6.4 Two Level Delegation Scheme

We described previously how proxy signing keys of each sub-authority B_i is created via the protected proxy signature scheme. This makes A unable to impersonate any proxy signer B_i . The warrant m_w decides the limitation and conditions on the proxy signing capacity. Each sub-authority B_i has a finite number of subscribers C_{ij} ($0 \leq j \leq$ total number of subscribers) that B_i provides with a certificates on behalf of A .

1. B_i signs the certificate $Cert_j$ of the subscriber C_{ij} using its proxy private key x_{p_i} , respecting the limits defined in the warrant m_w . B_i computes the elements of the Schnorr proxy signature: $r_j = g^{k_j}$, where k_j is a random number $\in_R Z_q$ $s_j = k_j + x_{p_i}h(r_j, Cert_j)$.
2. B_i provides each subscriber with $\sigma_j = (r_j, s_j, r_A, m_w, Cert_j)$.
3. The subscriber C_{ij} to verify the certificate computes

$$g^{s_j} = r_j \cdot (r_A \cdot (y_{B_i} \cdot y_A)^{h(r_A, m_w)})^{h(r_j, Cert_j)}$$

$$g^{s_j} = r_j \cdot (y_{p_i})^{h(r_j, Cert_j)}$$

y_A presence in the verification proves the delegation.

In this way, each mobile subscriber C_{ij} can authenticate locally or in the visited domains by presenting (σ_j, y_{B_i}) see (Figure 6.4). This proves without any doubt that C_{ij} has credential provided by B_i on behalf of A .

6.4.1 Threshold Delegation

To guarantee more security in the context of first responder networks and to avoid the frequent problem of Certificate Authority (CA) reachability, as well as non adapted centralization of CA , our architecture allows a second level of certificate signature delegation among the first responder agency subscribers. A threshold of subscribers can perform the certification duties with some prefixed restrictions. To perform this delegation among the subscribers C_{ij} , the first responder agency B_i creates a new Schnorr signature using its proxy private key X_{p_i} and a new warrant m_w^* . The signature is the tuple (r_{B_i}, s^*) . Then, B_i divides into shares, s^* and distributes among its local subscribers C_{ij} using (t, n) Shamir secret sharing, where:

$$s^* \longrightarrow \{s_1^*, s_2^*, s_3^*, \dots, s_n^*\}$$

To briefly recall Shamir secret sharing computation, let consider the secret to share s^* . We divide the latter to pieces s_i^* and we pick a random $(t-1)$ degree polynomial $f(x) = a_0 + a_1x + \dots + a_{t-1}x^{t-1}$ in which $a_0 = s^*$,

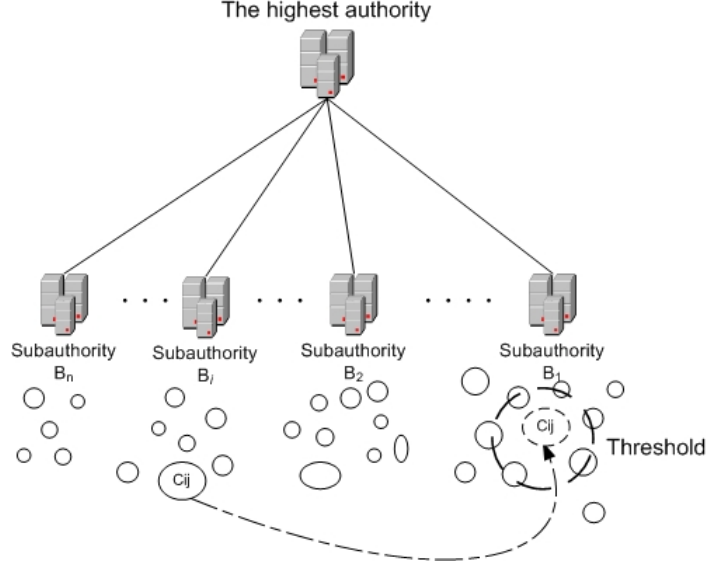


Fig. 6.4: The authentication process

$s_1^* = f(1)$, $s_i^* = f(i)...$, $s_n^* = f(n)$. Given any subset t of s_i^* values, we can find the coefficients of $f(x)$ by interpolation. We can then evaluate $s^* = f(0)$. The knowledge of $t - 1$ of these values is not sufficient to compute s^* .

Hence B_i provides in a secure way each subscriber C_{ij} with the tuple (r_{B_i}, s_j^*) where s_j^* is the share of subscriber j in the network of B_i and named C_{ij} . These shares allow any threshold t from the n subscribers to perform all functions of the certificate authority (B_i) (renewing, revocation, certifying, authentication, etc) by using a simple threshold Schnorr signature scheme with respect to the level of delegation warrant m_w^* . The new warrant specifies carefully the circumstances and conditions of this level of delegation. This second delegation allows the authentication of the new coming nodes in the network without intervention of the local sub-authority (agency) or the home network of the visitor node (see Fig 4 for a global view of the network). The following table shows the possible exchanged messages of a mobile node coming from B_i 's agency network to B_{n-i} 's agency network.

Tab. 6.1: Exchanged messages.

Message types	Messages contains
Authentication Request	certificate, signed B_i 's warrant authentication-request
Renew certificate	Certificate, signed B_i 's warrant Renew-Request
Threshold answers	partial signatures, signed B_{n-i} 's warrant, a signed threshold warrant, success message
Infrastructure answer	full signature, hight authority warrant, success

6.5 Scheme Evaluation

6.5.1 Security Evaluation

The partial signature delegation with warrant can be applied to any discrete logarithm problem (DLP) based on the signature scheme [70]. As an example for our construction, we use the Schnorr's signature scheme for both standard signing and proxy signing. The Schnorr and proxy Schnorr signature are proved to be secure in random-oracle models, under the assumption of hardness of the DLP. The security properties, satisfied in each level of delegation in our scheme, are:

Verifiability: The warrant of the first level of delegation, m_w , allows any verifier to check if the proxy signer B_i is allowed to perform a proxy signature on the behalf of A . It can also verify if the B_i signature matches the terms of the signing delegation in the warrant.

At the second level of delegation, the verifier can be convinced of B_i approval claimed in the signed message or on the certificate. Thus, any verifier can easily check from the warrant m_w^* if the threshold of the proxy signers is allowed to perform a proxy signature on behalf of B_i or not. It can also verify if the threshold signature matches the terms of this level of signing delegation, as described in the warrant.

Strong unforgeability: The highest authority A and a third party, who are not designated as proxy signers, cannot impersonate B_i to create a valid proxy signature. Since B_i combines its private key with m_w to create its proxy signing private key, no one can create the proxy signing key except B_i . We assume also that the highest authority A verifies that each proxy signer B_i possesses the private key corresponding to its registered public key to avoid public key substitution attacks. For this purpose, B_i does the same

with its subscribers.

At the second level of delegation the unforgeability propriety is verified, because only a coalition of " t " subscribers can create a valid proxy signature (i.e certificate), therefore a corrupted proxy signer $t - 1$ cannot create a valid proxy signature. It is also assumed that t or more subscribers cannot be corrupted without being detected. However, the second level of delegation does not verify the unforgeability propriety against B_i . Since B_i permits this level of delegation to be more flexible and successful in the authentication process, there is no interest for the latter to forge the signature of its threshold of subscribers. On the other hand, the protection of the threshold proxy signature against a possible forgery by B_i involves many complications. Thus, the visitor authentication across networks becomes impossible unless the nodes of each first responder agency store the public keys of the nodes of the others agencies. Moreover, it is not important to the verifier in the visited network that the visitor's certificate was signed by B_i or the B_i 's threshold proxy signers.

Strong identifiability: Anyone can determine the identity of the corresponding proxy signer from a proxy signature. Since we assume that A 's public key is available to all parties and A signs the warrant that contains A 's identity, B_i 's identity and B_i 's public key, it is easy to determine the identity of the proxy signer by transition in both levels .

Strong undeniability: Any proxy signer of A cannot repudiate a signature it has previously created. Since none can create the proxy signature that B_i creates on behalf of A , B_i can not repudiate its proxy signature. However, for the second level delegation, the threshold signature is considered as the B_i proxy signature. Hence, B_i is responsible of any threshold signature. This is not a big concern since the purpose of the signature is to provide a certificate for across networks authentication. We do not even distinguish if the certificate was signed by B_i or by the threshold, indeed the authentication across networks continues to be assured. Consequently, given the purpose of this second level signature delegation, namely only authentication and certificate renewing, there is no security issue of who performs this operation.

6.5.2 Efficiency Evaluation

The common approaches of certificate authority (CA) such as centralized, hierarchical, replication and partially distributed CA are not well adapted to the first responder mobile networks context.

Indeed, when a unique centralized CA [6] is used for several independent

networks, the CA intervenes as a trusted third party in each communication. This requires the universal knowledge of the CA's public key. This approach is cheaper for authentication partners, nevertheless suffers from a strong scalability problem. Furthermore, the CA represents a single point of failure, since it can be targeted by DoS attacks. Replicating the centralized CA is costly and risky, because if one replicated CA is corrupted, the all network security breaks down. For the previous reasons centralized CA is definitely not the best choice for first responder use.

Recently, the department of homeland security used a hierarchical CA to provide first responder's access control cards with credential to allow interregional first responder to access governmental building in disaster situation [54]. The hierarchical CA can have several organization forms like high authority subordinating sub-authorities [59], clustered [12] or in web of trust[26]. The hierarchical CA provides the mobile node with a cross signed certificate, therefore the mobile node must transport the public of CA and sub-authorities in the hierarchy. Thus, the mobile node performs a set of signature verification equivalent to the number of the hierarchy levels or the chain of trust to authenticate the communication partner. Moreover, the mobile node must carry a set of all the public keys of the sub-authorities in the hierarchy or the chain of trust. This approach also keeps high level of centralization and requests CA availability, which is no guaranteed in mobile networks. It suffers also from scalability problems.

The partially distributed CA [82] approach distributes the certification authority among certain specialized nodes in the network. Therefore, they have to act collectively for the authentication process. This approach cannot apply to first responders because it does not support across networks authentication, since the entire network node must be either known in advance to the CA before the network deployment or get bootstrapped manually with a new credential by the CA before visiting the network. The deployment of this approach presents many concerns because mobile network cannot guarantee the availability of the specialized nodes in case of attacks and network partition.

Our approach overcomes the reachability, the availability, the scalability and the one point failure limitation using the two levels of certification delegation. It permits successful authentication in hard network conditions in infrastructure mode as well as in ad hoc modes with a threshold of neighbors or first responder agencies. Differently from others approaches, our scheme manages online new coming and leaving nodes. These advantages cost the verifier less than two signature verifications (verification of the first level delegation warrant and the certificate signed by the home agency), because the

proxy schemes allow one exponent saving in advance. However, this costs to the authenticating node the verification of the first level delegation warrant, second level delegation warrant and the signature of the obtained certificate performed by the threshold from the visited network. This is equivalent to less than 3 signature verifications (i.e., equivalent to 6 exponentiation when Schnorr proxy signature is used).

Consider now our two-level delegation scheme and the hierarchical certification ones to compare the delay for authentication process. We analyze the case of 10 sub-authorities in which the mobile node must be able to move, to authenticate as well as to renew its certificate. Hierarchical schemes require sub-authorities to possess the public keys of each other. Not only, the mobile node must also store the public keys of the different sub-authorities in the hierarchy. By contrast, our scheme, does not require none of these elements, namely, neither the previous knowledge between sub-authorities, nor the storage by mobile nodes of any information about sub-authorities toward which they may move. The only required knowledge is the universal knowledge of the highest authority A public key. Our scheme allows mobile

Tab. 6.2: Authentication cost comparison in infrastructure and threshold mode for two-level delegation scheme and hierarchical scheme based on the values of [25] on P3-977MHZ device.

Parameters	Two level delegation delay in msec 10 subauthorities		Hierarchical schemes delay in msec 10 subauthorities	
	512 bits	2048 bits	512 bits	2048 bits
Key size	512 bits	2048 bits	512 bits	2048 bits
Cost for authenticator	$\leq 2 \times 1.940$ ms	$\leq 2 \times 15.030$ ms	10×1.940 ms	10×15.030 ms
authentication cost for MN with Threshold	$\leq 1,940$ ms $2 \times 1,940$ ms	≤ 15.760 ms + 2×15.030 ms	NA	NA
authentication cost for MN in infrastructure mode	$\leq 2 \times 1.90$ ms	$\leq 2 \times 15.030$ ms	10×1.940 ms	10×15.030 ms

nodes to authenticate and to renew the certificate either in ad hoc mode with a threshold of neighbors or in infrastructure mode with the visited network authority. From the point of view of the authenticator, in this example, the cost to verify the mobile node certificate consists of two Schnorr signature

verifications (see Table 6.2). While, if we consider the mobile node, the renewing of its certificate at the threshold of $t = 3$ costs the verification of one threshold plus two signatures.

On the other hand, in the case of hierarchical schemes, only infrastructure and peer to peer mode authentication are allowed. Thus, in our example, the authenticator computes ten Schnorr signatures verifications. Since, the certification and the certificate renewing are only allowed in the infrastructure mode with the sub-authorities of the hierarchy, this costs ten signature verifications to the authenticating node as showed in table 6.2.

6.6 Related Work

The first proxy signature scheme was introduced by Mambo et al. with several construction in [44] [45], and was into full delegation, partial delegation and delegation by warrant schemes. Later on, Kim et al. in [70] proposed new schemes and some improvements. They combine the advantages of the partial delegation and the delegation by warrant in the partial delegation by warrant scheme, and proposed a threshold proxy signature scheme, where an originator delegates its signing ability to a group of n proxy signers such that only the collaboration of t or more proxy signers can create the proxy signature. In 1999, H-M Sun et al. show the vulnerability of [44]. In fact, the proxy signer can transfer the right to create a proxy signature to others. Furthermore, they show the sensitivity of [70] to proxy signer public key substitution, something that instead we avoid in our construction. Recently, Lee et al. described in [19] the proprieties of a strong proxy signature scheme. They also create a strong proxy signature scheme that they apply for the construction of a multiproxy signature scheme where multiple original signers delegate their ability to sign to one proxy signer. Later Lee et al. apply the concept of non designated proxy signature introduced in [19] to a mobile agent [18]. In this scheme, designed for electronic commerce, the original signer does not specify the identity of the proxy signer in the warrant. Accordingly, this allows any possible proxy signer to sign on behalf of the originator. Moreover, Lee et al. apply the strong non designated proxy signature for multi-proxy mobile agent where plural customers delegate their signing capability to a mobile agent. In [33] Cheon et al. analyze the security importance of the secure channels for warrant delivery to the proxy signer. Girault proposes auto-proxy schemes for self delegation [27], while Park et al. published a nominative proxy signature scheme for mobile communication [29] where only the nominee can check the correctness of the proxy signature and prove its validity to third party when needed.

Recently Bao et al. proposed the first identity based threshold proxy signature scheme with known signers from bilinear pairings [10], while Lu et al. published the first identity based one-time proxy signature with formal security proofs [39].

Differently from existing schemes, we use the concept of proxy signature to solve the very challenging issue of roaming across different administrative networks in real situation. We also combine and adapt different types of proxy signatures to guarantee an efficient decentralization of the authentication decision.

6.7 Conclusion

This chapter presents an authentication solution that endows wireless mobile networks like first responder networks, with a desirable non-centralized certifying authority via two level delegation of signing capability. Our scheme permits the local authentication of mobile nodes in both infrastructure and ad hoc mode, either by the first responder agencies or a by threshold signer. The scheme allows high degree of freedom of roaming to mobile nodes across different first responder networks. Moreover, our scheme allows each first responder agency to define its own threshold of delegation depending on the level of security needed, without harming the security of the other first responder networks. These characteristics give to the scheme high level of flexibility and efficiency while keeping strong security requirements. A future possible extension of this work can generalize the scheme to include other networks exhibiting similar behavior like mesh networks.

7. SIMULATION RESULTS

7.1 *Introduction*

This section presents the simulation results of mGAP protocol in TOSSIM environment and MICA2 platform. The choice of TOSSIM as a simulator is motivated by its simplicity and flexibility in designing topologies. Moreover, microelectronic lab of UCL makes available the MICA2 hardware allowing real implementation. This encourages our choice and feeds our expectations to perform some comparisons with proxy authentication. Our simulation are concerned especially about the performances of mGAP but they also cover some performances aspects, treated in others chapters in this thesis.

7.2 *TOSSIM Simulator and TinyOS*

TinyOS is an open source operating system, designed for wireless embedded low power device networks. The architecture of TinyOS is based on components which enable rapid renovation and implementation while minimizing code size. This architecture takes in consideration different constraints of lower power devices. Moreover, TinyOS possesses a rich library that contains network protocols, distributed services, sensor drivers, and data acquisition tools. TinyOS owes also an event driven execution model, which allows a good management of unpredictability in wireless communication and physical interfaces through scheduling operations.

TOSSIM is a discrete event simulator for TinyOS low power device networks. It permits users to compile TinyOS application into the TOSSIM framework (run on PC) instead to compile it for mote (hardware nodes). TOSSIM runs on PC allowing users to debug, test, and analyze algorithms in controlled and repeatable environment. Moreover, it permits experiment application, protocols, and configurations with hundreds of nodes which is expensive and not easy to do with motes. TOSSIM possesses an extensive external communication system, so testers can monitor packets being transmitted as well as dynamically inject packets into the simulated network. Finally, TOSSIM allows also grained configuration of debugging output in

runtime.

7.3 Simulated Network Topology

We programmed two network topologies to make comparative analysis of authentication operations, namely when it uses a mGAP protocol and when it uses a proxy authentication. Each network possesses one authentication server (node 0, node 1), three access points (nodes 2 to 7); and to each access point is connected three nodes from the group nodes 8 to 31). During the simulation we use TinyViz the Java visualization and actuation environment for TOSSIM (see Figure 7.1. TinyViz is more than a visualizer, it is a framework in which pluggins can provide desired functionalities. By itself, TinyViz does not provide a multitude of functions but it performs the drawing of motes and their LEDs. It also includes a few examples and topology layouts.

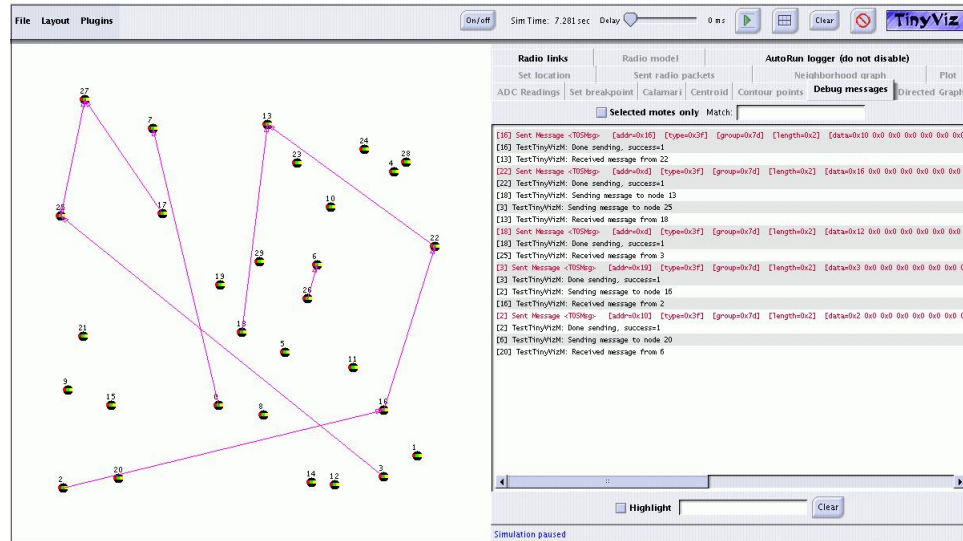


Fig. 7.1: TinyViz framework

7.3.1 Metrics and Mobility

The simulator is composed of two neighbor networks. When the simulator is power on, the nodes broadcast requests toward the access points (AP) in the neighborhood. Each node connects to an AP from its network. The

mobility of the AP is simulated via a table of visibility. Only the server of the network to which the AP is connected is visible to the AP in the network. Only the APs of the network are visible to the nodes of the same network. The mobile AP has in its visibility table the server of the neighbor network. When the quality of the link to the home server decrease, the mobile AP sends an authentication request toward the server of the neighbor network. The mobile AP becomes a part of the neighbor network, and the nodes of the mobile AP remain connected to the latter. The mobile AP becomes non visible to the nodes of the old network.

Tab. 7.1: Simulation metrics description

Simulation metrics	
Success ratio	The ration between the average of successful authentication and the average of total authentication request(failure+success) measured by percentage from 0 to 100%
Latency	Time spend by the Mobile AP point to execute successful authentication protocol. The latency is measured as the average latency of successful authentication within each run and for 20 runs. Latency is measured in milliseconds (ms)
Timeout	Is the maximum life time considered by the simulator for each message of the authentication. The Timeout is measured in (ms) fixed between 50 ms to 1400 ms
Load on the server	The load in the queue of the authentication, this load fix the period during which the server is busy and unable to process the coming request; the queue is fixed between 0 ms to 1400
Number of node in the network	Represents the totals number nodes in the network. The number from 32 to 250 nodes

7.3.2 Authentication Using mGAP and Proxy scheme

For this simulation we choose the random topology layout provided by TinyViz. Initially, all the 32 nodes in the network are not connected and they can pass through 3 states : connected, disconnected and connection

in progress. When the simulator is switched on, a discovery step starts, and the connection establishments among the node following our rules of visibility (whom can connect to whom) are active. The access point AP_2 is connected to the server AS_0 while the nodes attached to AP_2 are connected to the server through AP_2 . When the link quality to the AS_0 decreases, the AP_2 changes its network and request authentication using the mGAP protocol to the sever AS_1 of the neighbor network. This roaming operation across networks can be influenced by many factors. For our simulation we chose to measure the latency and the success ratio of such operation (across network authentication) under some environment conditions. The environment elements that our simulation consider for measurements are described in Table 7.1:

The mobile AP that tries to connect to a network, searches firstly for an authorized destination. The authorized destination was simulated via a table of visibility where access points can see only servers to which they are allowed to connect. To one of these servers the mobile AP sends an authentication request. If the selected server accepts the request, the mGAP authentication exchange starts and sequentially, the latency computation of mGAP authentication takes place.

In case of proxy authentication, the request sent by the mobile AP must transit by the local authentication server to reach the remote home server. The remote home server authenticates by proxy the mobile AP. The only visible server to the mobile AP when in the visited network is the local authentication sever. However the latter is not allowed to authenticate the mobile AP. The local server forewords the authentication request to another server in the home network of the mobile AP. The visibility table is used to manage the selection of the distant server for the proxy authentication. The latency computation starts when the authentication by proxy start via the local server.

In both cases, when the mobile access point is authenticated, it continues to send to the server the packets of the mobile nodes that access the network resources through it. The packets' maximum size, used in this network, is 127 byte, the nonce size is 20 bytes, symmetric encryption key AES-128 bits, message authentication code MAC size is 20 bytes, 160 bit ECC public key.

7.3.3 The results

The previous simulation of mGAP and proxy authentication was performed for different timeouts of the authentication request, different load on authentication servers, and for different number of nodes in the network. The

simulation objective is to study the impact of the listed metrics on an authentication process using as criteria of the authentication latency and authentication ratio of success. Finally, the results of the simulation obtained using mGAP are compared with the results of a simulation where a proxy authentication scheme is used.

7.3.4 Timeout Impact

From the point of view of the authentication requester, timeout is the time attributed for each execution of authentication of the protocol by authentication requester, after which the request is considered as lost or dropped. Then the requester renews his request.

The simulation result (see Figure 7.2) shows that the rate of authentication success increases when the timeout increases. Thus, when the timeout is less than 200ms the success of authentication is less than 20%. However, when the timeout of authentication request is more than 1200 ms the success ratio reach 100%. The ratio of success is low because the authentication server has no enough time to process all the authentication requests before the expiration of the request (timeout). Therefore, the failure rate of authentication request is very high. However, when the expiration time (timeout) of the authentication request is more than 1200 ms the success ratio reaches 100% because the server has enough time to process all the authentication requests.

Consider now the proxy authentication scheme. The analysis of timeout impact on this scheme shows that ratio of success of the proxy authentication is very low when the timeout is very small (see Figure 7.2). This happens because most of the authentication requests are considered by the requester as failed or dropped before the proxy has the time to answer the request. The results show that higher the timeout, higher is the success ratio of proxy authentication. However, the success ratio become less sensitive to the timeout size around 1200 ms. Therefore, we can conclude that another factor related to distant authentication limits the success ratio around 60%

The comparison of the authentication success ratio of mGAP and the proxy scheme shows better performance for mGAP. The success ratio reach 100% for mgap at 1200 ms timout; however capped at 60% for the proxy scheme. This because mGAP uses only fewer one-hop messages. However, the proxy scheme uses more messages,that are exchanged across networks.

On the other hand, the authentication latency is high, namely around 800 ms for low timeouts. This because the latency counter starts from the first request until the end of the successful attempt. Thus, the counter includes

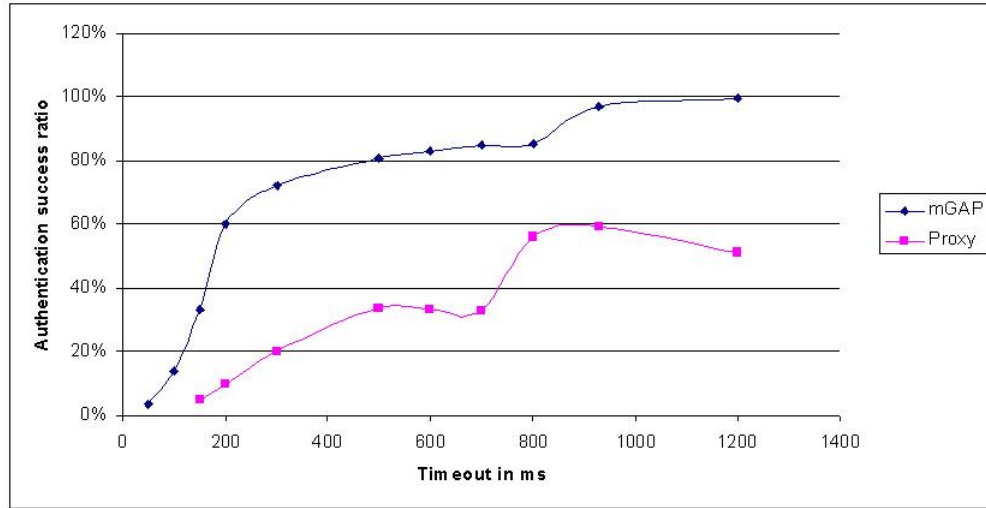


Fig. 7.2: Authentication latency for different flows' timeout

the time wasted for unsuccessful attempt. The low timeout reduces the chances of successful attempts, which increases the authentication latency. However, the high timeout increases the chance of a successful attempt and reduces the number of repeated attempts, which decreases drastically the authentication latency (see Figure 7.3).

As far as it concerns latency when authentication is by proxy, this is also influenced by the timeout attributed to the authentication request. The results of our simulator (see 7.3) shows high latency when the timeout is very low because the authentication latency is computed as an average of the delay between the request until the success. Thus, the latency of each successful authentication includes the latency of all intermediary failed attempts. Therefore, the low timeout induces high number of failure of authentication request then high latency. From the result of the simulator, we notice that higher the timeout, lower are the failed authentication attempts, thus lower is the authentication latency. This is true until the timeout does not influence any more the authentication latency, but other factors like the traffic on the path of the proxy authenticator can continue to do so.

The comparison between the authentication using mGAP and proxy scheme shows a clear advantage of mGAP. The latency at low load is 70% higher for proxy authentication than for mGAP authentication. At high load on the authentication server the proxy authentication latency is five times

higher than for mGAP.

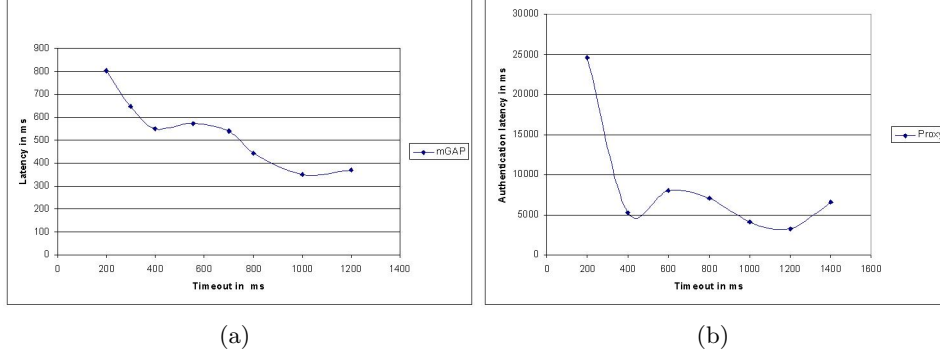


Fig. 7.3: Authentication latency as the timeout of the authentication flows increases for mGAP(a) and proxy scheme (b). The delay of each flows is averaged over 20 simulation runs

7.3.5 Load Impact

We consider the load on the authentication server as another factor that influences the performance of the authentication process. In our simulation, we added to the authentication server an increasing load to observe the impact on the performances of mGAP and the proxy scheme. The load is added to the normal load due to processed requests from other nodes in the network.

The result shows that the success ratio of authentication requests decreases dramatically with the increasing load on the authentication server (see Figure 7.4). The initial value of the load represented in the graphic by zero is not a null value, but it represents the normal load of a network in normal conditions (out of the intensive use of the network). To have non bias results, we fixed very high timeouts for each flow, to avoid the mixing dropped request due the request expiration with dropped request due to the server loads.

When the authentication is performed by proxy to a distant server, the ratio of success of the authentication process depend strongly to degree of availability of the distant server and the condition of the network on the path of the proxy authentication request. We performed the measurement for proxy authentication request, when different loads are applied to the proxy server. The result in Figure 7.4 show a decreasing success ratio when

the load on the proxy authentication server increases. At the beginning the success ratio is low because when we power on for the first time nodes take time to found their related server and access points.

We can notice that the best success ratio for the proxy authentication is 60% and it coincides to the server load of 100. This is different from mGAP based authentication which reaches almost 100% success for low loads (see Figure 7.4).

It is important to notify that during the measurement we considered a big timeout (2000 ms) for the authentication request. Thus, the measured success ratio was not influenced by any factor else than the load on the server.

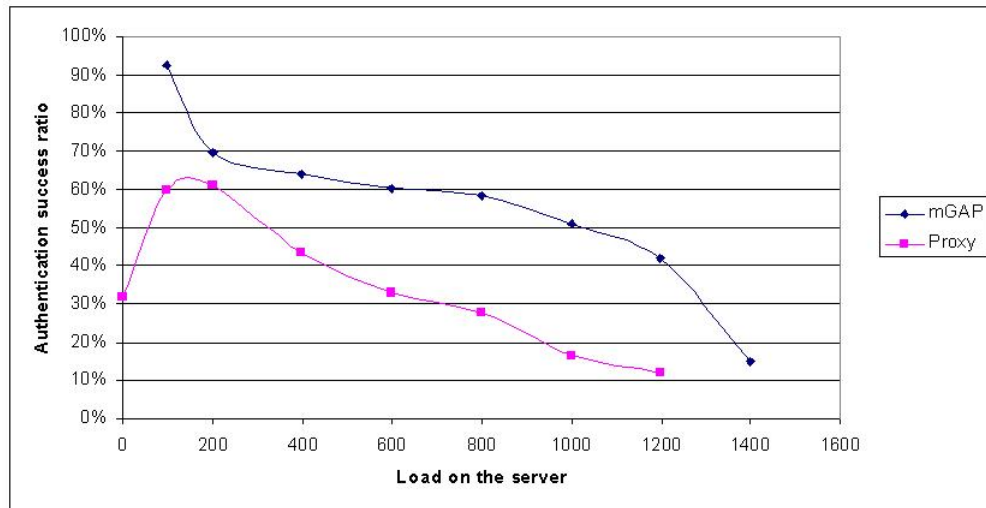


Fig. 7.4: Success ratio of authentication for different load at the server

The load on the server obviously has influence on the latency of the authentication process. To measure this impact we choose a high timeout for the authentication flows to prevent overlapping of timeout and load impact on the authentication latency. The authentication latency increases in a non linear way with the increased load. This non linearity is due to an event taking place in the network like for example the leave of a node and/or the change of the access point of a node to the network resources. Some surprising values appear for high load on the server, namely, the low value of authentication latency at 1400 load. This is due to the fact that some authentication requests are received at the moment where the server is at

the end of its period of inactivity (the server processed all the requests and operation in its queue). Consequently, the last authentication requests in the queue are treated quickly leading to the abnormal low latency even though the load is high. Remember that the low latency value coincides with a very low ratio of success of authentication (see Figure 7.4).

The measurements of authentication latency using the proxy authentication scheme shows an increasing latency with the increasing load on the server (see Figure 7.5). Differently from the local authentication with mGAP, the $\text{\textcircled{R}}$ proxy authentication latency starts at a high value (2000 ms) even for low loads at the server. By contrast, the with a mGAP server, the latency starts at a low value (400ms).

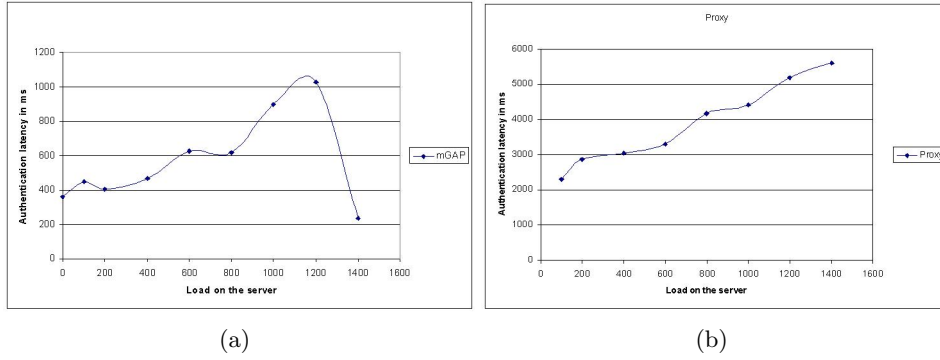


Fig. 7.5: Authentication latency increase as the load on the authentication server increase for both mGAP (a) Proxy scheme (b). The latency of each flows is averaged over 20 simulation runs

7.3.6 Impact of the number of nodes

The number of nodes in the wireless networks has a strong impact on the network performance because the limited resources of such networks are divided among the nodes in the network. Our simulator measures the success ratio and the latency of the authentication process, for both mGAP and proxy scheme, as the number of nodes increases.

The mGAP authentication ratio of success decreases with the increasing number of nodes in the network (see Figure 7.6) because the wireless network channel and infrastructure (access points servers) become increasingly shared among nodes. Thus, high traffic is generated in the network leading to longer periods of listening to the radio channel and a bigger number of

dropped packets.

In the same way and for the same reasons, the success ratio of proxy authentication decreases with increasing number of nodes in the network (see Figure 7.6). However, the success ratio decreases more quickly for the proxy authentication. This is due to the fact that the proxy authentication exchanges locally and across networks more messages than mGAP. This increases the number of undelivered and dropped packets. Thus the success ratio is affected when the network resources decreases.

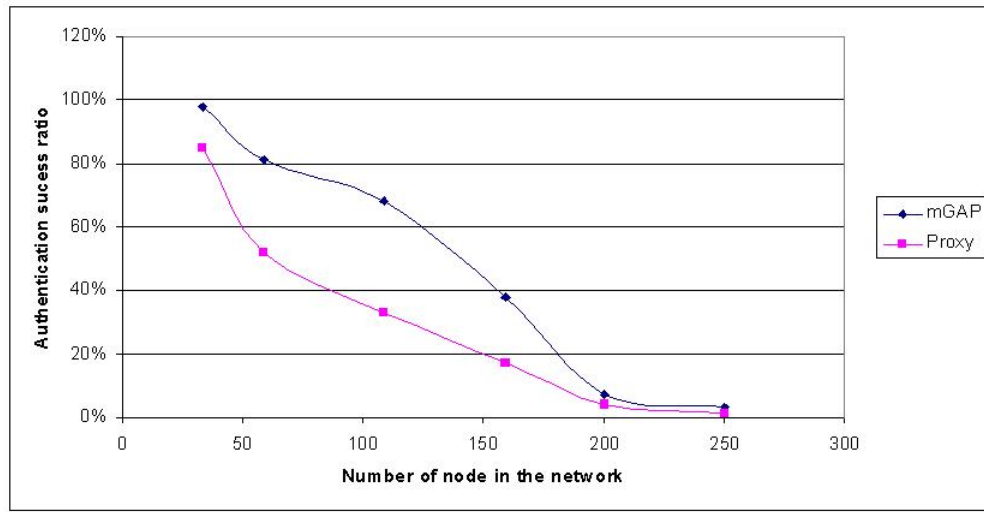


Fig. 7.6: Success ratio of mGAP authentication as the number of nodes in the network increases

For the same reason as explained above, higher the number of nodes in the network, higher the authentication latency of the mGAP protocol. The Figure 7.7 (a) shows the increasing of the authentication latency due to the distribution of the network resources and radio link among increasing number of nodes. The distribution of the wireless channel among increasing number of nodes increases the lessening time of the shared radio channel. In general the latency increases because of the load on the network and depends on the number of messages exchanged by mGAP and the proxy scheme. mGAP exchanges less messages than the proxy scheme. Thus it request less time on the radio channel, so the authentication latency is less than for proxy scheme

Differently, the latency of proxy authentication increases dramatically

when the number of nodes in the network increases (see Figure 7.7 (b)). The latency of the proxy scheme increases even faster when the resources in the visited network are divided among more nodes. This slows down the local operation in the visited network as well as the forwarding of the authentication request to the proxy sever. There are quite a lot failed attempts of authentication before the success. Thus, the latency of each successful authentication latency equals the average of the latency of several failed attempts plus the latency of the successful authentication. It is important to notice that for these simulations we have chosen high timeout.

When the network resources and radio channel is shared among increasing number of nodes; the latency of authentication process increases for both mGAP and proxy scheme. However, the latency of proxy authentication increase dramatically faster starting from 50 nodes because of the load on the network and the higher number of exchanges messages in the proxy scheme.

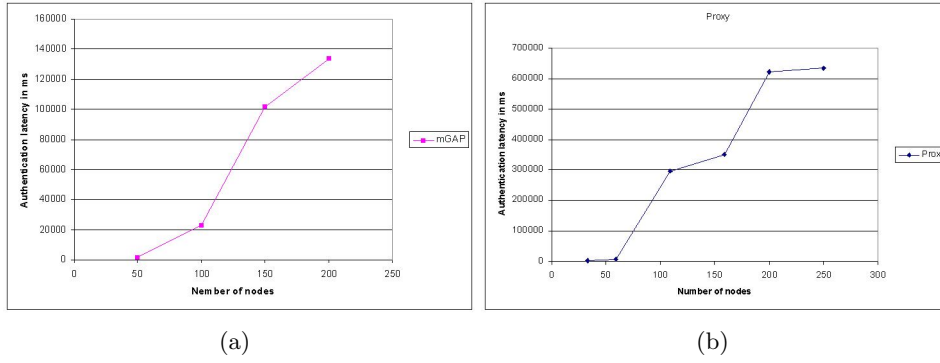


Fig. 7.7: Authentication latency as the number of nodes in the network increases. The latency is averaged over 20 simulation runs

7.4 Conclusion

As claimed in the introduction this section was dedicated to some simulation in TOSSIM environment. The simulation was performed for different timeouts of authentication request, for different loads on the authentication server and for different number of nodes. The impact of these conditions in the network was captured by the latency and ratio of success. Furthermore, the results obtained using mGAP were compared to those obtained using the proxy scheme. The simulation results clearly show the advantages of mGAP

against the existing classical proxy authentication. The authentication process with mGAP need fewer exchanges than the proxy authentication and do not request across network exchanges like the proxy scheme. Thus, mGAP performs the authentication with less latency and keeps better authentication success ratio than the proxy authentication. The comparison considers deferents timeout, deferents loads on the authentication server and different network's sizes.

8. CONCLUSION

As promised in the title, this PhD thesis addresses some authentication issues in wireless networks to achieve seamless mobility. First we introduced and analyzed the authentication issue as one of the most important security features in the context of mobility within wireless networks.

Chapter 2 investigated and classified different authentication protocols and standards for wireless networks. By doing so, we discovered several weaknesses in existing authentication solutions. Moreover, we identified several unsatisfied authentication needs. Our analysis allowed the classification into families of the authentication protocols and showed the possibility of new family combination for new authentication solutions. We also developed a generic authentication model for wireless networks, which we respect in the corresponding authentication protocol design. Our authentication solutions consider several wireless networks architectures as well as different mobility contexts. They are developed for WLAN, mobile group and mobile networks and they respect the constraints on computation and energy sources of wireless network components, especially at the mobile node side. We also manage and develop authentication architectures for decentralized authentication process.

In chapter 3 we developed two protocols for fast authentication in WLAN to keep ingoing session and avoid dependence of re-authentication success on mobile node speed, network load, and the authentication server loads. We also provided fast light solution for recurrent visitor re-authentication.

For the mobility across network and group mobility, we developed in chapter 4 an authentication architecture based on profiling to predict mobility destination of mobile nodes and groups. We also designed mGAP an authentication protocol for across domain (networks) authentication. The profiling architecture and mGAP provide the mobile entity with one credential valid at several roaming destinations.

The research accomplished in this thesis leads us to believe that it is impossible to create an optimized generic solution for all wireless networks and mobility contexts. Consequently, keeping in mind the possibility of cohabitation and complementarity among protocols, chapter 5 combines the

solution of mGAP and that of the authentication ticketing, to build a new protocol called Imbricated. This scheme provides an authentication solution for within and across networks mobility.

Chapter 6 treats another mobility context to provide an authentication solution for mobile networks like first responder networks. Our solution allows a secure collaboration and resource sharing among different mobile wireless networks. The latter can keep collaborating while keeping their own security level without forcing the network partners to adopt the same. Our solution based on two levels of signature delegation allows secure collaboration among mobile networks, which do not have previous relation or collaboration agreements. Moreover, our solution enlarge roaming freedom of individual nodes as well of mobile group without increasing the requirement of storage space neither computation overhead nor architecture infrastructure.

Chapter 7 is dedicated to simulations in TOSSIM environment. These simulations were performed for different timeouts of authentication request, for different loads on the authentication server and for different numbers of nodes. The impact of these conditions in the network was captured by the latency and ratio of success. Furthermore, the results obtained using mGAP were compared to those obtained using the proxy scheme. The simulation results clearly show the advantages of mGAP against the existing classical proxy authentication.

Research on the authentication of entities needs a global thinking concerning the environment parameters like the capacities of involved nodes, application type, radio channels constraints, level of security needed and the mobility context. These parameters are all considered during the design of our solutions. Indeed we covered several situation of mobility in wireless network that can be observed in the network of the same organization. Hence, we believe that our solution can be combined or deployed for other wireless networks exhibiting similar needs and environments such as mesh networks.

Finally the anonymity and privacy are two security concepts that were changed by the mobility. Even if this work was not dedicated to the anonymity and privacy issues in mobility within wireless network, we are aware that they are an increasing concern among mobile users about anonymity and privacy. Thus anonymity and privacy have become a hot topic of research where more creativity is needed. This aspect of security is not studied in this thesis, but it is one of the extension perspectives of the solutions provided in the previous chapters.

BIBLIOGRAPHY

- [1] IEEE Std 802.11f-2003 IEEE Trial-Use Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation, 2003.
- [2] IEEE Std 802.11i-2004 IEEE Standard for Information technology-Telecommunications and requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 6: Medium Access Control (MAC) Security Enhancements, 2004.
- [3] 3rd Generation Partnership Project 3G TS 33.12. Technical specification group services and system aspects, 3g security. Technical Report 5, 3GPP suport, 1995. http://www.3gpp.org/ftp/tsg_sa/WG3_Security/_Specs/33120-300.pdf.
- [4] Nidal Aboudagga, Jean-Jacques Quisquater, and Mohamed Eltoweissy. Group authentication protocol for mobile networks. *wimob*, 0:28, 2007.
- [5] Nidal Aboudagga, Mohamed Refaei, Mohamed Eltoweissy, Luiz DaSilva, and Jean-Jacques Quisquater. Authentication Protocols for Ad hoc Networks: Taxonomy and Research Issues. In *Q2SWinet '05: Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*, pages 96–104, New York, NY, USA, 2005. ACM Press.
- [6] Carlisle Adams and Stephen Farrell. RFC 2510: Internet X.509 Public Key Infrastructure Certificate Management Protocols, 1999.
- [7] An Liu and Peng Ning. TinyECC: Elliptic Curve Cryptography for Sensor Networks, 2006. <http://discovery.csc.ncsu.edu/software/TinyECC>, September 2006.
- [8] Giuseppe Ateniese, Michael Steiner, and Gene Tsudik. New multiparty authentication services and key agreement protocols. *IEEE Journal on Selected Areas in Communications*, 18(4):628–639, 2000.

-
- [9] Dirk Balfanz, D. Smetters, Paul Stewart, and H. Wong. Talking to strangers: Authentication in adhoc wireless networks, february 2002. In Symposium on Network and Distributed Systems Security (NDSS '02).
 - [10] Haiyong Bao, Zhenfu Cao, and Shengbao Wang. Identity-Based Threshold Proxy Signature Scheme with Known Signers. In *TAMC*, pages 538–546. Springer Berlin / Heidelberg, 2006.
 - [11] Marc Bechler, Hans-Joachim Hof, Daniel Kraft, Frank Pahlke, and Lars Wolf. Cluster-based security architecture for ad hoc networks. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2393–2403. IEEE Press.
 - [12] Marc Bechler, Hans-Joachim Hof, Daniel Kraft, Frank Pählke, and Lars Wolf. A Cluster-Based Security Architecture for Ad Hoc Networks. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2393 – 2403, 2004.
 - [13] Nathan Blaunstein, Ran Giladi, Avraham Freedman, and Moshe Levin. Unified approach of GOS optimization for fixed wireless access. In *IEEE Transactions on Vehicular Technology*, pages 200–208. IEEE press, January 2002.
 - [14] Alexandra Boldyreva, Adriana Palacio, and Bogdan Warinschi. Secure proxy signature schemes for delegation of signing rights. Cryptology ePrint Archive, Report 2003/096, 2003. <http://eprint.iacr.org/>.
 - [15] Colin Boyd and Anish Mathuria. *Protocols for Authentication and Key Establishment*, pages 190–195. Springer, June 2003.
 - [16] Emmanuel Bresson, Olivier Chevassut, Abdelilah Essiari, and David Pointcheval. Mutual authentication and group key agreement for low-power mobile devices. In *Proc. of MWCN '03*, pages 59–62. IFIP, World Scientific Publishing, October 2003.
 - [17] Emmanuel Bresson, Olivier Chevassut, and D. Pointcheval. Group Diffie-Hellman key exchange secure against dictionary attacks. In Y. Zheng, editor, *Proc. of Asiacrypt'02*, volume 2501 of *LNCS: Lecture Notes in Computer Science*, pages 497 – 514. Springer-Verlag, December 2002.

-
- [18] Byoungcheon Lee and Heesun Kim and Kwangjo Kim. Secure Mobile Agent Using Strong Non-designated Proxy Signature. In *ACISP '01: Proceedings of the 6th Australasian Conference on Information Security and Privacy*, page 474, London, UK, 2001. Springer-Verlag.
 - [19] Byoungcheon lee and Heesun Kim and Kwangjo kim. Strong Proxy Signature and its Applications. In *CCS '2001*, pages 603–608, 2001.
 - [20] John Clark and Jeremy Jacob. A survey of authentication protocol literature. Technical Report 1.0, 1997.
 - [21] David Pointcheval and Jacques Stern. Security Arguments for Digital Signatures and Blind Signatures. *Journal of Cryptology: The Journal of the International Association for Cryptologic Research*, 13(3):361–396, 2000.
 - [22] Hongmei Deng, Anindo Mukherjee, and Dharma Agrawal. Threshold and identity-based key management and authentication for wireless ad hoc networks. In *ITCC '04: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04)*, volume 2, page 107, Washington, DC, USA, 2004. IEEE Computer Society.
 - [23] Yvo Desmedt and Kaoru Kurosawa. Special issue on cryptography and information security practical and proven zero-knowledge constant round variants, 1998.
 - [24] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, November 1976.
 - [25] Wenliang Du, Ronghua Wang, and Peng Ning. An efficient scheme for authenticating public keys in sensor networks. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 58–67, New York, NY, USA, 2005. ACM Press.
 - [26] Andrew Fritz and Jehan-François Pâris. Maille authentication - a novel protocol for distributed authentication. In *SEC*, pages 309–322, 2004.
 - [27] Marc Girault. Relaxing Tamper-Resistance Requirements for Smart Cards by Using (Auto-)Proxy Signatures. In *CARDIS '98: The International Conference on Smart Card Research and Applications*, pages 157–166, London, UK, 2000. Springer-Verlag.

-
- [28] Seongil Hahml, Yongjae Jung¹, Seunghee Yi, Yukyoung Song, Ilyoung Chong, and Kyungshik Lim. A self-organized authentication architecture in mobile ad-hoc networks. In *LNCS: Lecture Notes in Computer Science*, volume 3391/2005, pages 689–696. Springer Berlin / Heidelberg, 2005.
 - [29] Hee-Un Park and Im-Yeong Lee. A Digital Nominative Proxy Signature Scheme for Mobile Communication. In *ICICS '01: The Third International Conference on Information and Communications Security*, pages 451–455, London, UK, 2001. Springer-Verlag.
 - [30] Gunther Horn and Bart Preneel. Authentication and payment in future mobile systems. In *ESORICS '98: Proceedings of the 5th European Symposium on Research in Computer Security*, pages 277–293, London, UK, 1998. Springer-Verlag.
 - [31] Stuart Jacobs. Mobile IP public key based authentication, March 1999. Internet draft, URL:Draft-jacobs-mobileip-pki-auth-02.txt.
 - [32] Don Johnsonm, Alfred Menezes, and Scott Vanstone. The elliptic curve digital signature algorithm ECDSA. In *International Journal of Information Security*, pages 36–63. Springer Berlin/Heidelberg, August 2001.
 - [33] Jung-Yeun Lee and Jung-Hee Cheon and Seungjoo Kim. An Analysis of Proxy Signatures: Is a Secure Channel Necessary? In *Lecture Notes in Computer Science*, volume 2612, pages 68–79, San Francisco, USA, April 2003. Springer Berlin/Heidelberg.
 - [34] David Levine, Ian Akyildiz, and Mahmoud Naghshineh. The shadow cluster concept for resource allocation and call admission in atm-based wireless networks. In *MobiCom '95: Proceedings of the 1st annual international conference on Mobile computing and networking*, pages 142–150. ACM Press, 1995.
 - [35] Wei Liang and Wenye Wang. A local authentication control scheme for efficient authentication in wireless networks.
 - [36] Wei Liang and Wenye Wang. A local authentication control scheme for efficient authentication in wireless networks. In *Proc. of 60th IEEE Vehicular Technology Conference, VTC 2004-Fall*, volume 7, pages 5276–5280. IEEE press, 2004.

-
- [37] Joseph Liu, Victor Wei, and Duncan Wong. Custodian-hiding verifiable encryption. In *the 5th International Workshop on Information Security Applications*, pages 51–64. LNCS: Lecture Notes in Computer Science, Springer-Verlag, October 2004.
 - [38] Konrad Lorincz, David Malan, Thaddeus Fulford-Jones, Alan Nawoj, Antony Clavel, Victor Shnayder, and et. al. Sensor networks for emergency response: Challenges and opportunities. *IEEE Pervasive Computing*, 3(4):16–23, 2004.
 - [39] Rongxing Lu, Zhenfu Cao, and Xiaolei Dong. Efficient ID-Based One-Time Proxy Signature and Its Application in E-Cheque. In *CANS*, pages 153–167. Springer Berlin / Heidelberg, 2006.
 - [40] Haiyun Luo, Petros Zerfos, Jiejun Kong, Songwu Lu, and Lixia Zhang. Self-securing ad hoc wireless networks. In *ISCC '02: Proceedings of the Seventh International Symposium on Computers and Communications (ISCC'02)*, page 567, Washington, DC, USA, 2002. IEEE Computer Society.
 - [41] Malbourn Barton and Derek Atkins and John Lee and Sanjai Narain and Deidra Ritcherson and Kemal Tepe and Daniel Wong. Integration of IP Mobility and Security for Secure Wireless Communications. In *IEEE International Conference on Communications*, pages 1045–1049. ICC, April 2002.
 - [42] Tal Malkin, Satoshi Obana, and Moti Yung. The hierarchy of key evolving signatures and a characterization of proxy signatures. In *EUROCRYPT*, pages 306–322. Springer, 2004.
 - [43] Mario Cappiello, Andrea Floris and Luca Veltri. Mobility Amongst Heterogeneous Networks with AAA Support. In *IEEE International Conference on Communications*, volume 4, pages 2064–2069. ICC, May 2002.
 - [44] Masahiro Mambo and Keisuke Usuda and Eiji Okamoto. Proxy Signatures: Delegation of the Power to Sign Messages. In *IEICE TRANSACTIONS on Fundamentals of Electronics*, volume E79-A No.9, pages 1338–1354, september 1996.
 - [45] Masahiro Mambo and Keisuke Usuda and Eiji Okamoto. Proxy Signatures for Delegating Signing Operation. In *CCS '96: The 3rd ACM*

- conference on Computer and communications security*, pages 48–57, New York, NY, USA, 1996. ACM Press.
- [46] Alfred J. Menezes, Scott Vanstone, and Paul Van Oorschot. *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1996.
- [47] Arunesh Mishra, Minh Shin, and William Arbaugh. An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process. *SIGCOMM Comput. Commun. Rev.*, 33(2):93–102, 2003.
- [48] Arunesh Mishra, Minh Shin, Tom Clancy, and William Arbaugh. Proactive key distribution using neighbor graphs. *IEEE Wireless Communications*, 11(1):26–36, 2004.
- [49] Roger. M. Needham and Michael D. Schroeder. Authentication revisited. *SIGOPS Oper. Syst. Rev.*, 21(1):7–17, 1987.
- [50] Edith Ngai and Michael Lyu. Trust-and clustering-based authentication services in mobile ad hoc networks. In *The 24th International Conference on Distributed Computing Systems Workshops - W7: EC (ICDCSW'04)*, volume 7, pages 582–587, Washington, DC, USA, 2004. IEEE Computer Society.
- [51] Edith Ngai, Michael Lyu, and Roland Chin. An authentication service against dishonest users in mobile ad hoc networks. In *IEEE Aerospace Conference*, volume 2, pages 1275–1285. IEEE Press, 2004.
- [52] Jean-Jacques Quisquater Nidal Aboudagga, Mohamed Eltoweissy. Fast Roaming Authentication in Wireless LANs. In *2nd International Computer Engineering Conference: Engineering the Information Society, ICENCO 2006*, Cairo, Egypt, December 2006.
- [53] Nidal Aboudagga and Mohamed Refaei and Mohamed Eltoweissy and Luiz DaSilva and Jean-Jacques Quisquater. Authentication protocols for ad hoc networks: taxonomy and research issues. In *Q2SWinet '05: Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*, pages 96–104. ACM Press, 2005.
- [54] Department of Homeland Security. First responder authentication cards, 2006. http://www.dhs.gov/xfirstresp/standards/editorial_0849.shtm.

-
- [55] Sangheon Pack and Yanghee Choi.
 - [56] Sangheon Pack and Yanghee Choi. Pre-Authenticated Fast Handoff in a Public Wireless LAN Based on IEEE 802.1x Model. In *Mobile and Wireless Communications*, pages 175–182. Kluwer, B.V., 2002.
 - [57] DongGook Park, Colin Boyd, and Ed Dawson. Classification of authentication protocols: A practical approach. In *ISW '00: Proceedings of the Third International Workshop on Information Security*, pages 194–208, London, UK, 2000. Springer-Verlag.
 - [58] Charles Perkins and Pat Calhoun. Mobile IPv4 challenge/response extensions, november 2000. RFC 3012 (request for comments).
 - [59] R. Perlman. An overview of PKI trust models. In *IEEE Network*, volume 13, pages 38–43. IEEE Press, 1999.
 - [60] Adrian Perrig, Robert Szewczyk, Doug Tygar, Victor Wen, and David Culler. Spins: Security protocols for sensor networks. *Wireless Networks*, 8:521–534, 2002.
 - [61] Adrian Perrig, Doug Tygar, Dawn Song, and Ran Canetti. Efficient authentication and signing of multicast streams over lossy channels. In *SP '00: Proceedings of the 2000 IEEE Symposium on Security and Privacy*, page 56, Washington, DC, USA, 2000. IEEE Computer Society.
 - [62] Asad Pirzada and Chris McDonald. Kerberos Assisted Authentication in Mobile Ad-hoc Networks. In *Proceedings of the 27th conference on Australasian computer science*, volume 56, pages 41–46. Australian Computer Society, Inc, 2004.
 - [63] Asad Amir Pirzada and Chris McDonald. Kerberos assisted authentication in mobile ad-hoc networks. In *ACSC '04: the 27th Australasian conference on Computer science*, pages 41–46, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.
 - [64] Bart Preneel. A survey of recent developments in cryptographic algorithms for smart cards. *Comput. Networks*, 51(9):2223–2233, 2007.
 - [65] Bart Preneel and Antoon Bosselaers et al. Nessie d17 - preliminary list of realistic performance estimates.
 - [66] Mohamed Tamer Refaei, Vivek Srivastava, Luiz DaSilva, and Mohamed Eltoweissy. A reputation-based mechanism for isolating selfish nodes in

- ad hoc networks. In *MOBIQUITOUS '05: Proceedings of the The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pages 3–11, Washington, DC, USA, 2005. IEEE Computer Society.
- [67] Roger Kalden and Ingo Meirick and Michael Meyer. Wireless Interned Access Based on GPRS. In *IEEE Personal Communication*, volume 7, pages 8–18, April 2000.
- [68] Claus-Peter Schnorr. Efficient Signature Generation by Smart Cards. *J. Cryptology*, 4(3):161–174, 1991.
- [69] Seigmond Redl and Mathias Weber and Malcom Oliphant. An Introduction to GSM. Technical report, Artech House, 1995.
- [70] Seungjoo Kim and Sangjoon Park and Dongho Won. Proxy signatures, revisited. In *ICICS '97: The First International Conference on Information and Communication Security*, pages 223–232, London, UK, 1997. Springer-Verlag.
- [71] Sangho Shin, Andrea Forte, Anshuman Rawat, and Henning Schulzrinne. Reducing MAC Layer Handoff Latency in IEEE 802.11 Wireless LANs. In *MobiWac '04: Proceedings of the second international workshop on Mobility management & wireless access protocols*, pages 19–26. ACM Press, 2004.
- [72] Frank Stajano and Ross Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *the 7th International Workshop on Security Protocols*, pages 172–194, London, UK, 2000. Springer-Verlag.
- [73] Stephen Farrell and John Vollbrecht and Pat Calhoun and Leon Gommans. AAA Authorization Application Framework, August 2000. RFC 2904 (request for comments).
- [74] Brian Tung. *Kerberos: A Network Authentication System*. Addison-Wesley Longman Publishing Co, Inc, 1999.
- [75] Hector Velayos and Gunnar Karlsson. Techniques to Reduce IEEE 802.11b MAC Layer Handover Time. Technical Report 03:02, April 2003.

-
- [76] Lakshmi Venkatraman and Dharma Agrawal. A novel authentication scheme for ad hoc networks. In *Wireless Communications and Networking Conference (WCNC)*, volume 3, Chicago, IL, USA, 2000. IEEE Publication.
 - [77] Andrea Vitaletti and Gianni Palombizio. Rijndael for sensor networks: Is speed the main issue? *Electr. Notes Theor. Comput. Sci.*, 171(1):71–81, 2007.
 - [78] Andre Weimerskirch and Gilles Thonet. A distributed light-weight authentication model for ad-hoc networks. In *ICISC '01: Proceedings of the 4th International Conference Seoul on Information Security and Cryptology*, pages 341–354, London, UK, 2002. Springer-Verlag.
 - [79] Andre Weimerskirch and Dirk Westhoff. Identity certified authentication for ad-hoc networks. In *SASN '03: the 1st ACM workshop on Security of ad hoc and sensor networks*, pages 33–40, New York, NY, USA, 2003. ACM Press.
 - [80] David Wong and Anthony Chan. Mutual authentication and key exchange for low power wireless communications. In *IEEE Military Communication*, pages 39–43. IEEE press, 2001.
 - [81] Yacov Yacobi and Zahava Shmueli. On key distribution systems. In *Advances in Cryptology-Crypto'89*, pages 344–355. Springer-Verlag, 1998.
 - [82] Lidong Zhou and Zygmunt Haas. Securing ad hoc networks. *IEEE Network Magazine*, 13(6):24–30, 1999.
 - [83] Lidong Zhou and Zygmunt J. Haas. Securing Ad Hoc Networks. *IEEE Network*, 13(6):24–30, 1999.
 - [84] Sencun Zhu, Sanjeev Setia, and Sushil Jajodia. Leap: Efficient security mechanisms for large-scale distributed sensor networks. volume 2, pages 62–72, New York, NY, USA, 2003. ACM Press.
 - [85] Sencun Zhu, Shouhuai Xu, Sanjeev Setia, and Sushil Jajodia. Lhap: A lightweight hop-by-hop authentication protocol for ad-hoc networks. In *ICDCS Workshops*, pages 749–, RI, USA, 2003. IEEE Computer Society.