

Ecole Doctorale EDITE

Thèse présentée pour l'obtention du diplôme de  
DOCTEUR DE L'INSTITUT NATIONAL DES TELECOMMUNICATIONS

*Doctorat délivré conjointement par  
L'Institut National des Télécommunications et l'Université Pierre et Marie Curie - Paris 6*

Spécialité : Informatique

Par  
Pierre E. ABI-CHAR

# A DYNAMIC TRUST-BASED CONTEXT-AWARE SECURE AUTHENTICATION FRAMEWORK FOR PERVASIVE COMPUTING ENVIRONMENTS

Soutenue le 30 Mars 2010 devant le jury composé de :

Bernard COUSIN	Rapporteur	IRISA, Université de Rennes, Rennes (France)
Lionel BUNIE	Rapporteur	LIRIS, INSA Lyon, Lyon (France)
Amal El Fallah SEGHROUCHNI	Examinateur	LIP6, Université Pierre et Marie Curie, Paris (France)
Zheng YAN	Examinateur	Nokia Research Center, Helsinki (Finlande)
Bachar EL HASSAN	Examinateur	LaSTRe, Université Libanaise, Tripoli (Liban)
Abdallah M'HAMED	Examinateur	Handicom, Telecom & Management SudParis (France)
Mounir MOKHTARI	Directeur de thèse	Handicom, Telecom & Management SudParis (France)

Thèse n° 2010TELE0006

A DYNAMIC TRUST-BASED CONTEXT-AWARE SECURE  
AUTHENTICATION FRAMEWORK FOR PERVASIVE COMPUTING  
ENVIRONMENTS

A DISSERTATION  
SUBMITTED TO THE DEPARTMENT OF RESEAUX ET SERVICE DE  
TELECOMMUNICATION-TELECOM SUDPARIS  
AND THE COMMITTEE ON GRADUATE STUDIES  
OF UNIVERSITE PIERRE ET MARIE CURIE - PARIS6  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY

Pierre E. ABI-CHAR

May 2010

© Copyright by Pierre E. ABI-CHAR 2010  
All Rights Reserved

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

---

(Mounir MOKHTARI) Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

---

(Abdallah M'HAMED)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

---

(Bachar EL-HASSAN)

Approved for the University Committee on Graduate Studies.



# ACKNOWLEDGEMENTS

The completion of this dissertation represents more than just a demonstration of competence as a researcher. Rather, the process involved in its formulation and writing has provided me with an opportunity for both personal growth and development. For me, the key to success represents the desire to accomplish something, and the belief that it can be done. The latter however, could not have been possible without the support of those who have been close to me. For that reason, there are a number of persons to whom I would like to express my gratitude.

First I would like to thank my supervisor Abdallah M'HAMED for his help, guidance, advise as well as his enthusiasm and many valuable contributions to this work. Without his input, I would not have been able to complete this thesis. I am also grateful to Mounir MOKHTARI and Bachar El-HASSAN for giving me the opportunity to pursuing my Ph.D. I need to thanks them for their help and many inspiring discussions.

Also, I would thanks the rest of my thesis committee for taking time out of their busy schedule to review this thesis and also for their support and comments. Their suggestions and observations were extremely helpful throughout this thesis.

A special thanks to everyone for hosting me during my abroad stay at Telecom Sud-Paris. It was a very nice and memorable experience. I also thank all the people I met there. Studying here at INT has been a great experience, caused by both the excellent environment and also because of all the friends that have crossed my path over the many years I have been here. Thank you all for having a great time here with you.

I would also like to express my thanks to my two brothers and to my sister. I want to express my gratitude to them for their love and support, and for bearing with me during

those periods when work took a very big part of my time. Without your support, encouragement, guidance, I never would have made it through the whole Ph.D. process. Thank you all.

Last, but not least, I would like to present all my respect to one very special person; my Mother, NADIA ABI-CHAR. Many thanks Mom; for all your understanding, your patience and most of all, for your love. You have given me the strength to pursue my dreams, the courage to stand up for my beliefs, and the confidence that I need to succeed. Love You Mom.

Finally, this dissertation is dedicated to the loving memory of my father Emile N. ABI-CHAR. I want to thank him for his constant care and unwavering belief in my abilities. Dad, thanks for showing me what it really means to care, and what can I achieve. Thanks for trusting me, accepting me, and supporting me. You were a man of impeccable ethos. You were also looking forward to the completion of this dissertation. Unfortunately, He is not here with us to see the whole thing finished. However, his unexpected and tragic death opened my eyes to life and helped me re-estimate and re-value my own life. I am sure he is somewhere out there always watching us closely. To You Dad, I Owe So Many Thanks.

# PUBLICATIONS RESULTING FROM THIS THESIS

- P. ABI-CHAR, M. Mokhtari, A. Mhamed and B. EL-Hassan, A Flexible Privacy and Trust Context-Aware Secure Framework. Presented to the 8th International Conference on Smart Homes and Health Telematics. (Accepted To Appear).

- P.ABI-CHAR, M. MoKhtari, A Mhamed and B. EL-Hassan, A Dynamic Trust-based Context-Aware Authentication Framework With Privacy Preserving, In Proceeding of the International Journal of Computer and Network Security, IJCNS, Vol. 2, No. 2, pp.87-102, 2010.

- P. ABI-CHAR, A. Mhamed, B. EL-Hassan and M. Mokhtari Controlling Trust and Privacy in Context-Aware Environments, State of Art and Future Directions In Proceeding of the IGI Publishing under Book Title: Trust Modeling and management in Digital Environments: From Social Concept to System Development. Book Edited by Nokia Research Center, Finland, 2009, pp 352-377, 2010.

- P.ABI-CHAR, M. MoKhtari, A Mhamed and B. EL-Hassan, Secure Authenticated and Key Agreement Protocols With Access Control for Mobile Environments, In Proceeding of International Journal of Computer Science and Information Security, IJCSIS, Vol. 6 No. 2, pp. 170-183, 2009, **Best Paper Award**.

- P. ABI-CHAR, M. Mokhtari, A. Mhamed and B. EL -Hassan, Towards a Robust Privacy and Anonymity Preserving Architecture for Ubiquitous Computing, in Proc. of the Third International Conference on Risks and Security of Internet and Systems. Tozeur, Tunisia, IEEE Computer Society Press, October 28-30, 2008, pp. 125-132.

- P. ABI-CHAR, A. Mhamed, B. EL -Hassan, A Secure Authenticated Key Agreement Protocol For Wireless Security, in Proc. of the Third International Symposium on Information Assurance and Security IAS2007, Manchester, United Kingdom, IEEE Computer Society Press, August 2007, 29-31 pp. 33-38.

- P. ABI-CHAR, A. Mhamed, B. EL-Hassan, A Fast and Secure Elliptic Curve Based Authenticated Key Agreement Protocol For Low Power Mobile Communications, in Proc. of the International Conference and Exhibition On Next Generation Mobile Applications, Services And Technologies. Cardiff, Wales, United Kingdom , IEEE Computer Society Press, September 12-14, 2007, pp. 236-241.

- P. ABI-CHAR, A. Mhamed, B. EL-Hassan, A Secure Authenticated Key Agreement Protocol Based on Elliptic Curve Cryptography, in Proc. of the Third International Symposium on Information Assurance and Security IAS2007, Manchester, United Kingdom , IEEE Computer Society Press, 29-31, August 2007, pp. 89-94.

- P. ABI-CHAR, B. EL-Hassan and A. Mhamed, IEEE 802.11i Standard: Review and Security Analysis Using AVISPA, in Proc. of the National Conference on Current Trends in the Theory and Applications of Computer Science, CTTACS07, NDU, Lebanon, 2007.

- P. ABI-CHAR, A. Mhamed, B. EL -Hassan, An Efficient Authenticated Key Agreement Protocol, in Proc. of the first International Conference on New Technologies, Mobility and Security, NTMS 2007, PARIS, France, May 2007, pp. 45.

# EXECUTIVE SUMMARY

The growing evolution of Information and Communication Technology (ICT) systems towards more pervasive and ubiquitous infrastructures contribute significantly to the deployment of services anywhere, at anytime and for anyone. To provide personalized services in such infrastructures, we should consider both user's privacy and security requirements within context-awareness environment. This can be really achieved owing to context awareness systems which allow us to benefit from sensing and mobile technologies to derive more accurate data, i.e user's profile and contextual information. While the availability of contextual information may introduce new threats against security and privacy, it can also be used to improve dynamic, adaptive and autonomic aspects of security, and user privacy. Moreover, context-aware information offers new opportunities for the establishment of trust relationship among involved entities (e.g. users, devices, and platforms).

Traditional authentication and access control methods require much user interaction in the form of manual login, logouts, and file permission. These manual interactions violate the vision of non-intrusive ubiquitous computing. Traditional authentication and access control mechanisms are context-insensitive, i.e. they do not adapt their security policies to a changing context. Moreover, in pervasive environments where principals are typically unknown and where contextual conditions frequently change, this traditional approach may lead to a combinatorial explosion of the number of policies to be written, force a long development time, and even introduce potential bugs. The traditional approach, when applied to pervasive scenarios, also lacks flexibility. Therefore, we conjecture that the conventional authentication schemes are incompetent in satisfying the needs in context-aware environments.

Furthermore, users are becoming increasingly concerned about their privacy and the

potential risks such as identity theft. In fact, users prefer to interact with services providers anonymously. Therefore, in a World with ubiquitous computing, privacy becomes more important for the users because there will not be anymore a private zone into which a user can retreat. Preserving user privacy can be particularly challenging in a ubiquitous environment, and if privacy is preserved (*through user anonymity*), how can we then convince a service provider that an anonymous user is trustworthy. This challenge is well addressed in this thesis process.

Our Contributions include a flexible and scalable ubiquitous security mechanism that integrates context-aware, trust with automated reasoning to perform context-based authentication and access control framework in ubiquitous computing environments that convincingly satisfy the demands in the (*CAC*). Our Framework is composed of various mechanisms that they altogether yield to a flexible and scalable context-aware framework. In our model, confidence and trust are defined based on users' contextual information. It uses trust engine to calculate user trustworthiness and role's required trustworthiness parameters. It uses fuzzy logic and PSI techniques to provide context-based authentication and dynamic reasoning. Moreover, Privacy is presented and integrated into the framework through the use of privacy control layer, *PSI* and new cryptographic techniques.

# Contents

<b>ACKNOWLEDGEMENTS</b>	<b>v</b>
<b>PUBLICATIONS RESULTING FROM THIS THESIS</b>	<b>vii</b>
<b>EXECUTIVE SUMMARY</b>	<b>ix</b>
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 Motivation . . . . .	2
1.2 Research Issues and Challenges . . . . .	3
1.3 Approach . . . . .	4
1.4 Thesis Contribution . . . . .	4
1.5 Thesis Organization . . . . .	6
<b>2 MATHEMATICAL BACKGROUND</b>	<b>8</b>
2.1 Notations . . . . .	8
2.2 Algebra and Number Theory . . . . .	9
2.2.1 Intractable Problems . . . . .	9
2.3 Cryptographic Primitives . . . . .	10
2.3.1 Digital Signature . . . . .	10
2.3.2 Hash Functions . . . . .	10
2.3.3 Random Number Generator . . . . .	11
2.4 Elliptic Curve Cryptography . . . . .	11
2.4.1 ECDLP-Based Okamoto Identification Scheme . . . . .	13
2.4.2 Identity-Based Signature Schemes . . . . .	14

2.5	Bilinear Pairing . . . . .	14
2.6	Fuzzy Logic . . . . .	15
2.7	Private Set Intersection . . . . .	15
2.8	Security Analysis Tools . . . . .	16
2.8.1	AVISPA and SPAN . . . . .	16
2.9	Conclusion . . . . .	16

**Bibliography** **18**

**3 AUTHENTICATION AND ITS EFFECTS** **21**

3.1	Introduction . . . . .	21
3.2	Cryptographic Techniques and Objectives . . . . .	22
3.2.1	Cryptography Objectives . . . . .	23
3.3	Authentication . . . . .	24
3.3.1	Factorized Authentication . . . . .	24
3.4	Authentication Protocols: Related Work . . . . .	25
3.4.1	Wireless Authentication Protocols . . . . .	25
3.4.2	Mobile Authentication Protocols . . . . .	29
3.5	Access Control . . . . .	32
3.5.1	Access Control: Related Work . . . . .	33
3.6	Privacy Consideration . . . . .	34
3.7	Conclusion . . . . .	35

**Bibliography** **36**

**4 PROPOSED PROTOCOLS AND EVALUATION** **43**

4.1	Introduction . . . . .	43
4.2	Key Management . . . . .	44
4.3	Desirable Properties for key agreement protocols . . . . .	45
4.4	Closely Related Work . . . . .	47
4.5	Protocol Application . . . . .	54
4.6	An ID-Based Pairing Protocol . . . . .	58

4.6.1	Parameters Initialization . . . . .	58
4.6.2	Proposed Protocol Assumption . . . . .	59
4.6.3	Proposed Protocol Description . . . . .	60
4.6.4	Security Analysis . . . . .	63
4.6.5	Formal Analysis . . . . .	64
4.7	Combining Authentication and Access Control . . . . .	65
4.7.1	Protocol Discussion . . . . .	66
4.8	Conclusion . . . . .	67
	<b>Bibliography</b>	<b>69</b>
<b>5</b>	<b>SECURITY IN CONTEXT-AWARE ENVIRONMENTS</b>	<b>73</b>
5.1	Introduction . . . . .	73
5.2	Pervasive Computing . . . . .	75
5.2.1	Properties and Features . . . . .	75
5.2.2	Requirements . . . . .	76
5.2.3	Security Challenges . . . . .	78
5.3	Context-Aware Computing . . . . .	79
5.3.1	Terminology . . . . .	79
5.3.2	Life-Cycle of Context-Aware Information . . . . .	80
5.3.3	Context-Taxonomy . . . . .	80
5.3.4	Reasoning about Uncertain Contexts . . . . .	81
5.4	Privacy In Pervasive Computing . . . . .	84
5.4.1	Privacy Definition . . . . .	84
5.4.2	General Principles and Privacy Requirements . . . . .	85
5.4.3	Privacy-Aware Design Guidelines . . . . .	85
5.5	Authentication in Pervasive Computing . . . . .	87
5.5.1	Authentications Requirements . . . . .	87
5.5.2	Designing Privacy-Based Context-Aware Authentication Systems . . . . .	88
5.6	Trust In Pervasive Computing . . . . .	89
5.6.1	Trust Management in Pervasive Computing . . . . .	90
5.6.2	Trust Establishment in Pervasive Computing . . . . .	91

5.6.3	Privacy in Trust Negotiations . . . . .	92
5.7	Related Work . . . . .	93
5.7.1	Security Infrastructure . . . . .	93
5.7.2	Privacy Related Researches . . . . .	94
5.7.3	Privacy-Enhanced Identity Management Systems . . . . .	96
5.7.4	Trust Researches . . . . .	96
5.8	Conclusion . . . . .	100

**Bibliography** **102**

**6 A TRUST-BASED SECURE FRAMEWORK** **112**

6.1	Introduction . . . . .	112
6.2	Review of Literature . . . . .	113
6.2.1	Statement of the Project Problem . . . . .	114
6.2.2	Closely Related Work . . . . .	115
6.2.3	Work Valuable . . . . .	116
6.3	Towards a New Solution . . . . .	117
6.3.1	The Entities . . . . .	118
6.3.2	Assumptions . . . . .	121
6.4	Context-Based Authentication Scheme . . . . .	122
6.4.1	The Scheme . . . . .	124
6.5	Service Layer Process . . . . .	139
6.6	Framework Interaction Summary . . . . .	140
6.7	Security Analysis and Discussion . . . . .	141
6.8	Conclusion . . . . .	142

**Bibliography** **144**

**7 FRAMEWORK IMPLEMENTATION** **148**

7.1	Scope Of The Prototype . . . . .	148
7.2	Preliminary . . . . .	149
7.3	Platform Extension . . . . .	149

7.3.1	Previous Platform . . . . .	150
7.3.2	New Platform: Extensions and New Design . . . . .	152
7.4	Platform: Implementation And Evaluation . . . . .	159
7.4.1	Implementation Setting . . . . .	159
7.4.2	Implementation and Testing Processes . . . . .	160
7.5	Implementation Outcomes . . . . .	164
7.6	Conclusion . . . . .	167
	<b>Bibliography</b>	<b>169</b>
<b>8</b>	<b>CONCLUSIONS AND FUTURE WORK</b>	<b>170</b>
8.1	Research Summary . . . . .	170
8.2	Summary of Contributions . . . . .	171
8.3	Future Work . . . . .	171
8.4	Conclusion . . . . .	172
<b>A</b>	<b>Basic Fuzzy concepts and Definitions</b>	<b>173</b>
A.1	Trust and Trustworthiness . . . . .	174
A.2	Private Set Intersection . . . . .	174
<b>B</b>	<b>VITA</b>	<b>176</b>

## List of Tables

3.1	Summary of Secret-Key Methods . . . . .	27
3.2	Summary of Public-Key Methods . . . . .	29
3.3	Summary of EAP-Tunnelled Methods . . . . .	29
3.4	Summary of ID-Based Protocol . . . . .	32
4.1	SAKA-Comparison of Performance-1- . . . . .	49
4.2	SAKA-Comparison of Performance-2- . . . . .	50
4.3	EC-SAKA-Comparison of Performance-1- . . . . .	52
4.4	EC-SAKA-Comparison of Performance-2- . . . . .	52
4.5	ECEGS-SKA-Comparison of Performance-1- . . . . .	54
4.6	ECEGS-SKA-Comparison of Performance-2- . . . . .	54
4.7	EC Mathematical Notations . . . . .	59
5.1	Protocol Security Features Comparison . . . . .	100
6.1	EC Mathematical Notations . . . . .	125

# List of Figures

1.1	Security In Pervasive Computing . . . . .	3
4.1	The A-Key Distribution Procedures . . . . .	56
4.2	The EC-based in BSS . . . . .	57
4.3	The EC-based In EAP Stack . . . . .	57
4.4	The EC-based In ESS Networks . . . . .	58
4.5	Our Protocol Protocol . . . . .	60
4.6	The OFMC Output . . . . .	64
6.1	Context-Aware Framework . . . . .	118
6.2	A High-Level View . . . . .	119
6.3	The Authentication Architecture Process . . . . .	123
6.4	The Trust/Risk Layer Architecture -L2- . . . . .	133
6.5	The Extended Access Control Process . . . . .	136
6.6	The Service Layer Architecture -L3- . . . . .	139
7.1	Proposed Service Provision Architecture . . . . .	150
7.2	U2 and P2 Process . . . . .	151
7.3	$U'3$ and $P'3$ Process . . . . .	153
7.4	New Architecture Re-Design . . . . .	158
7.5	ECC Packages Installation Classes . . . . .	161
7.6	ASM AccessServer Java Implementation . . . . .	162
7.7	ASMSecClient Java Implementation . . . . .	162
7.8	User Access Granted . . . . .	163

7.9	User Access Denied . . . . .	163
7.10	Policy Specification in XML . . . . .	164
7.11	Resources Access In Dynamic Environments . . . . .	165
7.12	User Selecting Service . . . . .	165
7.13	User Access Granted-1- . . . . .	166
7.14	User Access Granted-2- . . . . .	166
7.15	User Selection Service . . . . .	167
7.16	Access Decision . . . . .	167

# Chapter 1

## INTRODUCTION

Pervasive computing technology strives to simplify day-to-day life by providing mobile users with the means to provide out personal and business services via portable and embedded devices. These technologies promise to boost productivity through seamless interactions, and allow anytime, anywhere access to applications and services and the constructions of smart homes and environments. Although pervasive computing technology looks promising, a number of critical challenges need to be addressed before it can be widely deployed. These critical challenges include *Security, Privacy, Trust, and contextual Information*. The problem is serious because pervasive applications do not usually have well defined security perimeters and are dynamic in nature. Moreover, these applications and services use knowledge of surrounding physical and environments spaces. This requires a security measures based on contextual information which must be adequately protected from security breaches. Traditional authentication and access control mechanisms that focus merely on digital security are context-insensitive, i.e. they are unable to adapt to the rapidly changing of context parameters and thus are inadequate for securing new exposures and vulnerabilities within pervasive computing environments. Therefore, context-based authentication and authorization are one of the topics which have the potential to become the next hype.

The research presented in this thesis introduces a new vision for network security, namely context-aware based authentication. A three pronged efforts are required in this direction. First, both user and context-aware based authentication schemes with a high level

of confidence need to be developed. Second, a privacy control engine needs to be presented to protect and ensure user's privacy, and finally the services exported by the system need to be protected from unauthorized access. A unique combination of privacy control, contextual information, security (i.e., authentication, access control, etc.), and trust is presented in this thesis. A user is authenticated based on presented context-aware environments attributes and user's credentials. As part of the authentication process, the access acceptance and or rejection and the privileges of the user are retrieved automatically. The use of this combination of context-aware and authentication system with privacy control is expected to protect users' data as well as privacy.

In this chapter, we present the motivation for this thesis, research issues and challenges, the approach we have chosen to address these issues, and thesis contributions. We finally conclude this chapter with the structure of the thesis.

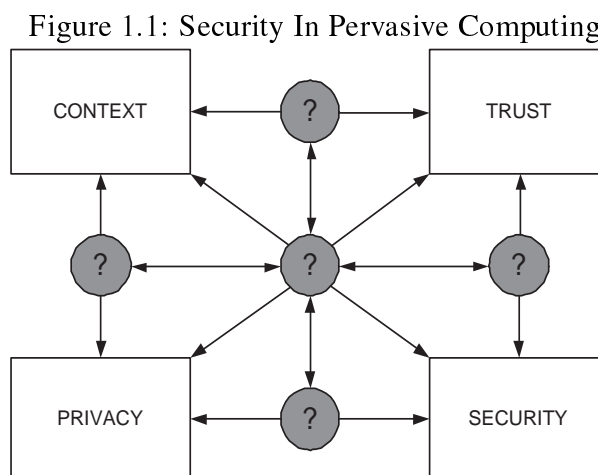
## 1.1 Motivation

With the growth of mobile and sensor devices, we are moving towards more pervasive and ubiquitous infrastructures which contribute significantly to the deployment of services anywhere, at anytime and for anyone. To provide personalized services in such infrastructures, we should consider both user's privacy and security requirements within context-awareness environment. Designing secure pervasive systems require one to understand what resources an entity has access to, how to provide privacy and confidentiality, which entities can be trusted, how these trust relationship change over time, etc. In traditional models, authentication and access control were context-less and depend on specific static credentials of the user and objects. Moreover, as the availability of contextual information may introduce new threats against security and privacy, it can also be used to improve dynamic, adaptive and autonomic aspects of security, and user privacy. Context-aware information offers new opportunities for the establishment of trust relationship among involved entities (e.g. users, devices, and platforms). As context awareness represents new challenges and new opportunities regarding privacy, trust and security of users in pervasive computing environments (PCE), the main purpose of this thesis aims to discover how contextual attributes can be used to support and enhance authentication, access control and trust in a dynamic, mobile

environment.

## 1.2 Research Issues and Challenges

Pervasive computing researchers are investigating specific security vectors that are related to each other in subtle ways and can not be addressed in isolation. These vectors include contextual information, security, privacy, and trust. One key challenge in pervasive applications is to create and manage a trustworthy pervasive system that makes advances along these vectors in combination. In this approach, permission to access resources or services is moderated by checking for a context-aware authentication and access control processes both associated with a trust based evaluation and where privacy is preserved. The logic presented in (Figure 1.1) illustrates how combining assessment of the interrelationships between these vectors in order to build a trustworthy system with confident decision-making.



In a pervasive environment, users are mobile and typically access resources using mobile embedded devices. As a result the context of a user (i.e. location, time, system resources, network state, network security configuration, etc.) is highly dynamic, and granting a user access without taking the users current context into account can compromise these security vectors as the users access privileges not only depend on *"who the user is"* but also on *"where the user is"* and *"what is the users state and the state of the users environment"*. Traditional authentication and access control mechanisms break down in

such an environment and a fine-grained authentication and access control mechanism that change the privilege of a user dynamically based on context information is required. Although a lot of work has been done in the area of authentication and access control, most of these works are user-centric, where only credentials of the user are considered when granting access permission. Moreover, the existing research does not address pervasive application where context is dynamic and users' privileges must continuously adapt based on the runtime context.

### 1.3 Approach

The approach we have used to achieve the goal of our thesis is by designing a context-aware security framework that integrates privacy control, authentication and trust evaluation with access control. We will propose a new approach to achieve attributes-based authentication by using the private set intersection techniques. A significant body of the work has emerged around the use of contextual information in computer systems. Context can be used to provide these systems with certain capabilities inherent to human perception and reasoning. Context describes a specific situation by capturing the setting in which an event occurs. These observations have led us to consider the impact of context on security services. In particular, we are interested in how contextual attributes can be used to support and enhance authentication in a dynamic, mobile environment while trust and privacy are preserved. Finally, and as a proof of concept, a prototype of the proposed framework was implemented.

### 1.4 Thesis Contribution

The research in this thesis was initiated to investigate the use of contextual information in authentication processes. We define context-aware authentication process, and proposed a framework that convincingly satisfies the demands in the Context-Aware Computing *CAC*. Our framework covers specifications of an authentication mechanism in context-aware environments. Our framework is composed of various mechanisms that they altogether yield a flexible, scalable context-aware authentication. In our model, trust calculation is based

on user's set attributes and role. It also uses fuzzy logic rules to unravel the complexities of setting authentication rules and policies. Moreover, privacy is always present in our work and it is integrated into the system by using a privacy control layer and also new cryptography techniques. In addition, special emphasis was placed on the contextual attributes related to people with special needs. The contributions of this thesis towards introducing and incorporating of new secure authentication mechanism in a context-aware model are listed below:

1. **Study of Existing Authentication Protocols:** A survey regarding authentication protocols has been done to determine the possible strengths and weakness. Several solutions were proposed to overcome some weaknesses that were identified. (Chapters 3 and 4).

2. **Study of Existing Context-Aware Frameworks:** similarities and differences frameworks that abstract sensory context from the applications have been studied and presented. Such a framework forms useful part of applications that require adaptation to sensory contexts where sensory contexts are those contextual information obtained from sensors. Examples include location, time, etc. (Chapter 5).

3. **Develop a Context-Aware Authentication Mechanism:** The purpose is to determine the authenticity of the user with a high level of confidence and provide the system with the data that is required to gauge the privileges of that user. This mechanism has been achieved by using the *Private Set Intersection* technique. (Chapter 6).

4. **Provide a Dynamic Discovery Mechanism:** The purpose is to automatically support dynamic discovery of services for users through a context-based provision process. In our attributes-based authentication, we aim to have a service provision framework that combines user's profiles and contextual information to select appropriate services to the end users from thousands of desultory services. (Chapter 6).

5. **Overhead-Less Model:** By combining the advertisement message and the access control within the authentication model. Moreover, the flexibility provided by our framework also helps in the sense of computational and storage/communication complexities. (Chapter 6).

6. **Privacy Preserving-Based Model:** Privacy violation through information leakage and personal information disclosures have been dealt by using a privacy control layer and advanced cryptography perspectives in our proposed model. Moreover, the help of the

fuzzy matching operations and rules, we can define and form a formal decision-making process to handling uncertainty in our attributes-based model. (Chapter 6).

**7. Dynamic Trust-Based Authentication Model:** Our Proposed model is a dynamic attributes-based authentication model based on trust measures for secure communications and access control decisions among the services. Moreover, Our Framework provides both user and context based authentication. With the help of the fuzzy operations and rules, we can define and form a formal decision-making process to calculate user trustworthiness and role's required worthiness parameters. (Chapter 6).

**8. Combining Authentication and Access Control:** Our Proposed model is a dynamic authentication model That combine authentication and access control simultaneously.

**9. Prototype Implement:** In addition to the design of our security architecture mentioned above, we discuss an implementation prototype that was deployed using the platform in Handicom lab at Telecom & Management SudParis (ex. INT). (Chapter 7).

## 1.5 Thesis Organization

The remainder of this dissertation is organized as follows:

**Chapter 2:** Provides the necessary background and foundations of Cryptography that will be used in the subsequent chapters. We give introduction to the topics of complexity theories, algebra, number theory. We then proceed to review various cryptographic primitives including encryption, digital signatures, etc. Finally we elaborate on private set intersection and trust protocols.

**Chapter 3:** We briefly survey the literature on security related to our thesis. They serve as a good tutorial on various security goals and notions, definitions and interaction. Moreover, we present a literature on related works concerning authentication protocols in *Extensible Authentication Protocol (EAP)* and in mobile communications.

**Chapter 4:** We investigate in depth our proposed authentication protocols that could be appropriate for low-cost devices computing. We first give an introduction regarding security requirement together with discussion on their importance and impact. we survey the literature on closely related works that has influenced our thesis by reviewing various security goals and notions, current state-of-art technology, similarities and differences among

various authentication schemes. Finally, we present a fully developed security model that captures the security requirements by combining authentication and access control and we present the security analysis for this protocol.

**Chapter 5:** We firstly introduce the features, security challenges and requirements of pervasive computing. Then, we give an overview of context-aware computing. Furthermore, the privacy issue is discussed, followed by the trust researches in pervasive computing. In addition, we specify the requirements of achieving authentication in the pervasive computing environment, especially for supporting context awareness and privacy. Finally we review the related work conducted in this area and conclude the chapter by proposing a number of future directions.

**Chapter 6:** In this chapter, we proposed our trust and context-aware based security framework. We first showed the importance of context-aware to be the key for our scheme to be practically deployed in smart environments where privacy and authenticity are presented. Moreover, we motivate the design of an access control scheme that addresses the context-aware issue for access decisions and we propose an extended, trust-enhanced, model that affects the level of trust associated with a user. We present the configuration mechanism needed to achieve the proposed framework and then we analyze the security of the protocol. Our proposed framework also supports flexibility in the sense of computational and storage/communication complexities.

**Chapter 7** This Chapter introduces the implementation architecture with a testing evaluation using the lab's platform. We first present a brief description of the platform existed at Telecom SudParis Lab. Then, we have detailed the prototype implementation of our framework by presenting our new platform's extensions that are needed to provide more privacy control, security and trust. Moreover, we have presented the new final platform re-design, and we have presented the implementation phases that were done with the evaluation process. Finally, we have concluded with an example scenario.

Finally, **Chapter 8** This Chapter present the conclusion by providing a summary of the research and its results, as well as proposing future directions for research, instigated by this thesis.

# Chapter 2

## MATHEMATICAL BACKGROUND

This chapter is a collection of necessary background and foundations of cryptography that have been used throughout this thesis. We first give an introduction to several notations that have been used through the thesis, and to the topics of complexity theories, algebra, number theory. We then proceed to review various cryptographic primitives including encryption, digital signatures, elliptic curve techniques, bilinear pairing, etc. Moreover, we elaborate on private set intersection and fuzzy logic matching for reasoning based protocols. Finally, we present two security tools used for specifying, validating, and verifying security protocols.

### 2.1 Notations

In this thesis, we denote by  $\mathbb{N}$  the set of positive integers, by  $\mathbb{Z}$  the set of integers, and by  $\mathbb{R}$  the set of real numbers. We denote by  $[a, b]$  the integers  $x$  satisfying  $a \leq x \leq b$ .  $|s|$  means the number of elements in  $s$  where  $s$  is a finite set. We denote by  $c$ , a prime number (or a prime), to be a natural number which has exactly two distinct natural number divisors: 1 and itself. We denote by  $gcd$ , greatest common divisor, to be the greatest or highest common factor of two or more non-zero integers. When we write  $x \in_R X$ , we mean  $x$  is chosen from the finite set  $X$  uniformly at random. We denote by  $d$  said to be *congruent to*  $e$ , written  $d \equiv e \pmod{n}$ , if  $n$  divides  $(d - e)$ . The integer  $n$  is called the *modulus* of the *congruence*. We denote by  $\mathbb{Z}_n$ , the integers modulo  $n$ , is the

set of (equivalence classes of) integers  $\{0, 1, 2, \dots, n - 1\}$ . Addition, subtraction, and multiplication in  $\mathbb{Z}_n$  are performed modulo  $n$ . We denote by  $\mathbb{Z}_n^*$ , the *multiplicative group* of  $\mathbb{Z}_n$ , to be equal to  $\{f \in \mathbb{Z}_n, \gcd(f, n) = 1\}$ . In particular, if  $n$  is a prime, then  $\mathbb{Z}_n^* = \{f | 1 \leq f \leq n - 1\}$ . We denote by  $\phi(n)$ , the order of  $\mathbb{Z}_n^*$ , to be the number of element in  $\mathbb{Z}_n^*$ , namely  $|\mathbb{Z}_n^*|$ .

## 2.2 Algebra and Number Theory

Algebra and Number Theory are the mathematical foundation of Modern Cryptography. Numerous cryptographic algorithms are designed around results from them. They are also the cornerstone of (provable) security of cryptographic schemes. We briefly introduce the following definitions [19]:

**Group** A group is a set  $G$  together with an associative binary operation  $*$  on elements of  $G$  such that  $G$  contains an identity element for  $*$  and every element has an inverse under  $*$ . Often, a group is denoted by  $\langle G, * \rangle$  or simply by  $G$ .

**Cyclic Group** A group  $G$  is cyclic if there is  $g \in G$  such that every element  $a \in G$  can be written in the form of  $g^k$  for some  $k \in \mathbb{Z}$ . We call such  $g$  a generator of  $G$  and write  $\langle g \rangle = G$  to indicate that  $g$  generates  $G$ . Let  $G$  be a group and  $a \in G$ .

**Group Order** Let  $G$  be a group and  $a \in G$ . The order of  $a$ , denoted by  $ord(a)$ , is the smallest positive integer  $n$  such that  $a^n = 1$ , provided that such an integer exists. If such an  $n$  does not exist, then the order of  $a$  is defined to be  $\infty$ .

### 2.2.1 Intractable Problems

Various cryptographic protocols rely their security on the intractability of one or more mathematical problems.

**Discrete Logarithm (DL)** Let  $G$  be a finite cyclic group generated by  $g \in G$  of order  $u = \#G$ . The discrete logarithm of some element  $a \in G$ , denoted by  $\log_g(a)$ , is the unique integer  $x$ ,  $0 \leq x \leq u$ , such that  $a = g^x$ . The *DL Problem* is to find  $\log_g(a)$ . The *DL Assumption* says that there exists no *PPT* algorithm that can solve the *DL Problem*, in time polynomial in the size of  $u$ .

**Computational Diffie-Hellman (CDH)** Let  $G$  be a cyclic group generated by  $g \in G$  of order  $u = \#G$ . Given  $g, g^a$  and  $g^b \in G$ , the *CDH* Problem is to find the element  $g^{ab} \in G$ . The *CDH* Assumption says there exists no *PPT* algorithm that can solve the *CDH* Problem, in time polynomial in the size of  $u$ . Obviously, if the *DL* problem can be solved in polynomial time, then the *DH* problem can be solved in polynomial time. For some groups, the *DH* and the *DL* problems have been proved to be computationally equivalent.

**Decisional Diffie-Hellman (DDH)** Let  $G$  be a cyclic group generated by  $g$  of order  $u = \#G$ . The *DDH* Problem is to distinguish between the distributions  $(g; g^a; g^b; g^c)$  and  $(g; g^a; g^b; g^{ab})$ , with  $a; b; c \in_R Z_u$ . The *DDH* Assumption says there exists no *PPT* algorithm solve the *DDH* Problem, in time polynomial in the size of  $u$ . The *DDH* problem was first mentioned in [20], although there are earlier cryptographic systems that implicitly rely on the hardness of this problem, e.g. [21].

## 2.3 Cryptographic Primitives

In this section, we will list the cryptographic primitives that have been used throughout this thesis.

### 2.3.1 Digital Signature

Digital signature scheme is a mathematical scheme for demonstrating the authenticity of a digital message or document. A valid digital signature gives a recipient reason to believe that the message was created by a known sender, and that it was not altered in transit. Digital signatures are commonly used for software distribution, financial transactions, and in other cases where it is important to detect forgery and tampering.

### 2.3.2 Hash Functions

Hash functions, also called message digests and one-way encryption, are algorithms that, in some sense, use no key. Instead, a fixed-length hash value is computed based upon the plaintext that makes it impossible for either the contents or length of the plaintext to be

recovered. Hash functions are often used to ensure that the file has not been altered by an intruder or virus. Hash functions are also commonly employed by many operating systems to encrypt passwords. The purpose of hash functions in cryptographic sense to provide data integrity and message authentication. For these usage, adopted hash functions(H) should satisfy the following requirements: *Compression*, *One – wayness*, *Collision-Avoidance*, and *Efficiency*. A one-way hash function OWHF is a hash function which offers preimage and 2nd preimage resistance. A collision resistant hash function CRHF is a hash function which is 2nd-preimage resistant and collision-freshness.

### 2.3.3 Random Number Generator

Random number generation is used in a wide variety of cryptographic operations, such as key generation and challenge/response protocols. A random number generator is a function that outputs a sequence of 0s and 1s such that at any point, the next bit cannot be predicted based on the previous bits. However, true random number generation is difficult to do on a computer, since computers are deterministic devices. Thus, if the same random generator is run twice, identical results are received. A pseudo-random number generator *PRNG* produces a sequence of bits that has a random looking distribution. Pseudo-random number generators are often based on cryptographic functions like block ciphers or stream ciphers.

## 2.4 Elliptic Curve Cryptography

Elliptic curve cryptography is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. Many researchers have examined elliptic curve cryptosystems, which were firstly proposed by Miller [1] and Koblitz [2]. The elliptic curves which are based on the elliptic curve discrete logarithm problem over a finite field have some advantages than other systems: the key size can be much smaller than the other schemes since only exponential-time attacks have been known so far if the curve is carefully chosen [3], and the elliptic curve discrete logarithms might be still intractable even if factoring and the multiplicative group discrete logarithm are broken. In this thesis we use an elliptic curve  $E$  defined over a finite field  $F_p$ . The elliptic curve parameters to be

selected [5] are:

1 -Two field elements  $a$  and  $b \in F_p$ , which define the equation of the elliptic curve  $E$  over  $F_p$  (i.e.,  $y^2 = x^3 + ax + b$  in the case  $p \geq 4$ , where  $4a^3 + 27b^2 \neq 0$ ).

2 -Two field elements  $x_p$  and  $y_p$  in  $F_p$ , which define a finite point  $P(x_p, y_p)$  of prime order in  $E(F_p)$  ( $P$  is not equal to  $O$ , where  $O$  denotes the point at infinity).

3 -The order  $n$  of the point  $P$ .

The Elliptic Curve domain parameter must be verified to meet the following requirements [5]. In order to avoid the Pollard-rho [6] and Pohling-Hellman algorithms for the elliptic curve discrete logarithm problem, it is necessary that the number of  $F_p$ -rational points on  $E$ , denoted by  $\#E(F_p)$ , be divisible by a sufficiently large prime  $n$ . To avoid the reduction algorithms of Menezes, Okamoto and Vanstone [7] and Frey and Ruck [8], the curve should be non-supersingular (i.e.,  $p$  should not divide  $(p + 1 - \#E(F_p))$ ). To avoid the attack of Semaev [9] on  $F_p$ -anomalous curves, the curve should not be  $F_p$ -anomalous (i.e.,  $\#E(F_p) \neq p$ ).

In the following subsections, we will briefly introduce the EC-discrete logarithm problem and Diffie-Hellman key exchange based on EC and then we will introduce the elliptic curve based digital signature algorithm (EC-DSA) and the elliptic curve-based Elgamal signature scheme (EC-EGS).

Let  $E$  be an elliptic curve defined over a finite field  $F_p$  and let  $P \in E(F_p)$  be a point of order  $n$ . Given  $Q$  where  $Q \in E(F_p)$ , the elliptic curve discrete logarithm problem (ECDLP) is to find the integer  $l$ ,  $0 \leq l \leq n - 1$ , such that  $Q = l.P$ . The Diffie-Hellman key agreement protocol runs as follows: The first party selects a random number  $n_a$  and computes  $Y_a = n_a B$ , he sends  $Y_a$  to the second party. Similarly, the second entity computes  $Y_b = n_b B$  and sends  $Y_b$  to the first party. Finally the two parties generate the same key  $K = n_a Y_b = n_b Y_a = n_a n_b B$ .

**EC-Based Digital Signature Algorithm:** The EC-DSA runs as follows: The signer selects a random number  $x_a$ , where  $2 \leq x_a \leq n - 2$ , as his secret key and computes the corresponding public key  $Y_a = x_a B$ . Therefore the public key and the private key are  $(E, Y_a, B, n)$  and  $x_a$ . To generate a signature for a message  $m$ , the signer will select a random number  $k$ , where  $2 \leq k \leq n - 2$  computes  $r = x(KB) \bmod n$ . If  $r \neq 0$ ,

then computes  $s = K^{-1}(h(m) + x_a \cdot r) \bmod n$  and the signature will be  $(r, s)$ . To verify the signature, the verifier will first confirm that  $r$  and  $s \in [2, n-2]$  and then computes  $c = s^{-1} \bmod n$  and  $h(m)$ , then computes  $t_1 = (h(m) * c) \bmod n$  and  $t_2 = (rc) \bmod n$ , also the verifier computes  $T = (t_1 B + t_2 Y_a) \bmod n$  and  $v = x(T) \bmod n$ . Finally the verifier will accept the signature if and only if  $(v == r)$ .

**EC-Based Elgamal Signature Scheme:** The EC-EGS runs as follows: The signer selects a random number  $x_a$ , where  $2 \leq x_a \leq n - 2$ , as his secret key and computes the corresponding public key  $Y_a = x_a B$ . Therefore the public key and the private key are  $(E, Y_a, B, n)$  and  $x_a$ . To generate a signature for a message  $m$ , the signer will select a random number  $k$ , where  $2 \leq k \leq n - 2$  computes  $R = kB$  and computes  $r = x(KB) \bmod n$ . If  $r \neq 0$ , then computes  $s = K^{-1}(h(m) + x_a r) \bmod n$ . The couple  $(R, s)$  will be the signer's signature of  $m$ . To verify the signature, the verifier will first confirm that  $r$  and  $s \in [2, n-2]$  and then computes  $v_1 = sR$  and  $v_2 = h(m)B + rY_a$ . Finally the verifier will accept the signature if and only if  $(v_1 == v_2)$ .

### 2.4.1 ECDLP-Based Okamoto Identification Scheme

In this subsection, we briefly describe the elliptic curve based Okamoto Identification Scheme. The Okamoto identification protocol is considered secure against active and concurrent attack under the assumption of the hardness of the discrete logarithm problem [10]. The set of system parameters are  $(q, FR, a, b, P_1, P_2, n, h)$ . The Prover's secret are  $(s_1, s_2)$  such that  $Z = -s_1 \cdot P_1 - s_2 \cdot P_2$ . the steps of the protocol are:

A prover: the prover picks  $r_i \in \{0, \dots, n - 1\}, i = 1, 2$  and sends  $X = r_1 \cdot P_1 + r_2 \cdot P_2$  to the reader.

The reader picks up a number  $e \in [1, 2^t]$  and sends it to the prover. The prover computes  $y_i = r_i + e \cdot s_i, i = 1, 2$  and sends them to the reader.

The Reader checks if  $y \cdot p + e \cdot Z = X$ , by computing  $y_1 \cdot P_1 + y_2 \cdot P_2 + e \cdot Z$  and comparing it to  $X$ . if they are equal, then the reader accepts else rejects.

### 2.4.2 Identity-Based Signature Schemes

An Identity-based signature [11] scheme consists of four phases namely Setup, Extract, Sign, and Verify. The PKG initializes the system in the Setup phase by generating the system public parameters. The PKG also chooses a master key and keeps it secret. The master key is used in the Extract phase to calculate private keys for the participating users in the system. A signer signs a message in the Sign phase using a private key given by PKG corresponding to his/her identity. To verify a signature of a user with identity ID, a verifier just uses ID in the Verify phase.

## 2.5 Bilinear Pairing

This section briefly describes the bilinear pairing, the BDHP and CDHP techniques.

Let  $G_1$  and  $G_2$  denote two groups of prime  $q$ , where  $G_1$  is an additive group that consists of points on an elliptic curve, and  $G_2$  is a multiplicative group of a finite field. A bilinear pairing is a computable bilinear map between two groups, which could be the modified Weil pairing or the modified Tate pairing [12, 13]. For our proposed architectures within this thesis, we let  $e$  denote a general bilinear map  $e : G_1 \times G_1 \rightarrow G_2$ , which has the following four properties:

1 -*Bilinear*: if  $P, Q, R \in G_1$  and  $a \in \mathbb{Z}_q^*$ ,  $e(P + Q, R) = e(P, R).e(Q, R)$ ,  $e(P, Q + R) = e(P, Q).e(P, R)$  and  $e(aP, Q) = e(P, aQ) = e(P, Q)^a$ .

2 -*Non – degenerate*: There exists  $P, Q \in G_1$ , such that  $e(P, Q) \neq 1$ .

3 -*Computability*: There exist efficient algorithms to compute  $e(P, Q)$  for all  $P, Q \in G_1$ .

4 -*Alternative*:  $e(P, Q) = e(Q, P)^{-1}$ .

**Definition 1** -The bilinear Diffie-Hellman problem (BHDP) for a bilinear pairing is defined as follows: Given  $P, aP, bP, cP \in G_1$ , where  $a, b$  and  $c$  are random numbers from  $\mathbb{Z}_q^*$ , compute  $e(P, P)^{abc} \in G_1$ . BDHP assumption: The BDHP problem is assumed to be hard, that is, there is no polynomial time algorithm to solve BDHP problem with non-negligible probability.

**Definition 2** -The computational Diffie-Hellman problem (CDHP) is defined as follows: Given  $P, aP, bP \in G_1$ , where  $a$  and  $b$  are random numbers from  $Z_q^*$ , compute  $abP \in G_1$ . CDHP assumption: There exists no algorithm running in polynomial time, which can solve the CDHP problem with non-negligible probability.

## 2.6 Fuzzy Logic

Fuzzy logic is a form of multi-valued logic derived from fuzzy set theory to deal with reasoning that is approximate rather than precise. Fuzzy logic theory is used to express fuzzy information, human's experience, human brain concepts, and cognitive process. It has been widely used in decision field. The use of fuzzy logic helps in supporting reasoning under uncertainty. The concept of fuzzy logic was first introduced by [22]. Fuzzy logic employs fuzzy sets to deal with imprecise and incomplete phenomena. A Fuzzy set [15, 18] is any set that allows its members to have different grades of membership (membership function) in the interval  $[0,1]$ . The membership function is a graphical representation of the magnitude of participation of each input. It associates a weighting with each of the inputs that are processed, define functional overlap between inputs, and ultimately determines an output response. The rules use the input membership values as weighting factors to determine their influence on the fuzzy output sets of the final output conclusion. A fuzzy set usually represented by  $A = \sum A(x)/x$  that  $A(x)$  is a member of the set and  $x$  is its membership degree. A fuzzy set  $A$  is a subset of a fuzzy set  $B (A \subseteq B)$  if and only if  $A(x) \leq B(x)$  for all  $x \in U$ . Basic operations of intersection, union, and complement [4, 5] are defined in terms of membership functions as follows:  $(A \cap B)(x) = A(x) \wedge B(x)$ ;  $(A \cup B)(x) = A(x) \vee B(x)$ ; and  $A(x) = 1 - A(x)$  for all  $x \in U$ .

## 2.7 Private Set Intersection

Private Set Intersection Engine (*PSIE*) are cryptographic techniques [23] allowing two or more parties, each holding a set of inputs, to jointly identify the intersection of their inputs sets (i.e, shared context), without leaking any information about credentials that each entity might have. Nevertheless, both entities, the prover and the verifier, need to protect their

credentials from each other. Moreover, any entity awaiting to be authenticated by a server has to establish enough confidence in it and be able to present the required attributes.

## 2.8 Security Analysis Tools

Security protocols are communication protocols that aim at providing security guarantees through the application of cryptographic primitives. In the following, we will present two security tools used for specifying, validating, and verifying security protocols.

### 2.8.1 AVISPA and SPAN

The AVISPA project aims at developing a push-button, industrial-strength technology for the analysis of large-scale Internet security-sensitive protocols and applications. In The AVISPA tool [16], security protocols are specified using the High Level Protocol Specification Language (HLPSL). The HLPSL specification is translated into an Intermediate Format (IF). The current version of the AVISPA tool integrates four back-ends: OFMC, CL-ATSE, SATMC and TA4SP. Before we run verifications from AVISPA [16, 17], our protocols were written in the High Level Protocol Specification Language, or HLPSL, and also was written in order to be suitable for the OFMC validation. Once the HLPSL specification was debugged, it was checked automatically for attack detection using the AVISPA verification tools. SPAN [16] is designed to help protocol developers in writing HLPSL specifications. From an HLPSL specification SPAN helps in interactively building Message Sequence Charts (MSC) of the protocol execution. Since SPAN implements an active intruder, it can also be used to interactively find and build attacks over protocols

## 2.9 Conclusion

In this chapter, we have briefly provided all the necessary background and foundations of cryptography that will be used in the subsequent chapters. We have given an introduction to the topics of complexity theories, algebra, number theory. We have reviewed various cryptographic primitives including encryption, digital signatures, etc. Moreover, we have

elaborated on private set intersection and Fuzzy Logic Matching concepts. Finally, we have reviewed security analysis primitives.

# Bibliography

- [1] V. Miller, *Uses of elliptic curves in cryptography*, In Proceeding of Crypto '85, Santa Barbara, pp. 417 - 426. 1986.
- [2] N. Koblitz, *Elliptic Curve cryptosystems*, Mathematics of Computation, vol 48., pp. 203 - 209, 1987.
- [3] N. Koblitz, *CM-Curves With Good Cryptography Properties*, In Proceeding of Crypto' 91, Santa Barbara, USA, 1992.
- [4] A. Berrached, M. Beheshti, A. De Korvin and R. Alo, *Applying Fuzzy Logic Relation Equations to Threat Analysis*, In Proceeding of the 35th Annual Hawaii International Conference on System Sciences, pp. 684-688, 2002.
- [5] L. Law, A. Menezes, M. Qu, J. Solinas and S. Vanstane, *An efficient Protocol for Authenticated Key Agreement*, Designs, Codes and Cryptography, vol. 28, pp. 119-134, 2003.
- [6] J. Pollard, *Monte Carlo methods for index computation mod p*, Mathematics of Computation, vol. 32, pp. 918-924, 1978.
- [7] A. Menezes, T. Okamoto and S. Vanstane, *Reducing elliptic curve logarithms in a finite field*, IEEE Transactions on Information Theory, vol. 39, pp. 1639-1646, 1993.
- [8] G. Frey and H. Ruck, *A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves*, Mathematics of Computation, vol 62, pp. 865-874, 1994.

- [9] I. Semaev, *Evaluation of Discrete logarithms in a group of  $p$ -torsion points of an elliptic curve in Characteristic  $p$* , Mathematics of Computation, vol. 67, pp. 353-356, 1998.
- [10] D. R. Stinson, *Cryptography Theory and Practice*, In Proceeding of Chapman and Hall/CRC, Third Edition, pages: 353-438, 2006.
- [11] J.C. Cha and J.H. Cheon, *An identity-based signature from gap Diffie-Hellman groups*, Public Key Cryptography- PKC2003, LNCS 2567, Springer-Verlag, 2003, pp.18-30.
- [12] D. Boneh and M. Franklin, *Identity-based encryption from the Weil Pairing*, Advanced in CRYPTO2001, LNCS 2139, pp. 213-229, 2001.
- [13] G. Frey, M. Muller and H. Ruck, *The Tate Pairing and the discrete logarithm applied to elliptic curve cryptosystem*, IEEE Transaction on Information Theory, Vol. 45, No.5, pp. 1717-1719, 1999.
- [14] Y. Xun, *An Identity-based Signature Scheme From the Weil Pairing*, IEEE Communications Letter, Vol. 7, No. 2, FEBRUARY 2003.
- [15] T. Hassan, A. Morteza and J. Rasool, *Enhancing Role-Based Access Control Through Fuzzy Relations*, In Proceeding of the Third International Symposium on Information Assurance and Security, pp. 131-136, 2007.
- [16] AVISPA, *Automated Validation of Internet Security Protocols and Applications*, <http://www.avispa-project.org>, 2006.
- [17] SPAN, *A Security Protocol ANimator for AVISPA*, <http://www.irisa.fr/lande/genet/span>, 2008.
- [18] S. Ovchinnikov, *Fuzzy Sets and Secure Computer Systems*, In Proceeding of the New Security Paradigms Workshop, Little Compton, Rhode Island, USA, pp. 54-62, 1994.

- [19] A. Menezes, P. Van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, In The CRC Press, <http://www.cacr.math.uwaterloo.ca/hac/>, 2nd edition, 1996.
- [20] M. Bellare, and P. Rogaway, *Random oracles are practical: a paradigm for designing efficient protocols*, In Proceedings of the 1st ACM conference on Computer and Communications Security, pages 62-73. ACM Press, 1993.
- [21] D. Chaum, and E. V. Heyst, *Group signatures*, In EURO- CRYPT 1991, volume 547, pages 257-265, 1991.
- [22] L.A., Zadhe *Outline of a new Approach to the Analysis of Complex Systems and Decision Processes* , In Proceeding of IEEE Trans. On Systems, Man and Cybernetics, 1973. 3(1), pp. 28-44.
- [23] M. Freedman, K. Nissin, and B. Pinkas, *Efficient Private Set Intersection*, In Proceeding In Advanced in Cryptology, Eurocrypt'04, vol. 3024 of Lecture Notes in Computer Sciences, Springer-Verlag, pp. 1-9, 2004.

## Chapter 3

# AUTHENTICATION AND ITS EFFECTS

In this chapter, we present authentication definitions and terminology. We introduce Cryptography by going briefly through its historical development and introducing its significance effects on authentication and privacy both of the users and protocols. We survey the literature on how authentication interacts in ways that negatively affect access control and security. In this chapter we also present a survey on relevant related work on authentication and access control. Several parts of the chapter are based on two accepted papers, one presented at CTTACS07 [7] National Conference, Lebanon, and the other presented at NTMS07 [8] International Conference, France.

### 3.1 Introduction

According to [58], *Authentication*, *Access Control*, and *Privacy* refer to the problems of ensuring that communications takes places in a secure manner and only between the right parties without disclosure of information to unauthorized eavesdroppers. Individuals authenticate themselves to information systems in many different context. The identifiers and attributes presented by individuals for authentication process vary, depending on the situation, environments, capabilities, etc. In the past, authentication was almost synonymous with password systems, but today's authentication system must do more. Moreover,

while authentication provides proof of identity, it does not describe the privileges an entry is intended to process. So for instance, you are authenticated before you access services and resources, but this does not tell the system which data you are entitled to access. This later function is known as the authorization or access control.

## 3.2 Cryptographic Techniques and Objectives

Cryptography is the science of writing in secret code and is an ancient art; the first documented use of cryptography in writing dates back to circa 1900 B.C. when an Egyptian scribe used non-standard hieroglyphs in an inscription. Cryptography, then, not only protects data from theft or alteration, but can also be used for user authentication. Roger and Schroeder [65] illustrate how authentication can be achieved with encryption. Gifford [66] observes that encryption can be used to assign capabilities, access control and information flow control by encapsulating data in encrypted objects. Authentication is necessary, but not sufficient, for providing confidentiality. Instead, information must be protected using cryptography. Cryptography requires the use of secrets, also known as a keys, that provide the means to encrypt and decrypt data. Many cryptographic solutions involve two-way authentication, where both the user and the system must each convince the other that they know the shared secret, without this secret ever being transmitted in the clear over the communication channel. As illustration, authentication protocols may employ a cryptographic nonce as the challenge to ensure that every challenge-response sequence is unique. Challenge-response authentication can help solve the problem of exchanging session keys for encryption where mutual authentication is performed using the challenge-response handshake in both directions. Using a key derivation function, the challenge value and the secret may be combined to generate an unpredictable encryption key for the session. This is particularly effective against a man-in-the-middle attack, because the attacker will not be able to derive the session key from the challenge without knowing the secret, and therefore will not be able to decrypt the data stream.

### 3.2.1 Cryptography Objectives

According to [67], Cryptography is the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication, and data origin authentication. Cryptography is not the only means of providing information security, but rather one set of techniques.

The fundamental objectives of any good cryptographic based algorithm are : (1) privacy or confidentiality; (2) data integrity; (3) authentication; and finally (4) non-repudiation.

**Confidentiality:** is a service used to keep the content of information from all but those authorized to have it. Secrecy is a term synonymous with confidentiality and privacy. There are numerous approaches to providing confidentiality, ranging from physical protection to mathematical algorithms which render data unintelligible.

**Data integrity:** is a service which addresses the unauthorized alteration of data. To assure data integrity, one must have the ability to detect data manipulation by unauthorized parties. Data manipulation includes such things as insertion, deletion, and substitution.

**Authentication:** is a service related to identification. This function applies to both entities and information itself. Two parties entering into a communication should identify each other. Information delivered over a channel should be authenticated as to origin, date of origin, data content, time sent, etc. For these reasons this aspect of cryptography is usually subdivided into two major classes: entity authentication and data origin authentication. Data origin authentication implicitly provides data integrity (for if a message is modified, the source has changed).

**Non-repudiation:** is a service which prevents an entity from denying previous commitments or actions. When disputes arise due to an entity denying that certain actions were taken, a means to resolve the situation is necessary. For example, one entity may authorize the purchase of property by another entity and later deny such authorization was granted. A procedure involving a trusted third party is needed to resolve the dispute.

## 3.3 Authentication

Authentication is the process of establishing confidence in the truth of some claim. The claim could be any declarative. Because access is typically based on the identity of the user who requests the resource, authentication is essential to effective security. A fundamental requirement of any secure system is the authentication of a valid user to the system. Access control is put into play only after the user is authenticated. Strong access control is meaningless without strong authentication. In recent years, authentication procedures have been increasing to ensure security/legitimacy of the individual. Simultaneously, users are becoming increasingly worried about infringement of their privacy, as they fear authorities are tracking their whereabouts and activities. The notion of anonymous authentication is considered to achieve this goal. Anonymous authentication is a means of authorizing a user without identification. The technology serves as a breakthrough to enhance the privacy of the user and yet to preserve the security of the system.

### 3.3.1 Factorized Authentication

An authentication factor is a piece of information and process used to authenticate or verify the identity of an entity requesting access under security constraints. Factors [1] are generally classified into three classes (in the order of strength of authentication):

**The Ownership Factors:** Something the user has (e.g., wrist band, ID card, security token, software token, phone, or cell phone).

**The Knowledge Factors:** Something the user knows (e.g., a password, pass phrase, or personal identification number (PIN)).

**The Inherence Factors:** Something the user is or does (e.g., fingerprint or retinal pattern, DNA sequence (there are assorted definitions of what is sufficient), signature or voice recognition, unique bio-electric signals, or another biometric identifier).

Two-factor authentication ( $T-F A$ ) is a system wherein two different factors are used in conjunction to authenticate. Using two factors as opposed to one factor generally delivers a higher level of authentication assurance. Multi-factor authentication is an extension to two-factor authentication.

Additionally other authentication factors include for example these categories: Location-based authentication, such as that employed by credit card companies to ensure a card is not being used in two places at once. Time-based authentication, such as only allowing access during normal working hours. Normally such authentication factors apply with individuals in conjunction with physically carried authentication factors.

## 3.4 Authentication Protocols: Related Work

In this chapter, we survey the literature on works related to our thesis. They serve as a good background on various security goals and notions, current state-of-art technology, similarities and differences among schemes. We hope that after reading this chapter, the readers can better understand the incentives that drive the writing of this thesis, and at the same time better evaluate the contribution of this thesis. We group these protocols into Wireless LAN Authentication Protocols, and Wireless Mobile Authentication Protocols.

### 3.4.1 Wireless Authentication Protocols

The IEEE 802.11 standard [12] for wireless LAN communications introduced the Wired Equivalent Privacy (WEP) protocol in an attempt to bring the security level of wireless networks closer to that of wired ones. However, the primary goal of WEP1 is only to protect the communications between mobile devices and access points using a pre-shared secret key (PSK) without specifying how to establish the key. IEEE 802.1x [13] is a specification for port-based authentication for wired networks. It has been extended for use in wireless networks. It provides user-based authentication, access control and key transport. 802.1x is designed to be flexible and extensible. It relies on Extensible Authentication Protocol (EAP) [14] for authentication, which was originally designed for Point-to-Point Protocol (PPP) but was reused in 802.1x. EAP is extensible; hence it can be use any authentication mechanism. It operates at the Network Layer (Layer 3) rather than the Data Link (Layer 2) which contributes to the flexibility of the protocol.

*EAP-MD5* is a very basic EAP type that is the equivalent of CHAP [15] in the wired networks. Its advantages are ease of implementation and the fact that its a legacy solution

on many networks. Like CHAP, MD5 is a one-way authentication method. In other words, it authenticates the client to the authentication server, but not the authentication server to the client, rendering the connection susceptible to man-in-the-middle attacks. In addition, EAP-MD5 offers no key management or dynamic key generation. *EAP-LEAP*, like EAP-MD5, LEAP accepts a username and password from the client and transmits them to the authentication server. What sets LEAP apart from EAP-MD5 are the extra security features such as dynamic key generation and mutual authentication. Unfortunately, the password-only protocols (EAP-MD5 and EAP-LEAP) turned out to be completely insecure against offline dictionary attacks [16]. That is due to the fact password is something chosen from a relatively small dictionary rather than a large range of cryptographically secure (high-entropy) key. As a result, even a passive (not active) attacker who eavesdrops and records the communications can enumerate, off-line in parallel, the password candidates in a dictionary of size, this dictionary being rather likely to include the user password. Consequently, designing a secure password-based authentication is not trivial.

In *Secret-Key* authentication methods, the access server (*AS*) and the client have the same secret and establish trust by proving to each other the knowledge of the shared secret key. (*Because the same secret key is shared between the authenticating parties, secret-key methods are also known as shared-key or symmetric-key methods.*) *Secret-key* authentication protocols are efficient and require little computational power. This advantage is especially important in *WLAN* because many wireless devices, such as *PDA*s and mobile phones, have little computational power.

*Secret-key* authentication methods have several drawbacks, however. Unlike in *Wired LAN*, in *WLANs* it is easy to eavesdrop on the communications between the authentication server and the client. Because most secret-key authentication protocols derive the shared secret from the user's password, and because most users choose bad password, it is easy for the attacker to gather enough encrypted message extract the secret key from them, using dictionary attacks [10, 11]. Although some secret-key authentication methods such as *EAP-SRP*, do protect the client's password from dictionary attacks, these methods require much greater computational power than other secret-key methods. Moreover, it is hard to securely distribute the shared secret to both parties. In the following we discuss and compare the most relevant secret-key authentication protocols that are being used for

WLAN authentication: *Lightweight Extensible Authentication Protocol (LEAP)*, *Kerberos*, *EAP over Secure Remote Password*, etc.

*Lightweight Extensible Authentication Protocol* [17, 6] was developed by Cisco. *LEAP* does not meet the level of security that *WPA* and *802.11i RSN* provide. Although *LEAP*'s authentication process provide mutual authentication and session key derivation, *LEAP* has some flaws. *LEAP* does not protect the client's identity because *EAP* identity messages are sent in plaintext. Moreover, *LEAP* is vulnerable to dictionary attacks [11] and does not consider desired properties such as delegation and fast reconnect. *Kerberos* [18], developed at *MIT* in 1993, is an authentication protocol designed for *TCP/IP*. Although *Kerberos* provide mutual authentication, fast reconnect (Limited) and delegation, it still vulnerable to dictionary attacks and does not protect client's identity. *EAP-SRP*, proposed by Wu [19], is a secret-key protocol which resists to dictionary attack using temporary asymmetric keys that are based on the shared symmetric key. However, *EAP-SRP* does not provide delegation and does not protect client's identity. The table below (Table 6.1) shows whether these secret-key protocols satisfy each of these properties.

Table 3.1: Summary of Secret-Key Methods

	LEAP	Kerberos	EAP-SRP
<i>Mutual Authentication</i>	Yes	Yes	Yes
<i>Identity Privacy</i>	No	No	No(Limited)
<i>Replay Attack Resistance</i>	Yes	Yes	Yes
<i>Dictionary Attack Resistance</i>	No	No	Yes
<i>Strong Per-Session Key</i>	Yes	Yes	Yes
<i>Delegation</i>	No	Yes	NO
<i>Fast Reconnect</i>	No	Limited	Limited
<i>Context-Aware</i>	No	No	No

Unlike the secret-key approach, the public-key approach uses a mathematically connected key pair, a public key and a private key. The public key approach is also known as the asymmetric key approach. To insure the client's public key is legitimate and to prevent an imposter from advertising his public key, the legitimates involved parties need to establish trust, typically through *Certification Authorities (CA)*, i.e. trusted independent third party that issues certificates. The requirement of well-implemented CAs makes most

public-key methods considerably more complicated to deploy than the secret-key methods. In the following we discuss and compare the most relevant public-key authentication protocols that are being used for *WLAN* authentication: *Extensible Authentication Protocol over Transport Layer Security (EAP-TLS)*, Identity-based authentication.

*IETF RFC 2716* [20] defines EAP-TLS. It is based on a certificate approach, and requires trusted CAs. TLS is a standardized version of the *Secure Socket Layer (SSL)* protocol, which was developed by Netscape. EAP-TLS extends EAP to provide certificate-based authentication for WLANs. Although EAP-TLS supports mutual authentication and resists replay and man-in-the-middle attacks, it does not provide a way of delegation, it does not provide identity privacy and some argue that most users do not understand or use the certificates properly. Moreover, EAP-TLS does not provide a way to authenticate clients who do not have a certificate that are signed by the CAs. As mentioned earlier, a certificates-based protocol, such as TLS, is hard to implement due to the requirement of CAs. *Identity-Based* cryptography takes advantages of public-key authentication without the complication of certificates. ID-based Crypto has the potential to simplify revocation and delegation, but missing features, such as lack of implementation and lack of session key derivation, make it an inappropriate choice for securing WLAN. The protocol proposed by Lee et al. [21] extends EAP to use ID-Based cryptography in the WLAN authentication process. However, session key derivation and identity privacy concerns are not mentioned in their paper and the authentication protocol is vulnerable to replay attacks.

The table below (Table 3.2) shows whether these public key protocols satisfy each of the desired properties.

Unlike symmetric and asymmetric approaches, tunnelled approach uses two phases. The first phase use the derived session key to establish an encrypted tunnel to encrypt their data communication. In the second phase, the authentication process will take place through the encrypted tunnel. Two tunnelled methods have been proposed: *Protected EAP (PEAP)* [22] and *EAP-Tunnelled TLS (EAP – TTLS)* [23]. The tunnel has two purposes. First it allows use of a less secure legacy protocol for client authentication in the second phase. Second, using the tunnel hides the client's identity privacy from an eavesdropper by hiding the EAP response-Identity message using the encrypted tunnel. Although these tunnelled approaches provide identity privacy, delegation and protection

Table 3.2: Summary of Public-Key Methods

	EAP-TLS	ID-Based
<i>Mutual Authentication</i>	Yes	Yes
<i>Identity Privacy</i>	No	No
<i>Replay Attack Resistance</i>	Yes	No
<i>Dictionary Attack Resistance</i>	Yes	Yes
<i>Strong Per-Session Key</i>	Yes	No
<i>Delegation</i>	No	Yes
<i>Fast Reconnect</i>	No	No
<i>Context-Aware</i>	No	No

against replay and dictionary attacks, PEAP and EAP-TTLS suffer from man-in-the-middle attack as it was recently discovered by Asokan et al. [24]. Moreover, PEAP and EAP-TTLS do not provide fast reconnect.

The table below (Table 3.3) shows whether these tunnelled protocols satisfy each of the desired properties.

Table 3.3: Summary of EAP-Tunnelled Methods

	PEAP	EAP-TTLS
<i>Mutual Authentication</i>	Yes	Yes
<i>Identity Privacy</i>	Yes	Yes
<i>Replay Attack Resistance</i>	Yes	Yes
<i>Dictionary Attack Resistance</i>	Yes	Yes
<i>Strong Per-Session Key</i>	Yes	Yes
<i>Delegation</i>	Yes	Yes
<i>Fast Reconnect</i>	No	No
<i>Context-Aware</i>	No	No

### 3.4.2 Mobile Authentication Protocols

Key agreement is one of the fundamental cryptography primitives. This required in situations where two or more parties want to communicate securely among themselves. Key agreement protocols fall naturally into two classes authenticated and unauthenticated. A wide variety of cryptographic authentication schemes and protocols have been developed

to provide authenticated key agreement to prevent man-in-the-middle, replay attack, forward secrecy, etc.

*2-Party Key Agreement Protocols:* The first two-key agreement protocol was introduced by Diffie-Hellman in [25]. It is an unauthenticated protocol in the sense that an adversary who has control over the channel can use man-in-the-middle attack to agree upon two separate keys with the two users without the users being aware of this. This was modified into an authenticated key agreement protocol by Matsumoto et al. [26], which was in turn showed to be insecure [27]. In 1999, Seo et al. [28] proposed a simple authenticated key agreement protocol (*SAKA*) for wireless mobile communications. The proposed protocol required 3 rounds in order to establish authentication process and to agree on the secret session key. However, *SAKA* protocol, as listed in [29, 30], is vulnerable to impersonate attack and does not provide perfect forward secrecy nor identity authentication. In 2001 [31], an anonymous authentication protocol was proposed for mobile devices to roam anonymously on distributed wireless networks. Their protocol is targeted to protect the mobile device identity from all entities other than its home server and the visiting foreign server. However, according to [32], it is found that a malicious foreign server which is not serving the mobile device can launch an impersonate attack to reveal the mobile device identity. Most password-based authenticated key exchange protocols are based on Diffie-Hellman key exchange protocol. However, the limitations of a low-power device makes these schemes not suitable for imbalanced wireless networks because of the modular exponential operations. In 2002 [33], Zhu et al. proposed a password-based authenticated key exchange protocol based on RSA with short public exponents. Their protocol run challenge-response protocol to establish the session secret key. Zhu et al. claimed that the protocol is efficient for low-power devices in wireless networks and is secure against dictionary attacks. However, Bao [34] pointed out that the password protocol of Zhu et al. is subject to offline dictionary attack if entity's identity is too short. In [35] Yeh et al. proposed a notion of security against undetectable on-line password guessing attack and argued that Zhu et al.'s protocol is insecure against this undetectable attack. Moreover, Yeh et al. proposed an improved protocol to defend against this attack. In [36, 37], Zhang pointed out that Zhu et al.'s protocol is vulnerable to some form of off-line dictionary attacks. Recently, [38, 39, 36, 40, 41, 42] pointed out that Yeh et al.'s improvement

is vulnerable to the off-line dictionary attack. To avoid off-line dictionary attack existed in Yeh et al.'s improved protocol, Lo [38] and Yang-Wang [39] proposed two improved protocols. However, in [43] authors pointed out that the Lo proposed protocol is still vulnerable to an active off-line dictionary attack and the Yang-Wang protocol is vulnerable to a passive off-line dictionary attack.

In 2002, Chien et al. [44] proposed a remote user authentication scheme using smart cards. Chien et al. claimed that their proposed scheme has the merits of providing mutual authentication, freely choosing password, no verification table, and involving only a few hashing operations instead of the costly modular exponentiations. In 2004, Ku et al. [45], however, pointed out that Chien et al.'s scheme is vulnerable to a reflection attack, insider attack, guessing attack and is not reparable once a user's permanent secret is compromised. Ku et al. also proposed an improved scheme to resolve these security pitfalls. Nevertheless, in 2004, Yoon et al. [46] showed that Ku et al.'s scheme is still susceptible to parallel session attack and is insecure for changing the user's password, and also proposed an enhancement to Ku et al.'s scheme to overcome such problems.

Thereafter, in 2007, Wang et al. [47] showed that both Ku et al.'s scheme and Yoon et al.'s scheme were vulnerable to a guessing attack, forgery attack and denied service attack, as well as inefficiency in password authentication. By introducing the two-variant hashing operation, Wang et al. proposed an improved scheme to keep the merits of original schemes that can be easily realized in the practical resource limited environment. However, Wang et al.'s improved scheme does not provide perfect forward secrecy and is still vulnerable to a guessing attack and Denning-Sacco attack. Accordingly to [48], authors demonstrate that Wang et al.'s scheme does not provide perfect forward secrecy and is susceptible to the guessing attack and Denning-Sacco attack.

To simplify the *PKI* system, authors in [49] have introduced the new idea of ID-Based systems. The advantages of ID-Based cryptosystems is that it simplifies the key management process which is a heavy burden in PKI based cryptosystems. The first ID-based authenticated key agreement scheme based on Weil pairing was introduced by Smart [50] using Shamir's ID-based concept. However, Shim [51] pointed out that Smart's protocol does not provide full forward security and proposed his own protocol. Nonetheless, Shim's protocol still suffers from an important security flaw because it is not protected against the

man-in-the-middle attack [52]. In 2004, Ryu et al. [53] proposed a new ID-based protocol which is more efficient requiring only one pairing computation and two point multiplication. However, Yaun et al. [54] pointed out that the protocol is insecure under the key compromise impersonate attack.

The table below (Table 3.4) shows these protocols weaknesses.

Table 3.4: Summary of ID-Based Protocol

Protocol	Weaknesses
Smart [50]	Forward Secrecy
Shim [51]	Man-in-the-Middle
Ryu-Yoon-Yon [53]	Key Compromise Attack Reveal Attack
McCullag-Barreto [55]	Key Compromise Attack Reveal Attack
McCullag-Barreto Revised	Reveal Attack

Aydos et al. [56] proposed an ECC-based authentication key agreement protocol for wireless communications. In their protocol, they used ECDSA and Diffie Hellman Key agreement to provide authentication and to obtain a session key for later communications. Because their protocol is based on ECC, the protocol is suitable for mobile devices in which the computational power is low. However, Sun et al. [57] demonstrate that Aydos et al.'s ECC-based protocol does not achieve forward security, known-key security and mutual authentication.

### 3.5 Access Control

The purpose of access control is to limit the operations that a legitimate user of a computer system can perform. In this way, access control seeks to prevent activity that could lead to a breach of security. It is important to make a clear distinction between authentication and access control. Correctly establishing the identity of the user is the responsibility of the authentication service. Access control relies on and coexists with other security services in a computer system. Access control refers to limiting what users can do after they identify themselves and are authenticated. An access control request is a list of user identifications

(user names, group names, user roles, etc), and it is associated with a set of permissions that define the user's rights and privileges.

### 3.5.1 Access Control: Related Work

The following section perform a review of historical, current, and emerging work in access control. In the mid 1980s to mid 1990s, a number of alternate models have been proposed. Of particular significance was Role Based Access Control (*RBAC*), which still remains an active area of research nowadays. *RBAC* is an alternative to traditional discretionary (*DAC*) and mandatory access control (*MAC*). *RBAC* is designed to centrally manage users privileges, roles, operations and resources. Significant benefits of *RBAC* include the simplification of system administration, the enhancement of organizational productivity, reduction in employee downtime, enhanced systems security and integrity, and simplified regulatory compliance. There have been many efforts over the past years to define *RBAC* and work towards a unified *RBAC* standard. The first comprehensive framework for *RBAC* models was defined by Sandhu et al. [61, 62]. The framework consisted of four models of *RBAC*, that ranged from simple to complex:

**RBAC0:** The most basic *RBAC* model, where users are associated with roles, and permissions are associated with roles.

**RBAC1:** Builds on *RBAC0* by introducing role hierarchies.

**RBAC2:** Builds on *RBAC0* by introducing constraints such as separation of duties.

**RBAC3:** Combines *RBAC1* and *RBAC2* such that constraints can be applied to a hierarchy of roles. Since this initial family of models, there has been much research, which to a great extent has resulted in the establishment of a standard *RBAC* model.

Although the *RBAC* models vary from very simple to pretty complex, they all share the same basic structure of subject, role and privilege. Other factors such as time, location, etc. are not considered in making access control decision in these models. Of particular interest to our research is the adaptation of these models and definitions to context aware environments.

Later on, several extended access control schemes have been proposed. Temporal

RBAC (TRBAC) was introduced by Bertino et al. [63], which proposed the use of time-based constraints and activation dependencies for role activation. TRBAC provides support for periodic activation/deactivation of roles, and temporal dependencies which are expressed by means of role triggers. Generalized RBAC (GRBAC) was introduced by Convington et al. [64], to support context-awareness in a context-aware home environment. In this model of RBAC, two additional types of roles were introduced in addition to traditional subject roles, object and environment roles. While GRBAC provides an improvement in the flexibility of security policy, it introduces complexity in management of monitoring access control policies.

### 3.6 Privacy Consideration

Achieving information security sometimes requires the disclosure of personal information (for example, by requiring authentication). At the same time, insufficient privacy protection may mean that personal information about others is easily discovered, calling into question the reliability of authentication systems that depend on such information. While this report urges that care must be taken to avoid unnecessary authentication and identification (and therefore avoid unnecessary privacy risks), the interplay between achieving privacy protection and authentication security in the development of information systems should be carefully considered. Privacy and security, while often in tension, are complementary as well. Security can protect private data, and maintaining privacy can aid in avoiding security breaches. Usability is a key component of this mix, since hard-to-understand or hard-to-use systems will be prone to errors and may drive an individual to work around either the security mechanisms or the mechanisms that would protect privacy. Lessons learned in trying to create secure and usable systems therefore apply when seeking to develop systems that protect privacy. System designers, developers, and vendors should improve the usability and manageability of authentication mechanisms, as well as their intrinsic security and privacy characteristics.

## 3.7 Conclusion

Non-cryptographic authentication was generally adequate in the days before the Internet, when the user could be sure that the system asking for the password was really the system they were trying to access, and that nobody was likely to be eavesdropping on the communication channel to observe the password being entered. Traditional authentication methods are not suitable for use in computer networks where attacker monitor network traffic to intercept critical data. Nowadays, authentication is critical for security of computer systems. Without the knowledge of the identity of a principal requesting an operation, it is difficult to decide whether the operation should be allowed. The use of strong authentication method that do not disclose personal data is imperative.

# Bibliography

- [1] R. Sandhu and P. Samarati, *Authentication, access control, and intrusion detection*, In *The Computer Science and Engineering Handbook* (A. B. Tucker, ed.), pp. 1929-1948, CRC Press, 1997.
- [2] EPIC's RFID Systems, *Radio Frequency Identification (RFID) Systems*, Available at <http://epic.org/privacy/rfid>.
- [3] M. Ellen, *RFID Is Really Getting under People's Skin*, NetWorld Magazine, April 4, 2005.
- [4] A. Gilbert, *Privacy Questions Arise as RFID Hits Stores*, CNet News.com, September 30, 2004.
- [5] Trans-Atlantic Consumer Dialogue, *Resolution on Radio Frequency Identification (RFID)*, May 5, 2005.
- [6] Cisco, *Wireless LAN Security White Paper*, Tech Note, Available at <http://www.cisco.com/warp/public/cc/pd/witc/a01200ap/prodit/>, August 2002.
- [7] P. Abi-Char, B. EL-Hassan and A. Mhamed, *IEEE 802.11i Standard: Review and Security Analysis Using AVISPA*, In *Proceeding of the National Conference on Current Trends in the Theory and Applications of Computer Science, CTTACS07*, NDU, Lebanon, 2007.
- [8] P. Abi-Char, A. Mhamed, B. EL-Hassan, *An Efficient Authenticated Key Agreement Protocol*, In *Proceeding of the First International Conference on New Technologies, Mobility and Security, NTMS 2007, PARIS, France, May 2007*, pp. 45.

- [9] W. Thomas, *A Real-World Analysis of Kerberos Password Security*, In Proceedings of the 1999 Internet Society Network and Distributed System Security Symposium, February 1999.
- [10] Cisco, *Dictionary Attack on Cisco LEAP*, Tech Note, Available at <http://www.cisco.com/warp/public/707/cisco-sn-20030802-leap.shtml>, August 2003.
- [11] S. Wong, *The Evolution of Wireless Security in 802.11 Networks: WEP, WPA and 802.11 Standards*, SANS Institute, March 2003.
- [12] IEEE 802.1x, *Port-Based Network Access Control*, <http://www.ieee802.org/1/pages/802.1x.html>
- [13] Internet Engineering Task Force, *PPP Extensible Authentication Protocol (EAP)*, IETF Internet, RFC 2284, March 1998.
- [14] Internet Engineering Task Force, *Challenge Handshake Authentication Protocol (EAP)*, IETF Internet, RFC 1994, March 1996.
- [15] J. Wright, *Weaknesses in LEAP Challenge/Response*, <http://home.jwu.edu/jwright/presentations/asleap-defcon.pdf>
- [16] M. Cameron, *Cisco LEAP Protocol Description*, Available at <http://www.missl.cs.umd.edu/wireless/ethereal/leap.txt>, 2002.
- [17] K. John and B. Neurman, *The Kerberos Network Authentication Service (Version 5)*, IETF RFC 1510, September 1993.
- [18] W. Thomas, *The Secure Remote Password Protocol*, Tech Note, In Proceeding of the 1998 Internet Society Symposium on Network and Distributed Systems Security, pages 97-111, San Diego, CA, 1998.
- [19] B. Aboda and D. Simon, *PPP EAP-TLS Authentication Protocol*, IETF RFC 2716, October 1999.

- [20] L. Byang et al., *Mobile IP and WLAN with AAA Authentication Protocol using Identity-based Cryptography*, IN the 10th International Conference on Telecommunications ICT, volume 1, pages 597-603, 2003.
- [21] P. Ashwin et al., *Protected EAP Protocol (PEAP) Version 2*, IETF Internet Draft, draft-josefsson-pppext-eap-tls-eap-07.txt, 2003.
- [22] F. Paul and W. Blake, *EAP Tunnelled TLS Authentication Protocol (EAP-TTLS)*, IETF Internet Draft, draft-ietf-pppext-eap-ttls-03.txt, August 2003.
- [23] N. Asokna and N. Valtteri and N. Kaisa, *Man-in-the-Middle in Tunnelled Authentication Protocols*, Cryptology ePrint Archive, Report 2002/163, 2003.
- [24] W. Diffie and M. Hellman, *New Directions In Cryptography*, In Proceeding of IEEE Transactions on Information Theory, IT-22(6), pp. 644-654, November 1976.
- [25] T. Matsumoto et al., *On Seeking Smart Public-Key Distributions Systems*, In Proceeding of the Transactions of the IECE of Japan, E69(1986), pp. 99-106, 1986 .
- [26] L. Law et al., *An Efficient Protocol for Authenticated Key Agreement*, Technical Report CORR 98-05, Department of C & O, University of Waterloo, 1998. Available at [Citeseer.nj.nec.com/law98efficient](http://Citeseer.nj.nec.com/law98efficient).
- [27] S. Dong and P. Sweeney, *Simple Authenticated Key Agreement Algorithm*, Electronics Letters, vol. 35, Issue 13, pp. 1073-1074, 1999.
- [28] K. Wei-Chi and W. Sheng-De, *Cryptanalysis of Modified Authenticated Key Agreement Protocol*, Electronics Letters, vol. 36, Issue 21, pp. 1770-1771, October 2000.
- [29] B.T. Hsieh et al., *Cryptanalysis of Enhancement for Simple Authentication Key Agreement Algorithm*, Electronics Letters, vol. 38, Issue 1, pp. 20-21, 2001.
- [30] J. Go and K. Kim, *Wireless Authentication Protocol Preserving User Anonymity*, In Proceeding of the 2001 Symposium on Cryptography and Information Security (SCIS 2001), pp. 159-164, Jan. 2001.

- [31] W. Duncan and K. Hong, *Security Analysis of Two Anonymous Authentication Protocols for Distributed Wireless Networks*, In Proceeding of the 3rd International Conference on Pervasive Computing and Communications Workshops (PerCom 2005), pp. 284-288, 2005.
- [32] F. Zhu et al., *RSA-Based Password Authenticated Key Exchange for Imbalanced Wireless Networks*, In Proceeding of the 5th Information Security Conference 2002, Lecture Notes in Computer Science, Springer-Verlag, vol 2433, pp.150-161.
- [33] F. Bao, *Security Analysis of a Password Authenticated Key Exchange Protocol*, In Proceeding of the 6th Information Security Conference 2003, Lecture Notes in Computer Science, Springer-Verlag, vol 2851, pp. 208-217.
- [34] H. Yeh et al., *Improvement of Password Authenticated Key Exchange Based on RSA for Imbalanced Wireless Networks*, IEICE Transactions Commun, vol. E86-B, pp. 3278-3282, Nov. 2003.
- [35] M. Zhang, *Breaking an Improved Password Authenticated Key Exchange Protocol for Imbalanced Wireless Networks*, IEEE Commun, Letter, vol. 9, pp. 276-278, Mar. 2005.
- [36] M. Zheng, *Further Analysis of Password Authenticated Key Exchange Protocol based on RSA for Imbalanced Wireless Networks*, In Proceeding of the 7th Information Security Conference 2004, Lecture Notes in Computer Science, Springer-Verlag, vol 3225, pp. 13-24.
- [37] J. W.Lo, *The Improvement of YSYCT Scheme for Imbalanced Wireless Network*, International Journal Network Security , vol 3, pp. 39-43, July 2006.
- [38] C. Yang and R. Wang, *Cryptoanalysis of Improvement of Password Authenticated Key Exchange Based on RSA for Imbalanced Wireless Networks*, IEICE Transactions Commun, vol. E88-B, pp. 4370-4372, Nov. 2005.
- [39] S. Wang and F. Bao and J. Wang, *Security Analysis on an Improvement of RSA based Password Authenticated Key Exchange*, IEICE Transactions Commun, vol. E88-B, pp. 1641-1646, Apr. 2005.

- [40] E. Yoon and K. Yoo, *Cryptoanalysis of Password Authenticated Key Exchange Based on RSA for Imbalanced Wireless Networks*, IEICE Transactions Commun, vol. E88-B, pp. 2627-2628, June. 2005.
- [41] Y. Chang et al., *An Efficient Password Authenticated Key Exchange Protocols for Imbalanced Wireless Networks*, Computers Standards & Interfaces, vol. 27, pp. 313-322, Mar. 2005
- [42] C. Tianjie and L. Dongdai, *Cryptoanalysis of Two Password Authenticated Key Exchange Protocols Based on RSA*, IEEE Communications Letter, vol 10, No. 8, pp. 623-625, August 2006.
- [43] H.Y. Chien and J.K. Jan and Y.M. Tseng, *An Efficient and Practical Solution to Remote Authentication Smart Card*, Computer and Security, vol. 21, no. 4, 2002, pp. 372-375.
- [44] W.C. Ku and S.M. Chen, *Weaknesses and Improvements of an Efficient Password Based Remote User Authentication Scheme Using Smart Card*, IEEE Transactions on Consumer Electronics, vol. 50, no. 1, 2004, pp. 204-207.
- [45] E.J. Yoon and E.K. Ryu and K.Y. Yoo, *Further improvement of an efficient password based remote user authentication scheme using smart cards*, IEEE Transactions on Consumer Electronics, vol. 50, no. 2, 2004, pp. 612-614.
- [46] X.M. Wang, W.F. Zhang, J.S. Zhang and M.K. Khan, *Cryptanalysis and improvement on two efficient remote user authentication scheme using smart cards*, Computer Standards & Interfaces, vol. 29, no. 5, 2007, pp. 507-512.
- [47] E.J. Yoon, E.J. Lee and K.Y. Yoo, *Cryptanalysis of Wang et al.'s Remote User Authentication Scheme Using Smart Cards.*, In Proceedings of the Fifth International Conference on Information Technology: New Generations, 2008, pp. 575-580
- [48] A. Shamir, *Identity-based Cryptosystems and Signature Schemes*, In Advanced in Cryptology- Crypto'84, LNCS 196, pp. 47-53, Springer-Verlag, 1984.

- [49] N.P. Smart, *An ID-Based Authentication Key Agreement Protocol Based on the Weil Pairing*, Electron. letter, 38(13), pp. 630-632, 2002.
- [50] K. Shim, *Efficient ID-Based Authentication Key Agreement Protocol Based on the Weil Pairing*, Electron. letter, 39(8), pp. 653-654, 2003.
- [51] H. Sun and B. Hsieh, *Security Analysis of Shim's Authenticated Key Agreement Protocols from Pairings*, Cryptology ePrint Archive, Report 2003/113, 2003. <http://eprint.iacr.org/2003/113>.
- [52] E. Ryu and E. Yoon and K. Yoo, *An Efficient ID-Based Authenticated Key Agreement Protocol*, Networking 2004 Volume 3042, 2004.
- [53] Q. Yuan and S. Li, *A New Efficient ID-Based Authenticated Key Agreement Protocol*, IEEE Communications Letter, vol 10, No. 8, pp. 623-625, March 1, 2005.
- [54] N. McGullagh and P. Barreto, *A New Two-Party Identity-Based Authenticated Key Agreement*, Cryptology ePrint Archive, Report 2004/122, 2004. In Proceeding of CT-RSA 2005. <http://eprint.iacr.org/2004/122>.
- [55] C.K. Koc and M. Aydos and B. Sunar *An Elliptic Curve Cryptography based authentication and key Agreement protocol for Wireless Communication*, In 2nd International Workshop on Discrete Algorithm and Methods for Mobile Computing and Communications, 1998.
- [56] H.M. Sun and B.T. Hsieh and S.M. Tseng *Cryptanalysis of Aydos et al.'s ECC-based Wireless Authentication Protocol*, In Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04), pp. 563-566.
- [57] T.K. Stephen, and L.I. Millett, *Who Goes There?: Authentication Through the Lens of Privacy*, Committee on Authentication Technologies and Their Privacy Implications, National Research Council, <http://www.nap.edu/catalog/10656.html>, 2003.
- [58] R. Sandhu, S. Gavrila, R. Kuhn, F. Ferraiolo, and R. Chandramouli *Proposed NIST Standard for Role-Based Access Control*, In Proceeding of ACM Transactions on Information and System Security, 4(3):224-274, 2001.

- [59] R. Sandhu, H. Feinstein, E. Coyne, and C. Youman *Role-Based Access Control Models*, In Proceeding of IEEE Computer, 29(2):38-47, 1996.
- [60] E. Bertino, P. Bonatti, and E. Ferrari *TRBAC: A Temporal Role-Based Access Control Model*, In Proceeding of ACM Transactions on Information and System Security, Volume 4, pp. 191-223, 2001.
- [61] M.J. Moyer, M.J.Covington, and M.Ahamad, *Generalized Role-Based Access Control for Secure Future Applications*, In Proceeding of the 23rd National Information Systems Security Conference, USA, 2000.
- [62] R.M. Needham, and M.D. Schroeder, *Using encryption for authentication in large networks of computers*, In Proceeding of Communications of the ACM, vol. 21, no. 12, pp. 993-999, 1978.
- [63] D.K. Gifford, *Cryptographic sealing for information secrecy and authentication*, In Proceeding of Communications of the ACM, vol. 25, no. 4, pp. 274-286, 1982.
- [64] A. Menezes, P. Van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, In The CRC Press, <http://www.cacr.math.uwaterloo.ca/hac/>, 2nd edition, 1996.

## Chapter 4

# PROPOSED PROTOCOLS AND EVALUATION

In this chapter we look at scenarios where the user must be authenticated towards some mobile environments. First we give an introduction to some security properties relevant for the rest of this chapter in Section 4.1. In Section 4.2 we give an overview of our previous proposed solutions that have been done in order to improve usability of mobile computing. This part of the chapter is based on three accepted papers, two presented at the IAS2007 international conference [6, 7] and the third presented at the NGMAST2007 international conference [8]. Finally, in Section 4.3 we look at the problem of privacy and anonymity in mobile computing, and propose an extended protocol for reducing that risk by presenting an identity-based authenticated key agreement protocol from pairings. This part is based on an accepted paper presented at the CRISIS2008 international conference [9] as well as some additional work.

### 4.1 Introduction

The increasing development in wireless mobile communications has attracted an important amount of attention on the security, anonymity and privacy issues. To provide secure communications, authenticated key agreement protocols are crucial primitive for establishing secure session keys. Achieving a secure group of communications is an important

issue for mobile environment. A group key agreement protocol enables a group of communicating entities over an intrusted network to establish a secure shared key. Anonymous authentication is a means of authorizing a user without revealing his/her identification. Mobile technologies such Personnel Digital Assistant (PDAs) and mobile phone systems are increasingly being deployed in pervasive computing. These mobile devices have raised public concern regarding violation of privacy, anonymity and information confidentiality. Considering these concerns, there is a growing need to discover and develop techniques and methods to overcome the threats described above. In this chapter we propose several architectures which enhance the privacy and anonymity of users in ubiquitous computing and yet preserve the security requirements of the system. Our proposed protocols are based on different cryptographic techniques including Elliptic Curve techniques, MapToCurve and MapToPoint functions, Weil Pairing techniques and elliptic curve based Identity Schemes. In addition, we present a formal validation of our protocol by using the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool. The main comparative study of our proposed architectures is to offer significantly improved performance in computational and communication load over comparably many authenticated key agreement protocols. Moreover, Our proposed architectures achieve many of desirable security requirements

## 4.2 Key Management

Key establishment refers to the situation where network users employ an inter-active protocol to construct a shared secret key called session key. This session key can then be used to achieve some cryptographic goal such as confidential communication channel between entities or data integrity. There are two kinds of key establishment protocols: Key transport protocols in which a key is created by one entity and securely transmitted to the second entity, and Key agreement protocols in which both parties contribute information which jointly establish the shared key [2]. A key agreement protocol is said to provide implicit key authentication if entity A is assured that no other entity aside from a specifically identified second entity B can possibly learn the value of a particular secret key. A key agreement

protocol which provides implicit key authentication to both entities is called an authenticated key agreement protocol. If both implicit key authentication and key confirmation are provided, then the key establishment protocol is said to provide explicit key authentication. A key agreement protocol which provides explicit key authentication to both entities is called an authenticated key agreement with key confirmation [2]. In this paper we will consider the case of key agreement protocol with symmetric two-entities setting. The idea of cryptographic challenge-response protocols is that one entity (the claimant) proves its identity to another entity (the verifier) by demonstrating knowledge of a secret known to be associated with that entity, without revealing the secret itself to the verifier during the protocol. This is done by providing a response to a time-variant challenge, where the response depends on both the entity's secret and the challenge. The challenge is typically a number chosen by one entity (randomly and secretly) at the outset of the protocol.

Apart from authentication, the other aspects of key agreement protocols are computational and communication efficiency. In key agreement protocols, all users should be able to agree upon a common secret key. The total number of bits exchanged in the protocol is a crucial parameter in judging the efficiency of the protocol. Further, in each round, user has to perform some computational like an exponentiation or a scalar multiplication. The total amount of computational required by all the users is another measure of efficiency of the protocol.

### 4.3 Desirable Properties for key agreement protocols

A number of desirable properties for key agreement protocols have been identified [3] and nowadays most of the protocols are analyzed using these properties which are described below:

**Known-key security:** Each run of a key agreement protocol between two entities A and B should produce a unique shared secret key called session key  $K_s$ . A protocol should still achieve its goal in the face of an adversary who has learned some other session key.

**Perfect forward secrecy:** If long-term private keys of one or more entities are compromised, the secrecy of previous session keys established by honest entities is not affected.

**Key-compromise impersonation:** Suppose that A's long-term private key is disclosed.

Clearly an adversary that knows this value can now impersonate A, since it is precisely this value that identifies A. However, it may be desirable that this loss does not enable an adversary to impersonate other entities to A.

**Unknown key-share:** Entity A cannot be coerced into sharing a key with entity B without A's knowledge, i.e., when A believes the key is shared with some entity  $C \neq B$ , and B (correctly) believes the key is shared with A.

**Key control:** No other entity should be able to force the session key to a preselected value.

In addition, authentication protocols should have other properties which are related to performance. Because round trips and large blocks are critical factors in terms of communication load and because exponentiations and random numbers are to be critical factors in terms of computation load, such properties are listed below:

**Computational efficiency:** this includes the number of operations required to execute a protocol. In order to achieve this property, the protocol should have the minimum number of operation as possible.

**Communication efficiency:** This includes the number of passes (message exchanges) and the bandwidth required (total number of bits transmitted).

Other desirable properties are:

**Nature of security guarantees:** including provable security and zero-knowledge properties.

**Storage of secrets:** This refer to the location and the method used (e.g., software only, local disks, hardware tokens, etc.) to store critical keying material.

Moreover, to protect the user privacy and anonymity, we consider the following requirement in cryptography point of view, [4, 5].

**Data Confidentiality:** The private information of an Embedded Device ( $ED$ ) must be kept secure to guarantee user privacy. The information of  $ED$  must be meaningless for its bearer even though it is eavesdropped by an unauthorized Reader ( $R$ ).

**Anonymity:** Although the data of  $ED$  is encrypted, the unique identification information of  $ED$  is exposed since the encrypted data is constant. An attacker can identify each  $ED$  with its constant encrypted data. Therefore, it is important to make the information of  $ED$  anonymous.

**Location Privacy:** Neither the system nor the users of the system will be able to know the exact location of a user, unless that user decides to disclose such information or if another person physically sees that user at that location.

**Data Integrity:** If the memory of  $ED$  is rewritable, forgery and data modification will happen. Thus, the linkage between the authentication information and  $ED$  itself must be given in order to prevent the simple copy for  $ED$

**Mutual Authentication and Reader Authentication:** In addition to access control, the mutual authentication between  $ED$  and the back-end server ( $DB_{ID}$ ) must be provided as a measure of trust. By authenticating mutually, the replay attack the man-in-middle attack to both  $ED$  and  $DB_{ID}$  is prevented.  $DB_{ID}$  must also authenticate  $R$  to avoid the man-in-the-middle attack by an illegitimate  $R$  on the insecure channel.

## 4.4 Closely Related Work

Key agreement is one of the fundamental cryptography primitives. This is required in situations where two or more parties want to communicate securely among themselves. Key agreement protocols fall naturally into two classes authenticated and unauthenticated. A wide variety of cryptographic authentication schemes and protocols have been developed to provide authenticated key agreement to prevent man-in-the-middle and other relevant attacks.

In [6], we present a new and efficient three-pass authenticated key establishment protocol that provides secure mutual authentication and key agreement with key confirmation. Our proposed protocol, named *SAKA*, is based on the challenge and response in the Secret-key setting, on KAS (Simplified Station-to-Station) scheme and on the Diffie-Hellman Key Predistribution [1, 2]. According to [6], the proposed protocol achieves the desirable security requirements and performances and it establishes a shared secret key  $K$  between the two entities.

In the following, we briefly describe the proposed protocol: The public domain parameters consist of a group  $(G, \cdot)$  and an element  $\alpha$  where  $\alpha \in G$  having order  $n$ , each user  $T$  has a secret exponent  $u_T$ , where  $0 \leq u_T \leq n - 1$  (where  $n$  is a large prime number and All computation are performed modulo  $n$ ) and a corresponding Public Key  $b_T = \alpha^{u_T}$ .

In addition Alice chooses a password  $P$  and computes:  $K_H = h(P||ID(Alice))$  and  $b_s$ . Two versions of SAKA protocol were proposed. These two versions are SAKA-v1 and SAKA-v2. For SAKA-v1,  $b_s$  is equal to  $\alpha^{K_H}$  while for SAKA-v2,  $b_s$  is equal to  $K_H$ . The SAKA-v2 could be proposed and implemented if the server, Bob, is well protected and unauthorized access is denied. Finally, Alice notify Bob in a secure way about  $b_s$ . Then Bob store  $b_s$  in a secure database server. The proposed protocol consists of three flows and is defined as follow:

Within the first flow, Bob chooses a random challenge  $u_b$ , where  $1 \leq u_b \leq n - 1$ , then he computes:  $b_b = \alpha^{u_b} + b_s$  and finally he sends  $b_b$  to Alice. Within the second flow, Alice chooses a random challenge  $u_a$ , where  $1 \leq u_a \leq n - 1$ , then Alice computes:  $b_a = \alpha^{u_a}$  computes  $b_s$  and computes  $K$ , where  $K = [b_b - b_s]^{u_a}$ . Also Alice computes  $K_h$  where  $K_h = MAC_K(b_s||K)$ , and computes:  $Y_1 = MAC_{K_h}(ID(Alice)||b_b||b_a)$ . Finally she sends  $Y_1$  and  $b_a$  to Bob. Within the third flow, Bob computes:  $K = [b_a]^{u_b} = \alpha^{u_a u_b}$  and computes  $K_h$  where  $K_h = MAC_K(b_s||K)$ . Also Bob computes:  $Y'_1 = MAC_{K_h}(ID(Alice)||b_b||b_a)$ . Bob can then verify the value of  $Y'_1$  by checking that  $(Y'_1 == Y_1)$  If so, Bob authenticates Alice. Furthermore, if  $Y'_1$  and  $Y_1$  are equal, Bob can be confirmed that Alice has actually established the same shared  $K$  with him because the value of  $K_h$  used in  $MAC$  is derived from the shared key  $K$ . Then Bob computes:  $Y_2 = MAC_{K_h}(ID(Bob)||b_a)$  and finally he sends  $Y_2$  to Alice. In order to authenticate Bob, Alice will compute:  $Y'_2 = MAC_{K_h}(ID(Bob)||b_a)$  and then Alice will verify the value of  $Y'_2$  by checking that  $(Y'_2 == Y_2)$ , if so, if they match, then Alice authenticates Bob and Alice can be confirmed that Bob has actually established the same shared  $K$  with her. Finally, Alice and Bob agree on the common session key  $K_s$  where  $K_s = MAC_{K_h}(ID(Alice)||ID(Bob)||K)$ . Both sides will agree on the session Key  $K_s$  if all steps are executed correctly. Once the protocol run completes successfully, both parties may use  $K_s$  to encrypt subsequent session traffic in order to create a confidential communication channel.

In addition to a complete security analysis presented in their paper [6], we compare their proposed protocols SAKA-v1 and v2 with the following protocols: Leakage-Resilient Authenticated Key Exchange (LR-AKE) protocol [12], Simple Key Agreement (SKA) protocol [13], Secure Remote Password (SRP) protocol [14], Simple Password Exponential

Key Exchange (B-SPEKE) protocol [15], Password-Authenticated Key Exchange (PAK-X [16] and PAK-RY [17]) protocols and Authentication Memorable Password (AMP) protocol [18]. The comparison is done in terms of number of rounds, random numbers, exponentiations, and hash functions. Table 4.1 shows the compared result for number of rounds, and exponentiations. Table 4.2 shows the compared result for random numbers and hash functions numbers

Table 4.1: SAKA-Comparison of Performance-1-

Protocol	Rounds	Exponentiations		
		Client	Server	Total
B-SPEKE	4	3	4	7
SRP	4	3	3	6
AMP	4	2	3	5
PAK-RY	3	5	4	9
PAK-X	3	5	4	9
SKA	3	2	3	5
LR-AKE	3	3	2	5
EC-AKE	4	2	2	4
EC-SRP	3	2	2	4
SAKA-v1	3	3	2	5
SAKA-v2	3	2	2	4

It is clear from Table 4.1 that the SAKA protocol has the minimal cost in terms of number of steps, and exponentiations compared with the previous protocols. It can be easily noticed that B-SPEKE, SRP and AMP require 4 rounds while PAK-RY, PAK-X, SKA, LR-AKE and SAKA (v1 and v2) require 3 rounds. In addition, the computational load was clearly improved using SAKA-v2 protocol because, as noted in table 4.2, SAKA-v2 requires four exponentiations, two for the client and two for the server, while the other protocols, including SKA and LR-AKE, require at least 5 exponentiations. Although SAKA-v1 requires 5 exponentiations, it shows better performance. The SAKA-v1 shows better performance in terms of computational load over B-SPEKE, SRP, PAK-RY, PAK-X and it is equal with SKA and LR-AKE. SAKA-v1 shows better performance over SKA because there is no revealed data as the case with SKA where  $X_A$ ,  $X_B$  and  $W$  are sent in clear-text.

From Table 4.2, it can be easily noticed that the SAKA (v1 or v2) protocol requires 2 random numbers and 9 hash functions while PAK-X requires more. SAKA (v1 or v2) also

Table 4.2: SAKA-Comparison of Performance-2-

Protocol	Random Numbers	Hash Function Numbers
SRP	2	6
AMP	2	9
PAK-RY	3	8
PAK-X	3	10
SKA	2	7
LR-AKE	2/4	6
SAKA-v1	2	9
SAKA-v2	2	9

requires two more hash functions than SKA protocol due to the two  $MAC$  computations of  $K_h$  which were necessary to bring more security and robustness to our proposed protocols. In addition, for the SRP and LR-AKE protocols, it can be easily noticed that the proposed protocols (v1 and v2) require one more hash function because, from SRP and LR-AKE schemes, the two entities did not agree on a common session key  $K_s$ , as in the case of proposed protocols; SRP and LR-AKE just agreed on the shared key  $K$ .

In [7], we have proposed another new and efficient EC-DSA-based three-pass authenticated key establishment protocol, named EC-SAKA, that provides secure mutual authentication and key agreement with key confirmation. The EC-SAKA is based on the Elliptic Curve Cryptography [1], on SKA (Simple Key Agreement) protocol [13] and on the assumption that the ECC discrete logarithm problem is secure [1]. The proposed protocol achieves many of desirable security requirements and performances.

In the following, we will briefly describe the proposed protocol: Before running the authentication procedure, the client, Alice, select an elliptic curve  $E(Z_p)$  defined on  $Z_p$ . Alice chooses a random point over the elliptic curve called  $P$  with order  $n$ .  $n$  is a large prime number. In addition, Alice chooses a password  $pw$ , computes  $x = h(pw)$  and calculates  $Q$  where  $Q = x * P$ . Finally, Alice generates strong number  $p$  and  $q$  where  $p = 2 * q + 1$ . Once the following parameters  $(E, Q, P, p, q, pw)$  are generated, Alice transfers the  $(E, Q, P, n)$  to the server in a secure way. The protocol is defined as follow:

Within the first flow, Bob chooses a random challenge  $b$ , where  $1 \leq b \leq n - 1$ , then he calculates the point  $B$  where  $B = b * P + Q$ . Finally he sends  $B$  to Alice. Within the second flow, Alice chooses a random challenge  $a$ , where  $1 \leq a \leq n - 1$ , then computes  $A$  where

$A = a * P = (x_A, y_A)$  and calculates  $\alpha$  where  $\alpha = a(B - Q)$  and  $K = Q + \alpha$ . In addition, Alice calculates  $r = (x_A) \bmod(n)$  and computes  $i = a^{-1}(h(\alpha) + x * r) \bmod(n)$ . Finally  $(A, i)$  becomes the signatures pair and Alice transfers  $A$  and  $i$  to the server. Within the third flow, Bob computes  $\beta = b * A$ , Computes  $K = Q + \beta$ , computes  $w = i^{-1} \bmod(n)$  and calculates  $u_1 = (h(\beta) * w) \bmod(n)$ , and  $u_2 = (x_A * w) \bmod(n)$ . In addition, Bob calculates  $u_1 * P + u_2 * Q = (x_0, y_0)$  and calculates  $v = x_0 \bmod(n)$ . Bob checks if  $(v == x_A)$ , so Bob authenticates Alice and Bob can be confirmed that Alice has actually established the same shared session key. Then Bob computes:  $Y_B = h(\beta)$  and finally he sends  $Y_B$  to Alice. In order to authenticate Bob, Alice will compute:  $Y_A = h(\alpha)$  and then Alice will verify the value of  $Y_A$  by checking that  $(Y_A == Y_B)$ , if so, if they match, then Alice authenticates Bob and Alice can be confirmed that Bob has actually established the same shared session key with her. Finally, Alice and Bob agree on the common session key  $K_s$  where  $K_s = h(ID(Alice) || ID(Bob) || K)$ . Both sides will agree on the session Key  $K_s$  if all steps are executed correctly. Once the protocol run completes successfully, both parties may use  $K_s$  to encrypt subsequent session traffic in order to create a confidential communication channel.

In addition to a complete security analysis presented in our paper [7], we also compared the proposed protocol with the same protocols used in [6] and also with EC-SRP and EC-AKE [19]. The comparison is done in terms of number of rounds, random numbers, exponentiations and hash functions. Table 4.3 shows the compared result for number of rounds and exponentiation. Table 4.4 shows the compared result for random numbers and hash functions numbers.

It is clear from Table 4.3 that the EC-SAKA protocol has the minimal cost in terms of number of rounds and exponentiations compared with other protocols. It can be easily noticed that B-SPEKE, SRP, EC-AKE and AMP require 4 rounds while PAK-RY, PAK-X, SKA, LR-AKE and EC-SAKA require 3 rounds. In addition, the computational load was clearly improved using EC-SAKA protocol because, as noted in table 2, EC-SAKA requires two exponentiations, one for the client and one for the server, while the other protocols, including SKA, LR-AKE, EC-AKE and EC-SRP require at least 4 exponentiations.

From Table 4.4, it can be easily noticed that the EC-SAKA protocol requires 2 random numbers and 5 hash functions while all the other protocols require more. In addition,

Table 4.3: EC-SAKA-Comparison of Performance-1-

Protocol	Rounds	Exponentiations		
		Client	Server	Total
B-SPEKE	4	3	4	7
SRP	4	3	3	6
AMP	4	2	3	5
PAK-RY	3	5	4	9
PAK-X	3	5	4	9
SKA	3	2	3	5
LR-AKE	3	3	2	5
EC-AKE	4	2	2	4
EC-SRP	3	2	2	4
EC-SAKA	3	1	1	2

Table 4.4: EC-SAKA-Comparison of Performance-2-

Protocol	Random Numbers	Hash Function Numbers
SRP	2	6
AMP	2	9
PAK-RY	3	8
PAK-X	3	10
SKA	2	7
LR-AKE	2/4	6
EC-AKE	2	6
EC-SRP	3	5
EC-SAKA	2	5

for the EC-SRP and EC-AKE protocols described in [19], it can be easily noticed that our protocol is better than these two protocols in terms of hash functions numbers. For the EC-SRP protocols described in [19], EC-SRP protocol was proposed for a one way authentication while our proposed protocol, EC-SAKA, provides mutual authentication.

Moreover, we have [8] proposed another new and efficient key agreement authentication protocol. In addition to providing mutual authentication and key confirmation between the client, their proposed protocol applies the EC-EGS to the SKA protocol for enhancing the safety level and protocol simplification in terms of computational and communications load. Their protocol is named ECEGS-SKA

In the following, we will briefly describe the proposed protocol: Alice chooses a random point over the elliptic curve called  $P$  with order  $n$  where  $n$  is a large prime number. In addition, Alice chooses a password  $pw$ , computes  $x = h(pw)$  and calculates  $Q$  where  $Q = x * P$ . Finally, Alice generates strong number  $p$  and  $q$  where  $p = 2 * q + 1$ . Once the following parameters  $(E, Q, P, p, q, pw)$  are generated, Alice transfers the  $(E, Q, P, n)$  to the server in a secure way. Once this step is done, the session key generation procedure will be executed as follow:

Within the first flow, Bob chooses a random challenge  $b$ , where  $1 \leq b \leq n - 1$ , then he calculates the point  $B$  where  $B = b * P + Q$ . Finally he sends  $B$  to Alice. Within the second flow, Alice chooses a random challenge  $a$ , where  $1 \leq a \leq n - 1$ , then computes  $A$  where  $A = a * P = (x_A, y_A)$  and calculates  $\alpha$  where  $\alpha = a(B - Q)$  and  $K = Q + \alpha$ . In addition, Alice calculates  $r = (x_A) \bmod(n)$  and computes  $i = a^{-1}(h(\alpha) + x * r) \bmod(n)$ . Finally  $(A, i)$  becomes the signatures pair and Alice transfers  $A$  and  $i$  to the server. Within the third flow, Bob computes  $\beta = b * A$  Computes  $K = Q + \beta$ , computes  $r = x_A \bmod n$ , computes  $v_1 = i * A$  and calculates  $v_2 = (h(\beta)P) + r * Q$ . Finally, Bob checks if  $(v_1 == v_2)$ , if so, Bob authenticates Alice and Bob can be confirmed that Alice has actually established the same shared session key. Then Bob computes:  $Y_B = h(\beta)$  and finally he sends  $Y_B$  to Alice. In order to authenticate Bob, Alice will compute:  $Y_A = h(\alpha)$  and then Alice will verify the value of  $Y_A$  by checking that  $(Y_A == Y_B)$ , if so, if they match, then Alice authenticates Bob and Alice can be confirmed that Bob has actually established the same shared session key with her. Finally, Alice and Bob agree on the common session key  $K_s$  where  $K_s = h(ID(Alice)||ID(Bob)||K)$ . Both sides will agree on the session Key  $K_s$  if all steps are executed correctly. Once the protocol run completes successfully, both parties may use  $K_s$  to encrypt subsequent session traffic in order to create a confidential communication channel.

In addition to a complete security analysis presented in our paper [8], authors compare the proposed protocol with the following protocols: Leakage-Resilient Authenticated Key Exchange (LR-AKE) protocol, Simple Key Agreement (SKA) protocol, Secure Remote Password (SRP) protocol, EC-SRP, Simple Password Exponential Key Exchange (B-SPEKE) protocol, Password-Authenticated Key Exchange (PAK-X and PAK-RY) protocols and Authentication Memorable Password (AMP) protocol. The comparison is done in

terms of number of rounds, random numbers, exponentiations and hash functions. Table 4.5 shows the compared result for number of rounds and exponentiation. Table 4.6 shows the compared result for random numbers and hash functions numbers.

Table 4.5: ECEGS-SKA-Comparison of Performance-1-

Protocol	Rounds	Exponentiations		
		Client	Server	Total
SRP	4	3	3	6
EC-AKE	4	2	2	4
EC-SRP	3	2	2	4
EC-SAKA	3	1	0	1

It is clear from Table 4.5 that the ECEGS-SKA protocol has the minimal cost in terms of number of rounds, exponentiations compared with these above protocols. It can be easily noticed ECEGS-SKA requires 1 exponentiation, one for the client and nothing for the server. While for the other protocols, including SRP, EC-AKE and EC-SRP, they require at least 4 exponentiations.

Table 4.6: ECEGS-SKA-Comparison of Performance-2-

Protocol	Random Numbers	Hash Function Numbers
SRP	2	6
EC-AKE	2	6
EC-SRP	3	5
EC-SAKA	2	5

From Table 4.6, it can be easily noticed that the ECEGS-SKA protocol requires 2 random numbers and 5 hash functions while all the other protocols require more.

## 4.5 Protocol Application

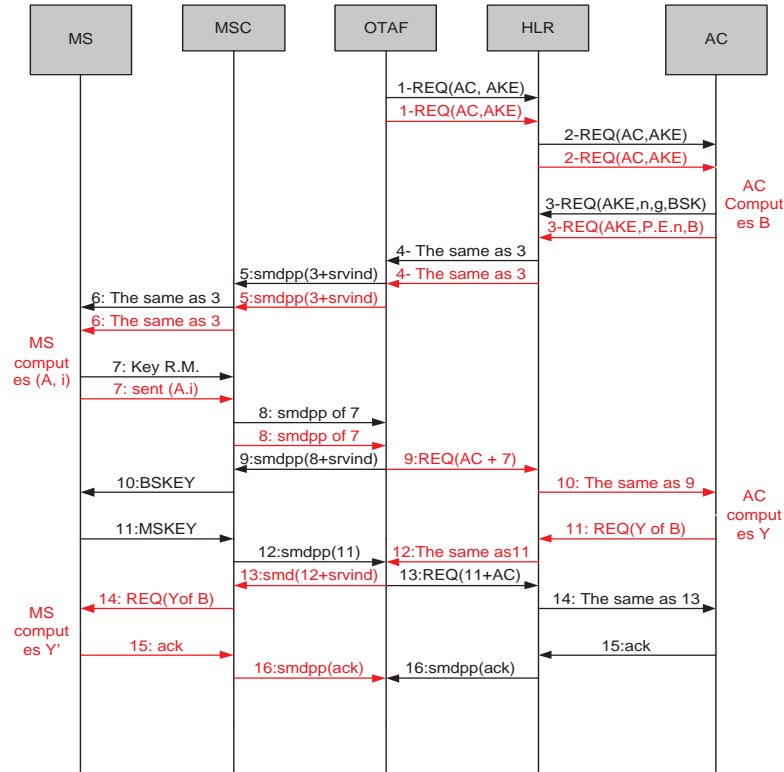
In this section, our proposed protocol [8] is applied to two applications scenarios. In the first scenario, the protocol is applied to improve the A-key distribution in *3GPP2* networks. While in the second scenario, the protocol is applied to wireless LAN, IEEE 802.11i, in order to provide a more robust WLAN communications. The meanings of different symbols used in figure (4.1) are as follows: *AKE* represents a key protocol version parameter. *AC*

represents the Action code.  $BSK$  or  $BSKEY$  represents the encryption key value from the network side.  $BSK = g^x \text{mod} n$ , where  $x$  is randomly selected by  $AC$ .  $srvind$  represents service indicator parameter.  $MSKEY$  represents the encryption key value from mobile subscriber.  $smdpp$  represents the short messages service delivery point-to-point and finally  $REQ$  represents the Over the Air Service Provisioning request.

**1–Application To 3GPP2:** According to [11], there are several proposed approaches for A-Key generation and distribution. The Over the Air Service Provisioning ( $OTASP$ ) is the preferred approach by 3GPP2. The A-Key generation and renewal procedure take place between a Mobile Subscriber ( $MS$ ) and its home network represented by the Authentication Center ( $AC$ ). This procedure takes place through the mobile switching subscriber ( $MSC$ ), the over the air function  $OTAF$ , and finally through the Home Location Register ( $HLR$ ). In A-Key distribution, the basic Diffie-Hellman key exchanged mechanism is used and 16 messages are needed. However, the method is not completely secure since it is subject to a man-in-the-middle attack. Using the same approach as in [11], our proposed protocol can be easily implemented in 3GPP2 networks. We assume that the MS device has the ability to implement the ECC techniques. We also assume that the password is chosen by the user or generated secretly and it is known by the MS and the AC of the home network. Figure (4.1) shows the normal A-Key generation procedure (black arrows) and the A-Key generation procedure using our EC-based protocol [8] (red and dash arrows).

The integration of our proposed protocol within 3GPP2 networks is performed as follows: the message exchange 1 and 2 are the same for the two protocols. After receiving message 2, the authentication center  $AC$  computes  $B$  as described in our proposed protocol [8], packages the message 3 as in Figure (4.1), and finally sends it to  $MS$ . From message 3 through 6, we transmit all needed parameters to  $MS$  that are required to compute  $(A, i)$  (Please refer to message 3 in figure (4.1)). Then messages 7-10 will be used to inform the authentication center about all parameters required to compute  $Y_B$ . Using messages 11-14, the authentication center will transmit  $Y_B$  to the  $MS$  for authentication, verification and key establishment. Compared to 3GPP2 specifications, our new protocol provides key validation, mutual authentication, perfect forward privacy, and it can thwart the man-in-the-middle attack. Moreover, the A-Key protocol require 4 exponentiation operations, while our proposed protocol require 1 exponentiation operation and it could be

Figure 4.1: The A-Key Distribution Procedures

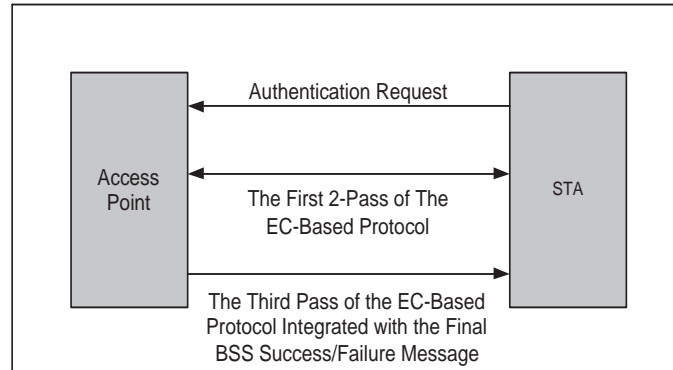


easily up-gradated to just require multiplication and addition operations This upgrade could be achieved by using a suitable digital signature.

**2–Application To WLAN:** Moreover, our EC-based proposed protocol [8], can be easily integrated into the BSS and ESS networks respectively by using the same approach as [27]. In case of BSS networks the entity Bob works as an access point *AP*, whereas in ESS networks it works as a RADIUS server. For both networks, the entity Alice works as a mobile station *STA*. In BSS networks, after the reception of the authentication request sent by the *STA*, the *AP* will start the EC-based protocol [8] and depending on the verification, the *STA* will accept or discard the session. Figure (4.2), shows the message exchange of EC-based protocol in the BSS network. The exchange of messages used by the EC-based protocol for BSS network are done using *WLAN* frame format.

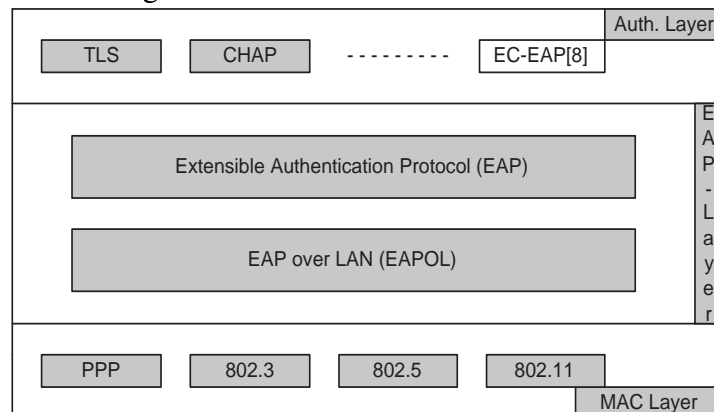
In addition, our EC-based proposed protocol [8], can be easily integrated into the ESS

Figure 4.2: The EC-based in BSS



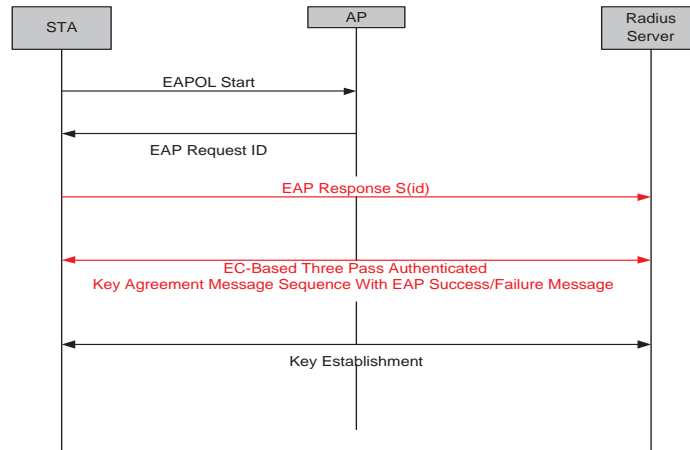
networks. The exchange of messages used by the EC-based protocol within the ESS network are done using *EAP* packet format. Figure (4.3), shows the implementation of the proposed EC-based protocol within the *EAP* stack.

Figure 4.3: The EC-based In EAP Stack



In *ESS* networks, after the association phase and after the *STA*'s response to *AP*'s EAP-request *ID*, *AP* becomes a pass through device and it passes the EAP-request *ID* to the radius server. The three-pass exchange messages in figure (4.4) start by the radius server sending the point *B* to the *STA*. Depending on the authentication process, the success/failure is issued and the *STA* can accept or discard the session. Figure (4.4), shows shortly the corresponding message exchanges of the proposed EC-based protocol in the ESS network.

Figure 4.4: The EC-based In ESS Networks



## 4.6 An ID-Based Pairing Protocol

As the elliptic curve **pairings** techniques have brought many interesting applications to authentication and key agreement protocols [10], we will present an identity-based authenticated key agreement protocols from pairings where an entity is proving its identity to the verifying server in such a way that privacy and anonymity are protected. The presented work is mainly based on [9] and also partially on [7, 8] by applying the EC pairings techniques. A user  $U_i$  represents a mobile client which has a mobile Phone or a PDA as an access device for accessing the needed services. In the following, we will present our proposed work and we discuss the security analysis.

### 4.6.1 Parameters Initialization

Our infrastructure involves a Trusted Key Generation Center ( $TKGC$ ), an embedded device  $ED$ , a Reader (or readers) ( $R$ ), a back end server for authentication ( $BS$ ), a Server for providing services ( $SS$ ) and users denoted by ( $U_i$ ). The trusted Key Generation Center ( $TKGC$ ) chooses two primes order group  $G_1$  and  $G_2$  of prime order  $q$ .  $q$  is a prime which is large enough to make solving discrete logarithm problem in  $G_1$  and  $G_2$  infeasible. The  $TKGC$  chooses  $G$  as a generator of  $G_1$ , chooses Map-To-Point/Curve function  $H$  and chooses  $e$  where  $e$  is the bilinear pairing map. The  $TKGC$  computes  $P_{TKGC} = s.G$ ,

where  $s \in Z_q^*$  is the TKGC 's private key and keep  $s$  secret. Finally, for each user  $U_i$  to be registered, TKGC calculates  $Q_i$ , where  $Q_i$  is user's partial public key with  $Q_i = H(ID_i)$ , and determines  $U_i$ 's partial private key  $S_i = s.Q_i$ . Moreover, the TKGC calculates the user's public key [26] as  $P_U = x_u.P_{Pub} = x_u.s.G$ , where  $x_u \in Z_q^*$  is generated on user's behavior.

The table below (Table 6.1) shows the mathematical parameters that are used for our proposed scheme.

Table 4.7: EC Mathematical Notations

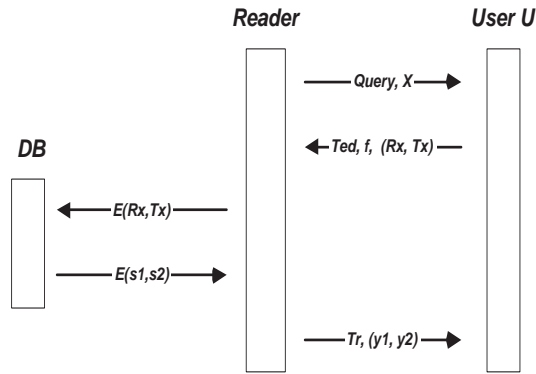
Index	Explanation
$TKGC$	The trusted key generation center
$G_1$	An additive group with prime order $q$
$G_2$	An multiplicative group with prime order $q$
$G$	A generator of $G_1$
$P_{pub}$	The public key of $TKGC$ , $P_{pub} = s.G$
$s$	It is chosen from $Z_q^*$ by $TKGC$ , $s$ is kept secret
$ID_i$	The identity of the user $i$ , $ID_i \in \{0, 1\}^*$
$S_i$	The long term private key of user $i$ , $1 \leq i \leq n$
$Q_i$	The long term public key of user $i$ , $Q_i = s.H(ID_i)$ , where $H$ is a Map function
$H_2$	Hash function
$H$	A map to curve algorithm where an ID is mapped into a point on $G_1$
$e$	$e$ denote a bilinear pairing map
$p, q$	Large prime numbers, where $p = 2.q + 1$
$P_1, P_2$	Random points over elliptic curve
$E$	Non-supersingular elliptic curve
$x(Q)$	$x$ coordinate of point $Q$

### 4.6.2 Proposed Protocol Assumption

We firstly assume that the user's public and private key  $(Q_i, S_i)$  are kept secure, which means that  $S_i$  for each  $U_i$  is stored on his own  $ED$  in a secure way. In additional, we assume that the communication channel between the reader and the back-end server ( $BS$  server or the authentication server) is insecure. In addition, and different from the previous

works, a reader is no more a trusted third party, which means that the reader have to be authenticated by the back-end server (*BS*). Finally, *TKGC* sends  $S_i$ ,  $P_i$ ,  $Z$  to the user via a secure channel. The Back-End Server *BS* manages, creates an stores, for each user  $U_i$  with an *ED*, a record pair consisting of  $\langle Q_i, S_i, s_1, s_2 \rangle$ , where  $(s_1, s_2)$  are unique and are the *BS*'s secret for each specific user such that  $Z = -s_1P_1 - s_2P_2$ .

Figure 4.5: Our Protocol Protocol



### 4.6.3 Proposed Protocol Description

Before running the authentication procedure (Figure 4.5), the reader must be able to address a particular embedded device, to singulate it, from among a population of many others devices. During singularization, multiple embedded devices responses may interfere with each other, necessitating an anti-collision algorithm. The Anti-Collision algorithm may either be probabilistic or deterministic. Following this situation, the reader  $R$  applies a collision-avoidance protocol like the secure binary tree walking [20, 21, 22]. Once the reader singulates one device, the three-pass authentication protocol process will be described in the following steps.

**Within the first round**, (From  $R$  to  $ED$  or  $User$ ), the reader starts the protocol by generating two fresh random nonce  $r_1$  and  $r_2 \in Z_n$ , then he calculates the point  $X$  where

$$X = r_1 \times P_1 + r_2 \times P_2 \quad (4.1)$$

and finally he sends the pair ("Query",  $X$ ) to the embedded device  $ED$ . (step 1 in Figure 4.5)

**Within the second round**, the queried  $ED$  generates two fresh random nonces  $f$  and  $a$ , where  $f \in_R Z_2^t$  and  $a \in Z_q^*$ , then computes  $(R_x, T_x)$  where  $(R_x, T_x)$  is the signature pair over the user's private key  $S_i$ . Moreover, it calculates  $T_{ED}$ , where  $T_{ED} = a.G$ . Finally  $ED$  sends  $(R_x, T_x)$ ,  $T_{ED}$ , and  $f$  to the Reader  $R$ . (Step 2 in Figure 4.5). We can choose to deploy one of many available secure signature algorithm. The choice of the algorithm depend on the Computation and communication cost factor regarding the choice of the  $ED$  type.

**Within the third round**, and as we have declared in the above assumption that the communication channel between the reader and the authentication server is insecure, and upon receiving the signature pair  $(R_x, T_x)$  from the  $ED$ , the reader  $R$  will deploy a Weil Pairing-based encryption algorithm on the signature pair. Finally he sends  $E_{K_e}(R_x, T_x)$  to the Back-end server  $BS$ . (step 3 in Figure 4.5)

Our two nodes, the reader and the back-end server, can directly compute a share key between them without exchanging any previous message. Based on the one's own private key and the other party's public key, they can directly compute the share key as follows. We denote their private key/public key by  $S_R = s.Q_R$ , where  $Q_R = H_1(ID_R)$  and by  $S_{BS} = s.Q_{BS}$ , where  $Q_{BS} = H_1(ID_{BS})$ . Now the reader computes  $K_{R/BS} = e(S_R, Q_{BS})$  and  $K_{BS/R} = e(Q_R, S_{BS})$ . And finally the share symmetric secret key will be

$$K_e = H_2(K_{R/BS}) = H_2[e(Q_R, Q_{BS})^s] = H_2(K_{BS/R}). \quad (4.2)$$

This approach is very efficient in terms of communications and computations and this feature makes it very attractive to the environments where the entities capabilities are limited.

**Within the fourth round**, and upon receiving the encrypted signature pair message  $E_{K_e}(R_x, T_x)$  from the  $R$ , the back-end server,  $BS$ , will decrypt the message, then verify the signature pair, if it is valid, then the back-end server accept, and the pair  $(s_1, s_2)$  associated with the authenticated  $ED$  is extracted from the back end server, encrypted using the Weil-Pairing-based encryption algorithm. Finally, the back-end server sends  $E_{K_e}(s_1, s_2)$  to the reader  $R$ . (step 4 in Figure 4.5)

**Within the fifth round**, the reader, generates a random nonce  $b \in Z_q^*$  and computes  $T_R = b.G$ . Then it decrypts the receiving message, extracts the pair  $(s_1, s_2)$  and then computes

$$y_i = (r_i + (f \times s_i))(\text{mod } n) \quad (4.3)$$

for  $i = 1$  and  $2$ . Finally sends  $(T_R, y_i$  for  $i = 1$  and  $2)$  to the  $ED$ . (step 5 in Figure 4.5)

The  $ED$  computes

$$\left(\sum (y_i \times P_i) + f \times Z\right) (\text{with } i = 1 \text{ and } 2) \quad (4.4)$$

and then checks that if  $(\sum (y_i \times P_i) + f \times Z)$  is equals to  $X$ , if so the  $ED$  accepts else rejects.

After the above messages,  $T_{ED}$  and  $T_R$  are exchanged, the reader and the user can agree and compute the secret shared key

$$K_{R/ED} = e(Q_{ED}, P_{ED})^b . e(x_r . S_r, T_{ED}) \quad (4.5)$$

and

$$K_{ED/R} = e(Q_R, P_R)^a . e(x_{ed} . S_{ed}, T_R) \quad (4.6)$$

respectively. We denote by  $K = K_{R/ED} = K_{ED/R}$ . Hence, the key  $K$  is a shared between the entities. To ensure forward security, we can use a the new shared key  $K_h$  after applying a hash function to  $K$ . Once the protocol run completes successfully, both parties may use the  $K_h$  to encrypt subsequent session traffic in order to create a confidential communication channel. In the following we will present a verification regarding the similarity of the shared key equations:

$$\begin{aligned} K_{R/ED} &= e(Q_{ED}, P_{ED})^b . e(x_r . S_r, T_{ED}) \\ &= e(Q_{ED}, x_{ed} . s . G)^b . e(x_r . S_r, a . G) \\ &= e(x_{ed} . s . Q_{ED}, b . G) . e(x_r . s . Q_R, a . G) \\ &= e(x_{ed} . S_{ed}, T_R) . e(Q_R, P_R)^a \\ &= K_{ED/R} \end{aligned} \quad (4.7)$$

#### 4.6.4 Security Analysis

Our proposed architecture is considered to provide privacy and anonymity for users. In the following, we evaluate our architecture regarding the security requirement addressed in *section4.3*

-Mutual Authentication: Considering the fact that the digital signature pair  $(R_x, T_x)$ , created by the *ED*, is verified by the Back-end server. Considering that the pair  $(s_1, s_2)$ , sent by the back-end server, is recalculated by the reader under  $(y_1, y_2)$  and verified by the *ED*. Therefore, our proposed architecture guarantees the secure mutual authentication between the embedded device *ED* and the back-end server.

-Passive attack: Suppose an attacker performs a passive attack, then the session will terminate with both legitimates parties accepting. That is, the two parties successfully identify themselves to each other. And regarding the fact that the exchanged messages between the reader and the *ED* are generated from random nonce which are generated with every new session, so it is infeasible that an attacker computes any useful information including the  $ID_i$  of a user  $U_i$ . Therefore the architecture resists against the passive attack.

-Man in the middle attack (or active attack): Suppose that an attacker intercepts  $X$  and replaces it with  $X'$ , the attacker then receives  $f$  and  $(R_x, T_x)$  from the *ED*. He would like to replace the pair with  $(R'_x, T'_x)$ , as before. However, and unfortunately for the attacker, he can not compute the value of the new pair because he does not know the users credentials and parameters and because the transmitted messages are meaningless. Therefore the proposed scheme thwarts the man in-the-middle attack.

-Perfect forward secrecy: Each run of the protocol computes a unique  $X$ , a unique Signature pair  $(R_x, T_x)$  and a unique pair  $(y_1, y_2)$ . In addition the transmitted messages are meaningless as they are generated for each new session using new random nonce. Thus, the architecture is secure against perfect forward secrecy.

-Data Confidentiality: Since our architecture provides secure mutual authentication between the *ED* and the system and since the information transmitted between the *ED* and system is meaningless, thus, our architecture provide data confidentiality and the user privacy on data is strongly protected.

-*ED* Anonymity and Location Privacy: During the authentication processes, a signature algorithm is used to produce the signature pair  $(R_x, T_x)$ . The pair  $(R_x, T_x)$  and  $f$  that are transmitted between the *ED* and *R* are randomized and anonymous since they are updated for each read attempt. Thus, our architecture provides user anonymity and location privacy is not compromised.

-Unauthorized Reader Detection: Our proposed architecture is based on the insecure communication channel between *R* and back-end server. The unauthorized reader  $R'$  is detected and prevented by the back-end server *BS* using the weil pairing based encryption algorithm between the reader and the back-end server, and by verifying the pair  $(y_1, y_2)$  by the legitimate user or *ED*. Thus, our scheme protects against Unauthorized reader.

#### 4.6.5 Formal Analysis

In The AVISPA tool [23], security protocols are specified using the High Level Protocol Specification Language (HLPSL). The HLPSL specification is translated into an Intermediate Format (IF). The current version of the AVISPA tool integrates four back-ends: OFMC, CL-ATSE, SATMC and TA4SP.

Figure 4.6: The OFMC Output

```
% OFMC
% Version of 2006/02/13
SUMMARY
SAFE
DETAILS
BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
C:\SPAN\testsuite\results\PAPAAvalidation.if
GOAL
as_specified
BACKEND
OFMC
COMMENTS
STATISTICS
parseTime: 0.00s
searchTime: 0.21s
visitedNodes: 208 nodes
depth: 11 plies
```

Before we run verifications from AVISPA [23, 24], our protocol was written in the High Level Protocol Specification Language, or HLPSL. Once the HLPSL specification was debugged, it was checked automatically for attack detection using the AVISPA verification

tools. Figure 6.4 shows the corresponding execution with AVISPA's OFMC tool where no reveal attack were found.

## 4.7 Combining Authentication and Access Control

Authentication and access control are decisive for the security and integrity of information. In this section, we propose a robust protocol through combining authentication and role-based access control (*RBAC*). To achieve this, we extend our previous protocol to cooperate with role-based access control. Our new scheme is based on identity-based cryptography and bilinear pairings. Our proposed protocol can check the validity of a user's identity and its activated roles simultaneously by verifying the user's signature, so the independent authentication procedure is eliminated. We extend the element *user* in our previous proposed protocol [9] to cooperate with role-based access control. We define each user as  $U_i = \langle ID, AK_{ra} \rangle$ , where *ID* is a user identity information and  $AK_{ra}$  is a set of assigned keys corresponding to the roles assigned to the user defined as  $AK_{ra} = \{K_{IDr_1}, \dots, K_{IDr_n}\}$ . In addition, we define a role as a set of pair of public and private keys belonging to the role. Each role is represented as  $r = \langle r_{pub}, r_{priv} \rangle$ . We also assume that the Trusted Key Generation Center (*TKGC*) in [9] is extended in a way to be able to define roles and to assigning these roles to users. When a role  $r_i$  is added to the system, the *TKGC* picks a random  $rpki$  as  $r_i$ 's private key and sets  $RPK_i = rpki \cdot G$  as  $r_i$ 's public key. To assign the role  $r_i$  to a user with an identity *ID*, the *TKGC* check the user *ID*, computes  $Q_{ID} = H(ID)$ , and generates the user's assigned key  $K_{IDr_i}$  corresponding to  $r_i$  with  $K_{IDr_i} = rpki \cdot Q(ID)$  and where  $rpki$  is the  $r_i$ 's private key. Finally, *TKGC* sends  $K_{IDr_i}$  or a set of  $K_{ID}$ ,  $S_i$ ,  $P_i$ ,  $Z$  to the user via a secure channel.

The process for our new three-round authenticated key agreement protocol will be as follows:

**Within the first round**, (From *R* to *ED*), the reader starts the protocol by generating two fresh random nonce  $r_1$  and  $r_2 \in Z_n$ , then he calculates the point  $X$  where  $X = r_1 \times P_1 + r_2 \times P_2$  and finally he sends the pair ("*request*",  $X$ ) to the embedded device *ED*. (Step 1 in figure 4.5).

**Within the second round** The queried *ED* selects a role or a corresponding set of roles

denoted by  $SR = \{r_1, r_2, \dots, r_h\}$ . Generates a message  $Q$  and a signature  $Sig_Q$  on  $Q$  with  $Q = ID|SR|p_{er}$  and where  $p_{er}$  is the permission that the user wants to enforce. Finally the  $Sig_Q$  will be denoted by  $\langle U, V \rangle$ . Moreover,  $ED$  generates two fresh random nonces  $f$  and  $a$ , where  $f \in_R Z_2^t$  and  $a \in Z_q^*$ , it calculates  $T_{ED}$ , where  $T_{ED} = a.G$ . Finally  $ED$  sends  $(Q, Sig_Q)$ ,  $T_{ED}$ , and  $f$  to the Reader  $R$ . (Step 2 in figure 4.5). We can choose to deploy one of many available secure signature algorithm. The choice of the algorithm depend on the computation and communication cost factor regarding the choice of the  $ED$  type.

**Within the third round**, and as we have declared in the above assumption that the communication channel between the reader and the authentication server is insecure, and upon receiving the signature pair  $(Q, Sig_Q)$  from the  $ED$ , the reader  $R$  will deploy a Weil Pairing-based encryption algorithm on the signature pair. Finally he sends  $E_{K_e}(Sig_Q)$  to the Back-end server  $BS$ . (step 3 in Figure 4.5)

**Within the fourth round**, and upon receiving the encrypted signature pair message  $E_{K_e}(Q, Sig_Q)$  from the  $R$ , the back-end server,  $BS$ , will decrypt the message, then verify the signature pair, if it is valid, then the back-end server accept, and the pair  $(s_1, s_2)$  associated with the authenticated  $ED$  is extracted from the back end server, encrypted using the Weil-Pairing-based encryption algorithm. Finally, the back-end server sends  $E_{K_e}(s_1, s_2)$  to the reader  $R$ . (step 4 in Figure 4.5)

**Within the fifth round**, the reader, generates a random nonce  $b \in Z_q^*$  and computes  $T_R = b.G$ . Then she decrypts the receiving message, extracts the pair  $(s_1, s_2)$  and then computes  $y_i = (r_i + (f \times s_i))(modn)$  for  $i = 1$  and  $2$ . Finally sends  $(T_R, y_i$  for  $i = 1$  and  $2)$  to the  $ED$ . (step 5 in Figure 4.5)

The  $ED$  computes  $(\sum(y_i \times P_i) + f \times Z)$  and then checks that if  $(\sum(y_i \times P_i) + f \times Z)$  is equals to  $X$ , if so the  $ED$  accepts else rejects.

After the above messages,  $T_{ED}$  and  $T_R$  are exchanged, the reader and the user can agree and compute the secret shared key as in equations 6.15 and 6.16.

### 4.7.1 Protocol Discussion

**Protocol Correctness** We can choose one of many identity-based signature scheme to compute the  $Sig_Q$ . Therefore, we will adopt the signature scheme that was used by [25].

To compute the  $Sig_Q$ , the user selects a random  $r \in Z_q^*$ , computes  $U = r.Q_{ID}$ , computes  $h = H(Q, U)$ , computes  $K_{SR} = \sum_{i=1}^h K_{IDr_i}$ , and finally computes  $V = (r + h)K_{SR}$ . The validity of  $Sig_Q$  can be accomplished by verifying if  $e(P, V) =? e(P_{AR}, U + hQ_{ID})$ .

Proof:

$$\begin{aligned}
 e(P_{AR}, U + hQ_{ID}) &= e(\sum_{i=1}^k P_i, rQ_{ID} + hQ_{ID}) \\
 &= e(\sum_{i=1}^k s.P, (r + h)Q_{ID}) \\
 &= e(P, (r + h) \sum_{i=1}^k s_i Q_{ID}) \\
 &= e(P, (r + h)S_{IDAR}) \\
 &= e(P, V)
 \end{aligned} \tag{4.8}$$

Since the key establishment process in our current proposed protocol is similar to the key establishment in our previous ID-based protocol, the same correctness verification will be applied here as well.

**Security Analysis** Since our new proposed authenticated protocol is also a direct extension of the protocol described in [9], the security analysis and validation will be applied to the proposed authenticated key agreement protocol as well.

## 4.8 Conclusion

Mobile environment is an emerging research area with great potential. Authentication and key agreement are two crucial factors to provide security and integrity of data information. Moreover, the privacy and anonymity of users in pervasive environments should be carefully considered. In this chapter, we present several architectures and scenarios to provide authentication with key agreement, to preserve privacy and anonymity. These scenarios are based on elliptic curve techniques, MaptoPoint/Curve algorithm, Weil Pairing and on Identity-based scheme. Moreover, our proposed protocol was extended and modified to support authenticated key agreement mechanism and dynamic keying. In addition, our protocol was also extended to combine authentication and role-based access control using identity based signature. In general, our schemes are simple, easy to realize, and meets security and privacy objectives including, mutual authentication, man-in-the-middle attack, confidentiality, replay attack and user anonymity and location privacy. Our proposed protocols are flexible so they can be configured to use one of many secure communication

scheme desired (signature schemes, identity-based schemes and weil pairing based encryption algorithms). For the communications between the reader and the back-end server, we use the static pair wise key agreement for pair-wise communications. In order to achieve more robust security between the reader and the back-end server, another dynamic key agreement could be used. In addition, the choice of the signature and identity schemes could be done regarding the implementation parameters and environment computing. In coming chapter, we extend and develop these protocols in order to be applied for context-aware environment where context-aware authentication and trust will be presented.

# Bibliography

- [1] D. R. Stinson, *Cryptography Theory and Practice*, In Proceeding of Chapman and Hall/CRC, Third Edition, pages: 353-438, 2006.
- [2] A. Menezes, P.V. Oorschot and S. Vanstone, *Handbook of Applied Cryptography*, in Proceeding of CRC Press, 2nd Edition, 1996.
- [3] S.B. Wilson, D. Johnson and A. Menezes, *Key Agreement Protocols and Their Security Analysis*, In Proceeding of Sixth IMA International Conference on Cryptography and Coding, UK, pp.:30-45, 1997.
- [4] M. Ohkubo, K. Suzuki and S. Kinoshita, *Cyptography Approach to Privacy-Frindly Tags*, In Proceeding of the Privacy WorkShop, MIT, MA, USA, Nov. 2003.
- [5] S. Weis, *Security and Privacy in Radio Frequency Identification Devices*, Master's thesis, MIT, 2003.
- [6] P. Abi-Char, A. Mhamed, B. El Hassan, *A Secure Authenticated Key Agreement Protocol For Wireless Security*, In Proceeding of the Third International Symposium on Information Assurance and Security IAS2007, Manchester, United Kingdom, IEEE Computer Society Press, August, pp. 33-38, 2007.
- [7] P. Abi-Char, A. Mhamed, B. El-Hassan, *A Secure Authenticated Key Agreement Protocol Based on Elliptic Curve Cryptography*, In Proceeding of the Third International Symposium on Information Assurance and Security IAS2007, Manchester, United Kingdom , IEEE Computer Society Press, pp. 89-94, 2007.

- [8] P. Abi-Char, A. Mhamed, B. El-Hassan, *A Fast and Secure Elliptic Curve Based Authenticated Key Agreement Protocol For Low Power Mobile Communications*, In Proceeding of the International Conference and Exhibition On Next Generation Mobile Applications, Services And Technologies, NGMAST07. Cardiff, Wales, United Kingdom , IEEE Computer Society Press, pp. 236-241, 2007.
- [9] P. Abi-Char, A. Mhamed and B. El-Hassan and M. Mokhtari, *Towards a Robust Privacy and Anonymity Preserving Architecture for Ubiquitous Computing*, In Proc. of the Third International Conference on Risks and Security of Internet and Systems (CRISIS08). Tozeur, Tunisia, IEEE Computer Society Press, October 28-30, 2008, pp. 125-132.
- [10] D. Boneh, and M. Franklin, *Identity Based encryption From the Weil Pairing*, In Proceeding of CRYPTO 2001, LNCS 2139, pp. 213-229, SPRINGER-Verlag, 2001.
- [11] A.F. Sui et al., *An Improved Authenticated Key Agreement Protocol with Perfect Forward Secrecy for Wireless Mobile Communication*, In Proceeding of the International Conference of Wireless Communications and Networking, IEEE Press, pp. 2088-2093, 2005.
- [12] I. Hideki, S. Seonghan, and K. Kobara, *Authenticated Key Exchange for Wireless Security*, In Proceeding of the IEEE Wirless Communications and Networking Conference, pp. 1180-1186, 2005.
- [13] E. Ryu, K. Kim, and K. Yoo, *A Simple Key Agreement Protocol*, In Proceeding of IEEE 37th Annual 2003 International Carnahan Conference, pp. 128-131, 2003.
- [14] T. Wu, *Secure Remote Password Protocol*, In Proceeding of the Interent Symposium on Network and Distribution System Security, 1998.
- [15] D. Jablon, *Extended password Key exchange Protocols immune to dictionary attack*, In Proceeding of the WETICE Workshop, pp. 248-255, 1997.

- [16] V. Boyko, P. Mackenzie and S. Patel, *Provably Secure Password Authenticated Key exchange using Diffie-hellman*, In Proceeding of the EuroCrypt, pp. 156-171, 2000.
- [17] P. Mackenzie, *More Efficient Password Authenticated Key Exchange*, In Proceeding of the CT-RSA, pp. 361-377, 2001.
- [18] T. Kwon", *Ultimate solution to authenticate via memorable password*, In Proceeding of the Contribution to the IEEE P 1363 Study group for Future PKC Standards, available for <http://grouper.ieee.org/groups/1363/passwdPK/contribution.html>.
- [19] K. Jung, J. Kim and T. Chung, *Password-Based Independent Authentication and Key Exchange Protocol*, In Proceeding of ICICS-PCM 2003, Singapore, 2003.
- [20] A. Juels, R.L. Rivest, and M. Szydlo, *The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy*, In Proceeding 10th ACM Conference on Computer and Communications, pp 103-111, 2003.
- [21] S. Weis, *Security and Privacy in Radio Frequency Identification Devices*, Master's thesis, MIT, 2003.
- [22] ISO/IEC-JTC 1/SC-31/WG, *Information technology AIDC techniques-RFID for item management air interface*, Part 3: Parameters for air interface communications at 13.56 MHZ, Apr. 2004.
- [23] <http://www.avispa-project.org>, *Automated Validation of Internet Security Protocols and Applications*, 2006.
- [24] <http://www.irisa.fr/lande/genet/span>, *A Security Protocol ANimator for AVISPA*, 2008.
- [25] J. Wang, J. Yu, D. Li, X. Bai, and Z. Jia, *Combining User Authentication With Role-Based Authorization Based on Identity-Based Signature*, In Proceeding of International Conference on Computational Intelligence and Security, CIS, pp.847-857, 2006.
- [26] S. Wang, Z. Cao, and H. Bao, *Efficient Certificateless Authentication and Key Agreement (CL-AK) for Grid Computing*, In Proceeding of the International Journal of Network Security, vol.7, No.3, pp. 342-347, 2008.

- [27] A.A. Mohammad, and A. Jamalipour, *An Efficient Elliptic Curve Cryptography Based Authentication Key Agreement Protocol for Wireless LAN Security*, In Proceeding of High Performance Switching and Routing, HPSR2005 Workshop, pp 376-380, 2005.

## Chapter 5

# SECURITY IN CONTEXT-AWARE ENVIRONMENTS

In this chapter, we present both pervasive computing and contextual information definitions, characterizations, and terminologies. As context awareness represents new challenges and new opportunities regarding privacy, trust and security of users in pervasive computing environments (PCE), the main purpose of this chapter aims to survey each of the involved issues to understand and address the interdependencies among them. In this chapter we also present a survey on relevant related work on authentication and access control within pervasive computing and we also present a survey on relevant work related to trust. This chapter is based on an accepted chapter book edited by Nokia Research Center, 2009 [85], and published under Book Title: *Trust Modeling and management in Digital Environments: From Social Concept to System Development*.

### 5.1 Introduction

Beside Security, Privacy and Trust in pervasive computing are currently hot issues in digital information technology area. As observed by [65], security is used to describe techniques that control who may use or modify private data and context information, privacy is viewed as the ability of an entity to determine whether, when, and to whom information is to be released and finally trust denotes the grounds for confidence that a system will meet

its security objectives. Privacy preservation has been identified as an important factor to the success and acceptance of pervasive computing systems. The development of mobile communications technologies and ubiquitous computing paradigm and the convergence of m-healthcare, m-business, m-entertainment and m-education services have raised the urgency of dealing with privacy threats (i.e. personal information, etc.). These threats are caused by the detection of personal sensitive information such as location, preferences, and activities about individuals through sensors or invisible computing devices, gathering collating data and deriving user context, available anywhere and at any time and for anyone. Organizations and service providers collect large amounts of personal information about individuals in order to deliver suitable services to them; this could lead to a conflict between personal information owners (individuals) and information collectors (e.g. service providers) regarding privacy control. This conflict is mainly caused by the confrontation between service providers, aiming to collect more information about users in order to provide personalized services, and users' requirements of controlling their privacy attributes. In [15], it is mentioned that people dislike automatic transfer of identifiable and personal data, especially when information is spread to other entities beyond their control. Context-aware computing environments may use information provided by many sensors to acquire knowledge about the users' context. These sensors can be invisible to users who consider the act of gathering information about them without being notified as a great threat to their privacy. If the risks of privacy violation when using a context-aware application can not be estimated, users may be unwilling to use such systems. This is why privacy control is essential to be integrated in the design of any new context-aware computing platform. However, the quests for authentication, access control, and user privacy protection conflict with each other in many aspects and the problem is highly complex as the context information of users is more of a concern. On one hand, service providers want to authenticate legitimate users and make sure they are accessing their authorized services in a legal way. On the other hand, users want to maintain the necessary privacy without being tracked down for wherever they are and whatever they are doing. Furthermore, new provided services generally depend on the user identity information, context-awareness information and corresponding pre-established and context-aware dynamically evaluated trust relationship to accomplish user privacy and authentication and to conduct access control. The tradeoff

between privacy and authentication poses great challenges to security designers. This is why the conflict between user privacy protection and user authentication process makes security design in Pervasive Computing Environments (PCE) a very challenging task.

## 5.2 Pervasive Computing

Pervasive computing refers to the emerging trend toward numerous, casually, accessible, often invisible computing devices, frequently mobile or embedded in the environment, and finally connected to an increasingly ubiquitous network infrastructure composed of a wired core and wireless edges [49]. Pervasive computing is expected to enter more and more everyday life in the foreseeable future. It will surround users with a comfortable and convenient information environment that merges physical and computational infrastructures into an integrated environment

### 5.2.1 Properties and Features

The pervasive computing environment will feature a proliferation of hundreds or thousands of computing devices and sensors that will provide new functionalities, offer specialized services, enhance management and control, expand usability and efficiency, and improve interaction. Before addressing the challenges associated with security in pervasive computing environments, we list the salient features of pervasive computing, which were observed by [11]:

*Extend Computing Boundaries:* The pervasive computing should be able to transform traditional computing environment into interactive, dynamic, and programmable environment.

*Invisibility and Non-Intrusiveness:* In pervasive computing, computers should blend in the background allowing people to perform their duties without having machines at the center of their focus. *Creating Smart and Sentient Spaces:* The pervasive computing environment should become intelligent enough to understand users' intent and become an integral part of users' everyday life.

*Context Awareness:* The pervasive computing environment should be able to automatically tailor itself, by capturing and integrating different contexts with users and devices, to meet users' expectations and preferences.

*Mobility and Adaptability:* The pervasive computing environment should be mobile as its users and should be able to adapt itself to evolve and extend once more resources become available.

The vision of ubiquitous computing bears (among others) an obvious problem: privacy (i.e. the capability to determine what one wants to reveal and how accessible one wants to be [3]) is under great risk. Ubiquitous or pervasive Computing essentially relies on intensive collection, processing and dissemination of large amounts of data. Much of this data is related to users (e.g., personal information) and can be very sensitive and of great value for other parties. Langheinrich [41] had identified four key properties of Ubiquitous Computing:

*Ubiquity:* The infrastructure will be everywhere consequently affecting every aspect of life.

*Invisibility:* The infrastructure will be cognitively or physically invisible to the user. The users will have no ideas when or where they are using computer.

*Sensing:* Input to the ever-present invisible computer will be everything we do or say, rather than everything we type.

*Memory Amplification:* Every aspect of these interactions, no matter how personal, has the potential to be stored, queried and replayed.

The descriptions of these key properties show that the pervasive computing environment is characterized by massive numbers of almost invisible miniature sensing devices that can potentially observe, collect and store personal information.

## 5.2.2 Requirements

To deal with the new vulnerabilities introduced by pervasive computing, security and privacy guarantees in pervasive computing environments should be specified and drafted early

into the design process. Previous efforts in retrofitting security, privacy, authentication, access control, and anonymity into existing systems have proved to be inefficient and ineffective. In the following, we will outline some of security requirements for building security and privacy based infrastructures, which have been addressed in [11]:

*Multilevel:* The design for new security architectures should be able to provide different levels of security services based on system policies, preferences, rules, context information, environmental situations, temporal circumstances, available resources, etc.

*Context-Awareness:* Traditional security systems are somewhat static and context insensitive. Pervasive computing integrates context information, transforming the computing environment into a sentient space. Security services should make extensive use of context information available. Security policies must be able to change dynamically regarding the environment changes. In addition, there is a need to verify the authenticity and integrity of the context information acquired.

*Flexibility and customizability:* The security subsystem should be flexible, adaptable, and customizable. It must be able to adapt to environment changes by evolving and providing additional functionality when more or new resources become available.

*Interoperability:* The security architecture should be able to support multiple security mechanisms and levels (e.g. policy discovery, authentication and access control, etc.) and to negotiate security requirements.

*Extend Boundaries:* While traditional information security was restricted to the virtual world, security now should incorporate some aspects of the physical world, e.g. preventing intruders from accessing physical spaces. In essence, virtual and physical security becomes interdependent.

*Scalability:* The security services should be able to scale to the dust of mobile and embedded devices available at some particular instant of time. In addition, the security services need to be able to support huge numbers of users with different roles and privileges, under different situational information.

### 5.2.3 Security Challenges

Recent research on pervasive computing focuses on building infrastructures for managing smart spaces, connecting new devices, and providing useful applications and services. Privacy, trust, and security issues in such environments, however, have not been explored in depth. Indeed, several researchers [40, 41, 58] have admitted that pervasive computing environments are vulnerable to security and privacy threats and that securing pervasive computing [35, 38, 39] presents critical challenges at many levels. Below, we will outline the privacy and security challenges which have been addressed in [11]:

*Privacy Issues:* Increasing active spaces with active sensor and actuators enables the construction of more intelligent capabilities. Unfortunately, these devices could threaten the privacy of users severely because they can be exploited by intruders, malicious insiders, or tracking systems. Thus the privacy aspects have to be considered to protect personal and confidential data of users.

*Users Interaction Issues:* One of the main characteristics of pervasive applications is a richer user interface for interaction between users and the environment. The access control mechanisms have to be integrated in a way allowing users and devices to use the environment in a manner that facilitates collaboration, while enforcing the appropriate access control policies and preventing the unauthorized users.

*Security Policies:* Another characteristic of pervasive computing is to have a suitable and convenient method for defining and managing security policies in a dynamic and flexible fashion. Policy management tools provide administrators the ability to specify, implement, and enforce rules to exercise advanced control over the behavior of entities in their systems.

In addition to the security challenges introduced above, and based on deep relevant researchs [68, 74], we argue the necessity to introduce two more security challenges that should be added to the above list and which are:

*Quality of Privacy:* It is important in pervasive computing to have a flexible and convenient mechanism in order to satisfy the level of Quality of Privacy (*QoP*). This flexible mechanism should allow balancing the trade-off between the amount of privacy a user is willing to concede and the value of the services that can be provided by the

Ubiquitous Computing (*UbiComp*) application.

*Trustworthy authentication*: Trustworthy authentication is defined as entity authentication accompanied by an assurance of trustworthy behavior of the authenticated entity. For example, a wireless printer should provide a trustworthy authentication to a user who wishes to use it. Trustworthy could mean, for instance, that this printer ensures that no other party has access to the data which it is resident on the printer and that the printer itself will not use the data in a malicious way. Moreover, the concept of trustworthy authentication can be generalized for security devices and servers such as portals, firewalls, and intrusion detection systems (IDS).

## 5.3 Context-Aware Computing

Context-Aware computing is an emerging computing paradigm that tries to exploit information about the context of its users to provide new or improved services. Dey et al. [20] have defined context as "*any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves*". This definition is widely used in literature today. Nowadays, the increasing interest in using context awareness information [19, 20, 81] is turning up to become an important factor for future mobile information systems requiring more advanced mobile context based applications and services. Context information can have a strong impact on application adaptation. Models like ambient intelligence and pervasive computing systems rely on context information in order to personalize services provided to their end users. Several terminologies and classifications for context-aware computing have been proposed.

### 5.3.1 Terminology

As observed in [70], context information is defined as any kind of information, which can be used to characterize the state of an entity. An entity might be any kind of assets of a computing system such as user, software, hardware, media storage or data [51]. Moreover, a *Context Information Source* (CIS) is defined as any kind of entity which delivers

context information. A CIS can be a thermometer, GPS, etc. A context-aware system is defined as a backend system that uses any kind of information before or during service provisioning, including, e.g., service design, implementation, and delivery.

### 5.3.2 Life-Cycle of Context-Aware Information

Context information provider delivers context information to a context-aware system following the life-cycle process. As it is outlined in [70], the main steps in a life cycle are:

*Discovery of Context Information Providers:* In this step, a context-aware system discovers available context information providers. The discovery can be performed either in a push or a pull mode [60], i.e. the context-aware system can actively look for CISs or can passively receive information about available CISs.

*Acquisition of Context Information:* In this step, a context-aware system collects context information from the discovered context information providers and stores it in a context information repository for further reasoning. The process of acquisition is performed either in a pull or a push mode. In pull mode, the context-aware system explicitly requests for context information whereas in a push mode, context information providers push context information to the context-aware system.

*Reasoning about context Information* Reasoning mechanisms enable applications to take the advantage of the available context information. The reasoning can be performed based on a single piece of context information or on a collection of such information.

### 5.3.3 Context-Taxonomy

In general, context information can consist of very different information including, e.g., user's location [30], user's identity, activity pattern, IP address, time, role, etc. In [14, 59, 70], the main basic categories for context information can be classified as:

*System Context:* A mobile application has to take into account context information related to both the computing system it is running on, e.g. the particular type of mobile device, and to communications system being used, e.g. the particular type of wireless network. System context deals with any kind of context information related to a computing

system, e.g. computer CPU, network, IP address, status of a workflow, wireless network, etc.

*User Context:* Refers to any kind of context information related to the user and characterizing him. User context information can be user's age, location, medical history, etc. User context can also include context information related to user's tasks, social connections, personal state, and spatial-temporal information.

*Environmental Context:* Consists of any kind of context information related to the physical environment, which is not covered by system and user context. Environment context information include, e.g., lighting, temperature, weather, etc.

*Temporal Context:* Defines any kind of context information related to time. Time and day are typical temporal context information.

### 5.3.4 Reasoning about Uncertain Contexts

In Pervasive computing, context-aware systems can not always identify the current context precisely, so it is crucial that these systems might be integrated with formal decision-making process for handling uncertainty. These systems allow applications and services to reason about uncertainty using new mechanisms such as ontology and fuzzy logic.

The term ontology originates from philosophy and refers to the discipline that deals with existence and the things that exist. In computer science, things that exist are those which can be represented by data. Different definitions for ontology in computer science can be found. As an example, the definition by [23] is given as follow: "*Ontology is a formally defined system of concepts and relations between these concepts. Ontology contains, at least implicitly, rules*".

Ontology in context is needed in order to deal with a heterogeneous character of context information. Ontology focuses on identifying objects by classifying and characterizing them with properties [12]. Context-awareness is important for pervasive computing environments to adapt computational entities to changing situations such as the users' needs and technical capabilities. The fundament for context-awareness is a formal context model which is needed to represent the context in a way computers can interpret it, and a formal context reasoning which is needed to reason on the context knowledge. Context modelling

is the specification of all entities and relations between these entities which are needed to describe the context e.g., information on location, time, current or planned activity, etc. Context reasoning means to automatically deduce further previously implicit facts from explicitly given context information.

A context model is a system of concepts (entities) and relations, which makes ontology a possible mean for context modelling. An ontology is formally defined, which is a precondition for a computer to interpret it, e.g., for reasoning purposes. Rules can be used to implement context reasoning. There are three main areas of application of ontology in context-aware computing [25]:

*Communication and knowledge sharing:* Ontology serves as a common vocabulary of different agents (computational entities and human).

*Logic interfering reasoning:* Ontology can be used to deduce implicit knowledge from explicit knowledge by applying rules.

*Knowledge reuse:* Common ontology (e.g., on time and spatial concepts) can be reused when building domain specific ontology.

Moreover, the real benefit of using ontology for context information in pervasive computing environments will not become effective before there is widely-accepted standard context ontology.

Fuzzy logic theory is used to express fuzzy information, human's experience, human brain concepts, and cognitive process. It has been widely used in decision field. The use of fuzzy logic helps in supporting reasoning under uncertainty. The concept of fuzzy logic was first introduced by [79]. Fuzzy logic employs fuzzy sets to deal with imprecise and incomplete phenomena [4]. A fuzzy set is defined by a so-called membership function. The study of fuzzy sets differs from the study of the probability theory because fuzzy sets depend on subjectivity in perceiving and representing concepts with member functions and not on randomness as in probability theory and statistics [34]. Moreover, trust and risk play a very important role in the field of trusted decision. Risk will evaluate the security of the interaction process between trustor and trustee. Currently, fuzzy logic theory is used to integrate privacy, trust, and risk into trusted decision for providing robust trust models.

However, although people have recognized the importance of security and privacy for their personal information; they remain uncertain when they have to define and enforce

their own access control rules or have to handle indirect information. The trust relationships in pervasive computing environments are hard to assess due to the uncertainties involved. If the trust relationship relies on subjective judgment based on indirect information, it will be very uncertain and any operations related to that relationship may cause unexpected results. The theory of fuzzy logic extends ontology's concepts and theories to be a composite which leverages quality and quantity, and which contains certain fuzziness. With the help of fuzzy operations and rules, we can form a formal decision-making process for handling the imprecise nature and uncertainty in trust management, and for modelling trust representation, trust aggregation, and trust evolution.

Trust management provides trust systems designers the flexibility to manage the enforcement of trust policies. According to [72], this flexibility could be achieved by applying fuzzy logic which can help handling uncertainty and fuzziness in trust management models. Moreover, fuzzy set theory has been used to improve user-role assignment in role-base access control (RBAC). In [7], authors presented an algorithm for reinforcing access control based on heuristic information about the user, the data being accessed, and the various system components. The model uses fuzzy set theory to access the risks involved in granting the requested services based on uncertain information. Takabi et al. [67] applies fuzzy relations into the RBAC model. Their proposed model extends RBAC with fuzzy parameters to allow imprecise access control policies using the concept of trust and trustworthiness. Authors have used fuzzy set theory for measurement and prediction of trustworthiness. Moreover, Rehak et al. [54] showed that trust can be represented by using fuzzy numbers to capture the trust value and its uncertainty. They used the fuzzy rule computation and fuzzy control domain to take trusting decision. In [2], authors expressed trust relationships by using fuzzy logic. They presented a mathematical and probabilistic trust evolution model to decrease the uncertainty for making decisions in pervasive computing environments.

## 5.4 Privacy In Pervasive Computing

This section gives the background information which helps to understand the growing needs of user privacy involved by ubiquitous computing. It presents privacy aspects, privacy laws and technology, and general principles supporting privacy control and management.

Clearly, privacy is a social, ethical and legal issue, beyond technical threats. In order to establish acceptance of Ubiquitous Computing vision, protecting the privacy of users is of central importance. If those privacy concerns are not addressed appropriately, the continuous surveillance through countless sensors may be perceived as a serious downside for those living and working in smart environments like dependant people in a medical center, tourists and visitors in public locations and museums, medical staff moving in their workspace, public users in an airport or public locations, etc.

### 5.4.1 Privacy Definition

Many definitions have been given for privacy. One of these definitions that seem to cover the most of the aspects of privacy was given by *the British Committee on Privacy and related Matters* [9]:

*”The right of the individual to be protected against intrusion into his personal life or affairs, or those of his family, by direct physical means or by publication of information”.*

Privacy is divided into the following aspects which were identified as new technologies or social changes providing new ways of intrusion [41]:

*Behavioral or Media Privacy:* The right to know whom is gathering information about a user.

*Territorial Privacy:* This aspect of privacy relates to the right to have one’s own place, where nobody can enter without permission.

*Communication Privacy:* As direct personal communications through the telephone system became more frequently used, the possibility to tap conversations led to privacy concerns.

*Bodily Privacy:* This form of privacy relates to physical intrusion, and expresses individual's right not to be safeguarded. One example is medical experiments conducted without permission.

*Information Privacy:* With the increase of electronically stored data, ease and speed of access and the possibility of data mining also raised the issue of privacy.

As this last aspect of Privacy is most relevant when dealing with a context-aware system, a definition for privacy has been chosen accordingly to [69]:

*"Privacy is the claim of individuals, groups, and institutions to determine for themselves, when, how and what extent information about them is communicated to others."*

### 5.4.2 General Principles and Privacy Requirements

Based on previous work addressed by Westin [69], Langheinrich [41] has pointed to the Principles of Fair Information Practices:

*Openness and Transparency* No secret record should be kept. Individual Participation: The subject of a record should have the privilege to see and correct the record.

*Collection Limitation:* Record collection should be appropriate for the application. Data quality: Data record should be accurate and relevant to the purposes for which they are collected and also should be kept up to date.

*Use Limitation:* Record should only be used for their well specified purpose and only by relevant authorized people.

*Reasonable Security:* Appropriate security safeguards should be deployed regarding the sensitivity of these records.

*Accountability:* Record keepers must be accountable.

### 5.4.3 Privacy-Aware Design Guidelines

This paragraph tries to serve as an introductory reading to give a comprehensive set of guidelines for designing Privacy-Aware Pervasive Systems. In order to design a general architecture of a privacy awareness system, we should follow six principles set out earlier for preserving privacy in ubiquitous computing [41]. These principles, which are based

on the well-known Fair Information Practices [50], have been adopted as general rules for the development of privacy enhanced pervasive systems (e.g. European Disappearing Computer Privacy Design Guidelines) [42]. In the following, we will outline each of these concepts as observed by [41, 42]:

*Notice:* Given a ubiquitous environment where it is often difficult for data subjects to realize that data collection is actually taking place, we will need not only mechanisms to declare collection practices (i.e., privacy policies), but also efficient ways to communicate these data to the user (i.e., policy announcement).

*Choice and Consent:* In order to give users a true choice, we need to provide a selection mechanism (i.e., privacy agreements) so that users can indicate which services they prefer. These principles state that a user must not only be informed about data collection, but also be offered a choice whether or not to use a data-collecting service.

*Anonymity and Pseudonymity:* Data should not be linked to individuals and one should be able to conceal one's true identity with pseudonyms. Moreover, a service provider should not collect more data than absolutely necessary for performing the service a user requests. Thus, wherever possible anonymous data should be gathered if this kind of data does not pose a threat to privacy.

*Proximity and Locality:* The system should support mechanisms to encode and use locality information for collected data that can enforce access restrictions based on the location of the person wanting to use the data.

*Access and recourse:* Any new system needs to provide a way for users to access their personal information in a simple way through standardized interfaces (i.e., data access). Users should be informed about the usage of their data once it is stored. Moreover, and in order to gain trust, users must be aware of what information is stored about them. Basically, users should be in control of their own data.

*Adequate Security:* To achieve confidentiality in information and communication technology, cryptography is classically regarded as the main mechanism used. However, this is a difficult trade-off for simple, small, and low power devices which will not be able to use robust encryption techniques because of limited systems resources, e.g. computational overhead.

## 5.5 Authentication in Pervasive Computing

Pervasive Computing Environments, (PCE), change constantly by sensing and processing information about users and their environments. This environment is considered intelligent, since it is equipped with sensory means to be aware of changes in the environment. An intelligent environment is also referred to as a smart space in the computer literature. In this environment, user authentication should be an integral part of security of the whole system. In PCE, user authentication may include context authentication in addition to the entity authentication. The concept of context authentication and access control is to collect and recognize the user's current situation and to generate and control a secure user environment based on the current context. In such an environment, protecting the privacy of users is no longer optional, and it must be integrated within authentication service. Additionally, conventional authentication systems usually operate on a fixed set of rules and do not have to track and respond to changes in the environment. Therefore, the conventional authentication schemes are incompetent in satisfying the needs in a context-aware environment that is composed of heterogeneous parts and systems.

### 5.5.1 Authentications Requirements

Most traditional authentication methods either do not scale well in massively distributed environments, with hundreds or thousands of embedded devices like smart spaces, or they are inconvenient for users roaming around within smart environments. In addition, authentication in smart environments can not use a one-size-fits-all approach, as authentication requirements differ greatly among different spaces and different applications and contexts within the same smart space. In general, users must be able to authenticate with other entities with a varied level of confidence, in a transparent, convenient, and private manner. Therefore, any proposed privacy preserving authentication framework must satisfy the following requirements [55, 57]:

*Identity Anonymity:* The identity of the users should be transparent to an Authentication System (AS) whenever an authentication procedure is processed. This can prevent the ASs from mapping a user's identity with its location.

*Mutual Authentication:* During an authentication process, a mutual authentication

process is required and needed. On one hand, a user is required to be authenticated as a legal and legitimate user. On the other hand, the user needs to authenticate the pervasive environment through the AS.

*Context Privacy:* Neither the service nor other users of the services should be able to learn the exact context information (e.g., location, duration, type of service request, etc.) of the user, unless the user decides to disclose such information. Users' context information should be protected against both outsiders and services providers they interact with.

*Confidentiality and Integrity:* During the authentication process, users should be ensured that their transactions cannot be read by unauthorized parties, and the authenticator should be able to detect any intentional or unintentional changes to data that occur in transit.

*Fast Authentication:* The authentication latency must be very short; otherwise, the long authentication delay will disrupt the continuity of the current session or connection.

*Non-Linkability:* The moving route of a particular user should be protected, even if the identities are hidden. The AS should not be able to figure out the relationship between the user and the pervasive environment whenever the authentication mechanism is processed.

### 5.5.2 Designing Privacy-Based Context-Aware Authentication Systems

An inherent tension exists between authentication and privacy because the act of authentication often involves some disclosure or confirmation of personal information. System designers sometimes fail to consider the myriad impact that authentication affects privacy. When designing an authentication system, selecting one for use, or developing policies for one, we should authenticate only for necessary (well-defined purposes), minimize the scope of the data collected, articulate what entities will have access to the collected data, articulate what kinds of access to and use of the data will be allowed, and finally provide means for individuals to check on and correct any information held about them for use in authentication. Context-aware services should be able to trust context data provided to them from these various sources and to respond to changes.

The dynamic nature of a context-aware environment necessitates the need for a very active and flexible authentication mechanism that allows members across different domains

to identify and communicate with each other with a reasonable level of trust. More generally, systems architects' developers should focus more on reconciling authentication and privacy goals when designing, developing, and deploying systems. Understanding security needs and developing appropriate threat models are keys for determining whether and what authentication are necessary and what kind is needed. According to [48], the context-aware authentication service has to hold the following distinguishing properties:

*Context-Awareness:* A context-aware service has to use context data to provide relevant services to users. The security system adapts itself to match with the dynamism of context information. It also has to be able to prune its services accordingly to changes in context data, such as changes in time, location, activity, etc. Therefore, it is critical to check the authenticity and integrity of the context data from context-providers.

*Autonomy:* The context-aware service should involve the last human intervention possible. The security may improvise new policies based on the available or new context data.

*Scalability:* The authentication service has to be capable of bootstrapping trust and authentication across heterogeneous domains.

*Flexibility:* In an open, massively distributed, pervasive computing system, using different means of authentication should be made possible, and it does not have to be constrained to a specific format. Therefore, the system has to be able to provide a great level of customization to each individual.

*Privacy-Preserving:* In a context-aware environment, there will be thousands of sensors recording every type of important information about users. They will silently track user's location, preferences, and activities in the environment. Therefore, protecting privacy of the user is important, and there has to be a provision to protect it against abuse.

## 5.6 Trust In Pervasive Computing

Trust in pervasive computing is a complex subject relating to belief in the honesty, trustfulness, competence, and reliability of entities and agents participating in the network activities. In the context of pervasive computing, trust is usually specified as a set of relations between a resource or service requester and a resource or service provider. These trust

relations are based on previous behaviors of agents and entities. To trust pervasive computing systems, we must be able to manage the privacy, confidentiality, availability, and controlled access to digital information as it flows through the system. Trust forms the basis for allowing a requester to use services or manipulate resources owned by a service provider. Also, it may influence a requester's decision to use a service or resource from a provider. Moreover, the mechanisms required to effectively enforce and deploy trust-based strategies across distributed network are becoming increasingly complex. This complexity arises not only because of the size of distributed users accessing needed services but also because of the fact these trust-based systems should be able to capture security-relevant contextual information, such as time, location, behavioral history, personal characteristics, and capability (e.g. user's intelligent, skills, etc.) at the time access requests are made. These context parameters directly affect the level of trust associated with a user, and hence the decision-making consequence granted to him/her.

### 5.6.1 Trust Management in Pervasive Computing

Trust management was first introduced by [5]. With the application of trust management in research for network security, a more general definition is proposed by Grandison [24]: *"Trust Management is the activity of collecting, encoding, analyzing, and presenting evidence relating to competence, honesty, security or dependability with the purpose of making assessments and decision regarding trust relationships."* Trust management has been introduced in the context of access control [43], public key architecture [10], and peer-to-peer reputation systems [21].

Trust management is a multifunctional control mechanism. We can consider several important key aspects of trust management, including trust establishment, trust negotiation, trust delegation, trust based on reputation, etc. Trust management enables security systems designers to increase the security and privacy of shared resources and collaborative activities without increasing workload. In order to manage a collection of trusted-related activities, flexibility is needed in the enforcement of trust policies. This flexibility could be achieved by applying fuzzy logic to trust management systems in order to reasoning about uncertain contexts in pervasive network environment.

### 5.6.2 Trust Establishment in Pervasive Computing

Due to the mobility of pervasive computing environment, trust management and modelling are identified nowadays as one of the important issues. Moreover, pervasive computing applications may need to interact with entities that are not known a priori and therefore can not be trusted. One of the critical trust management processes is trust establishment [75]. Trust establishment is mainly achieved in the following way [22]: the system collects the trust evidence of the clients, defines the trust policies, and builds up the trust level of the clients based on the trust evidence and policies. As more evidence becomes available, the system iteratively updates the trust information including trust evidence and policies. Because there are different application scenarios and trust policies specified by the systems, different kinds of trust evidence are needed. So are different trust strategies for the trust establishment. For trust establishment in the pervasive computing environments, the mobility and uncertainty of the systems and clients need more dynamic and flexible trust strategies. In addition to the traditional trust strategies such as access control and PKI, other trust strategies are proposed and used for trust establishment and management in pervasive computing environments [75]. These trust strategies are:

*Trust Negotiation:* Is needed when system does not have the client information and there is no third party to consult with on the trustworthiness of the client. In this case, it is only reasonable and practical for the client and system to build their trust relationship by disclosing their credentials gradually to meet the access control policies of each other.

*Trust Delegation:* Is needed when one entity in the system trusts the client and can assign its rights to the clients.

*Trust Based on Reputation:* Is used when the system can derive the clients' trustworthiness from the client's behavior records. Because the system may need to collect the clients' reputation from other peer systems, the trust level of the network and the peers systems are taken into account when deciding the trust reputation of the clients.

*Trust Based on Context and Ontology:* Can be used when clients and the systems may have smart sensing devices. This ontology information can help the system to determine the trust levels of its clients or assign them trust rights in the given context.

*Securing Dataflow and Information Privacy:* Is used to ensure that the sensitive

information is not disclosed and the privacy of the clients is kept.

### 5.6.3 Privacy in Trust Negotiations

Trust negotiation is a compromising approach for establishing trust in open systems, where sensitive interactions may often occur between entities with no prior knowledge of each other. However, although several efficient and powerful negotiations systems have been developed so far, few of them provide a comprehensive solution to protect privacy during the negotiation process. In particular, few of them support Privacy Preferences (P3P) policies, where P3P is the platform for privacy preferences designed to help users express their requirements and preferences in a standard way [16]. During trust negotiations credentials play a key role, in that they represent the means to prove parties properties required to establish trust. Moreover, one of the major concerns users have in adopting negotiation systems is that trust negotiation does not control or safeguard personal information once it has been disclosed. Another potential vulnerability of trust negotiation arises because of the common strategy of postponing actual credential disclosure. Indeed, during the policy evaluation phase, privacy can be compromised in several ways, since there are no guarantees about counterpart honesty until the end of the process. Policy disclosure can be used to determine the value of sensitive attributes without the credential ever being disclosed. Furthermore, during policy exchange it is not to determine whether a party is a legitimate party or not until the credentials are actually disclosed. According to [62], trust negotiations systems allow different subjects to securely exchange protected resources and services. This process is achieved by first establishing trust through a bilateral, iterative process of requesting and disclosing user attributes and policies. Attributes are exchanged through the disclosure of digital credentials. Digital credentials can collect several attributes which can be used to verify identification information. The second key issue of any trust negotiation system is represented by disclosure policies, protecting sensitive resources, credentials, and even other policies from unauthorized accesses. However, trust negotiation systems may represent a threat to privacy in that credentials, exchanged during negotiations, often contain sensitive personal information that may need to be selectively released. In addition, users may need to minimize the released information, thus enforcing the need to know

principle in disclosing relevant credentials to other parties.

## 5.7 Related Work

Recently, many papers have been published to address mechanisms designed against security, privacy threats, and trust in pervasive computing environments. However, most of these designs fall in the scope of establishing a general security framework identifying general security and privacy requirements. Some of these efforts focused on designing security infrastructures to protect users' personal information such as Mix-Network architecture, Mist system, Aware Home Architecture, Solar, etc. Others focused on designing identity management approach. Some efforts focused on providing privacy control through integrating privacy preferences (P3P), policies and context-aware systems. Various trust management strategies including, trust negotiations and trust establishments, have been proposed to prevent unauthorized disclosure of any relevant information that can be used for inferring sensitive credentials. This section discusses these diverse studies, approaches and architectures.

### 5.7.1 Security Infrastructure

*Mist* [1], (is an infrastructure that preserves privacy in ubiquitous environments), that facilitates the separation of location from identity. This allows authorized entities to access services while at the same time preventing the disclosure of their location privacy. *Mist*'s operation is based on *Mist* routers and *Mist* circuits. As drawback, *Mist* has an architectural limitation as it has a limited application area and therefore can not address all privacy issues. In addition, *Mist* is not a context-aware system and do not deal with user preferences and policies. In all context-aware systems, context information is only handled if the user gives its consent to the system and the user is able to define whether his personal information may flow to third-parties or not by using privacy preferences. The *Aware Home* System is integrated into a house with a rich computation and communication infrastructure [13]. The system allows users to control and manage resources in the house from a variety of location. The system takes a privacy approach towards access control by using

Role Based Access Control (RBAC). The Aware Home System provides means to define who can access what types of information or services at specific times. As drawback, the system does not deal with other services than the one inside the house and thus have limited application scope. In addition, the system has no means to protect the kind of information that is given to application. Information needed to active services is more privacy-sensitive than the others that are protected by the system itself. Moreover, the system does not give consent to users. *Solar* is a middleware that supports the collection, processing and dissemination of context information for context-aware applications [47]. To preserve privacy in *Solar*, an Access Control list (ACL) is combined with different roles. The system combines access restriction with policy preference. As drawback, the third parties and services providers are contacted directly without looking at the privacy preference of the user which could cause a security threat regarding user's privacy.

### 5.7.2 Privacy Related Researches

The Privacy Preferences (P3P) helps web sites announce their privacy practices while letting users automate their acceptance or rejection decision [16]. P3P specifies an architecture comprising user agents, privacy reference files and privacy policies. Although P3P provides a technical mechanism for helping inform Web site visitors about privacy policies before they release personal information, it does not provide a mechanism for ensuring that sites act according to their policies. However, P3P was designed for static environment such as Internet where users' privacy preferences are not expected to change. Several existing projects and architectures have extended P3P research with providing security services in context-aware environments. Langheinrich [40] proposed a privacy-awareness system for ubiquitous environments (pawS) that uses P3P for privacy policy description. Langheinrich expresses the need to extend P3P with the capability to describe contextual information. APPEL (A P3P Preference Exchange Language) is proposed as the language for expressing user preferences. PawS aims to allow data collector to announce and implement adequate privacy policies, as well as to provide users with the capability to be aware of how their personal data are processed. As drawback, the system does not propose solutions that support anonymity and pseudonymity. Therefore, although pawS does not offer

a high level of anonymity by itself, its effectiveness, regarding the level of anonymity, is proportional to the level offered by the underlying infrastructure solutions adopted by it. Jiang et al. [33] proposed a privacy control system based on defining information spaces. Zuidweg in [80] has developed a privacy control architecture based on P3P. The architecture aims at providing privacy control for a context-aware application platform developed in the Web Architectures for Services Platforms (WASP) project. Another context-aware system using P3P was proposed by Myles in [46]. The system focuses on requests for location information of users initiated by services providers. The system uses a modified version of P3P in order to not specify the gathered data in the privacy policy.

Hong et al. [28] proposed Confab architecture for privacy-sensitive UbiComp. They assume that a user is in control of his context data by devising an infrastructure that captures, stores, and processes personal information on the user's devices. In case a user decides to disseminate personal data, e.g., his location determined by his GPS system, to a third party, he specifies his privacy preferences and attached them as metadata. Moreover, Confab implements a social component of privacy protection, i.e. users are able to provide white lies (Requested data unknown), to hide their real privacy preferences. Hong and Landay called this ability plausible deniability. As a severe drawback, the Confab architecture does not address the cases, in which context is acquired by external sensors. This underlying assumption does not hold for the vision of smart UbiComp environments.

Hull et al. [27] proposed another privacy preferences mechanism based control system. It is based mainly on users self-provisioning of preferences and rules. In this approach, users are assumed to be heavily involved which represents a challenge when considered for a context aware mobile environment due to the time strictness and complexity of manually managing preferences. Kapadia et al. [36] described the concept of virtual walls, i.e. usable policy abstractions. Like a physical wall controls physical access, a virtual wall controls access to acquired sensor data. Users are enabled to setup their privacy preferences using three predefined levels of configuration, namely transparent, translucent and opaque. As a severe drawback the translucent level, which allows some private data to be accessed from outside, preferably chosen in most cases, does certainly need adjustment to personal demands.

### 5.7.3 Privacy-Enhanced Identity Management Systems

Jendricke et al. [32] introduced an identity management system for PCE where users are issued multiples identities, and the user uses them depending on applications. The paper presents a general framework of using multiple identities to protect users when performing access control and authentication, but did not give any concrete solutions. He et al. [31] proposed a simple anonymous ID mechanism for pervasive computing. As drawbacks, the scheme cannot prevent double spending problem and the scheme does not provide differentiated services access control. In addition, the scheme does not achieve Non-Linkability feature. Moreover, [37, 56, 57] proposed novel schemes which can satisfy the requirements for PCE. These schemes provide differentiated services access control, mutual authentication, and Non-Linkability. As a drawback, these schemes do not provide service discovery mechanism which is nowadays an essential element to access network services.

### 5.7.4 Trust Researches

Herzberg et al. [29] has developed a system for establishing trust between entities based on a trust Policy Language with XML syntax to map these entities to predefined business roles. English et al. [22] described the dynamic aspects for the dynamic trusts models in pervasive computing and provided ideas for trust management processes including trust formation, trust evolution and trust exploitation. Kagal et al. [35, 39] proposed a distributed trust model based on trust delegation in which access rights can be assigned dynamically through delegations. His model also uses ontology to help specify the access rights for users. Yu [77] defined the concepts for trust negotiations, strategies and protocols, and proposed a couple of strategies for automated trust negotiation between two unknown entities. Winsborough in [44, 73] provided trust negotiation models that focus on trust negotiation concepts, strategies and their mathematical interpretation. In trust negotiation, establishment, and management systems, Xiu et al. [75] proposed a formal dynamic trust model for providing a comprehensive solution to solve the trust establishment problems in PCE. Their trust model is a distributed model and works by incorporating different trust strategies in one system. Each trust strategy is implemented by a trust application. When a client requests to access a resource, the trust model determines the trust application modules based

on trust policies for the resource.

Seigneur et al. [63] argued an inherent conflict between trust and privacy because both depend on knowledge about an entity. The more knowledge a first entity knows about a second entity, the more accurate should be the trustworthiness assessment; the more knowledge is known about this second entity, the less privacy is left to this entity. This conflict needs to be addressed because both trust and privacy are essential elements for a smart environment. They proposed a solution to achieve the right trade-off between trust and privacy by ensuring minimal trade of privacy for the required trust. They proposed a model for privacy/trust trade based on linkability of pieces of evidence. They proposed to use pseudonymity as a level of indirection, which allows the formation of trust without exposing the real-world identity. They introduced the *liseng* algorithm to ensure that the minimal linkability principle is taken into account.

A formal framework for trust negotiations has been proposed by [73]. The authors provided an approach for safe enforcement of policies that focus on credentials exchange. A formal notion of safety in automated trust negotiations is given which is based on the possibility by third parties of inferring information on negotiation parties' profiles. However, the framework does not support the development of credentials and policy language based on ontology. Bertino et al. [6] presented a system for trust negotiation specifically designed for preserving privacy during a negotiation. The system provides a support for P3P policies, which can be exchanged at various steps of the negotiation, and for different credentials formats, providing different degrees of privacy protection. The authors have extended the recent work done by [6] and [66] by adding techniques for preserving privacy, such as the selective disclosure of credentials and the integration with P3P platform. However, the authors did not introduce the notion of trust requirements or the notion of reference ontologies. Trust-X is a comprehensive framework for trust negotiations, providing both a language for encoding policies and certificates, and system architecture [6]. In [17, 18] authors extend the existing access control architecture (RBAC) to incorporate trust-based evaluation and reasoning in order to have a more dynamic form of policy that can reason with uncertainty. Both of these approaches are risk-aware. Huynh et al. [26] introduced a trust model, called FIRE, which has four components: interaction trust, role-based trust, witness reputation, and certified reputation. FIRE incorporates all those components to

provide a combined trust framework. However, FIRE is not a risk-aware.

Ray et al. [53] proposed anonymization techniques, generalization and substitution techniques, where a subject can transform its disclosure set into an anonymous one. They proposed that trust negotiation requirements can be expressed using property-based policies where a property-based policy can be implemented by a number of disclosure policies. The property-based policy lists the properties the counterpart has to provide and the conditions it must satisfy in order to obtain some resource. A disclosure policy lists the attributes and credentials types needed to obtain a given resources. In addition, as anonymity may present an important requirement for trust negotiating subjects, the authors included the concept of identity disclosure. An identity disclosure is said to occur for the subject who submits the credentials if the data released to the counterpart contain attributes and credentials that uniquely identify her. Identity disclosure happens when either the identity of an individual is directly revealed or it can be derived from the released data. Trust management and negotiation are a key aspect of secure knowledge management [8]. Secure knowledge-management technologies include technologies for secure data management and information management including databases, information systems, and data mining. Therefore, only authorized individuals must be permitted to execute various operations and functions. Moreover, the work presented by [64] is developed in the system context of [6]. The authors addressed the problem of preserving privacy in trust negotiations by proposing three orthogonal privacy preserving mechanisms that can be used in trust negotiations. They have addressed the notion of privacy preserving disclosure by introducing substitution and generalization techniques. In Bertino et al. [64], authors discussed trust management and negotiation by establishing different trust negotiation rules for collaboration between different parties. Squicciarini et al. [64] proposed a protocol that supports anonymization in trust negotiations. They mentioned that credentials, exchanged during trust negotiations, often contain sensitive attributes that attest to the properties of the credential owner. Uncontrolled disclosure of such sensitive attributes may cause grave damage to the credential owner. The proposed protocol gives assurance to the credentials submitter that his disclosure set is  $k$ -anonymous. Moreover, their protocol ensures that the credentials submitted by a subject cannot be linked to the ones previously submitted by him. In [78, 82] authors

propose a context-based method for trust model to find reliable recommendations and filter out unfair recommendations in PCE. Context is exploited to analyze users' behavior, state and intention. Moreover, authors use learning based neural network to cope with the context to catch doubtful recommendations.

Ries in [52] has developed a new trust model that can easily be interpreted and adjusted by users and software agents. One key feature is that it is capable of expressing the certainty of a trust opinion by using contexts which are associated with different levels of risk in interactions. In [71], trust and recommendations are formally defined and analyzed by incorporating belief, disbelief and uncertainty to each interaction. Xu et al. [76] proposed a novel trust framework for pervasive computing. They presented a hybrid model including a trust model, a security model, and a risk model. The proposed framework is dynamic and lightweight enough to be applicable in PCE. However, the proposed protocol does not address users' role and context factors. Squicciarini et al. [61] investigated privacy in the context of trust negotiations. They proposed a framework for negotiating the release of sensitive attributes. Authors proposed a set of privacy-preserving features such as the support for the P3P. They discuss several interoperable negotiation strategies for improving privacy and efficiency. Uddin et al. [84] proposed a context-based trust model for open and dynamic systems called CAT. Authors presented an interaction-based context-aware trust model by considering services as contexts. The proposed protocol uses rule-based trust calculation, direct trust calculation, and direct and indirect recommendation calculation. However, in CAT, it is considered that the network is secure from malicious attacks, and therefore CAT presents a major drawback. Moreover, CAT can not perform authentication process. Mohan et al. [45] proposed a framework for evaluating trust. They presented attribute trust, a policy-based enhanced framework, for aggregating user attributes and evaluating confidence in these attributes. Authors addressed the problem by integrating a reputation system model based on transitive trust.

In the table below (Table 5.1), we compare some of the most important features for the above schemes. The comparison is done based on privacy and security related features. The following comparison cover these features: Trust Management (TM), Context-Awareness (CA), Mutual Authentication (MA), User Context Privacy (UCP), Non-Linkability (NL), Data Confidentiality and Integrity (DCI), Differentiated Service Access Control (DSAC),

Level of Anonymity (LA), Quality of Privacy (QoP), and Risk Awareness (RA).

From this table, we can deduce that much research still needs to be done concerning privacy, trust, and security. To overcome these limitations, a deep study is required and a cohesive model should be created to reflect user's real world and its perception on privacy, trust, and risk in different situations and environments.

Table 5.1: Protocol Security Features Comparison

	MA	UCP	NL	LA	DCI	DSAC	QoP	CA	TM	RA
<i>Mist</i> [1]	Partially	N.A.	Yes	High	Yes	No	No	No	No	No
Aware H. [13]	Yes	Yes	N.A.	N.A.	Yes	No	No	No	No	No
Solar [47]	N.A.	No	N.A.	N.A.	N.A.	N.A.	No	No	No	No
PawS [40]	Partially	Yes	N.A.	High	Yes	No	No	Yes	No	No
Jend02 [32]	No	No	No	Medium	No	Yes	No	No	No	No
He04 [31]	Yes	Yes	No	Medium	No	No	No	Yes	No	No
Ren05 [56]	Yes	Yes	Partially	High	Yes	Yes	No	Yes	No	No
Ren06 [55]	Yes	Yes	Partially	High	No	Yes	No	Yes	No	No
Kim07 [37]	Yes	Yes	Yes.	High	Yes	Yes	No	Yes	No	No
Ren07 [57]	Yes	Yes	Yes	High	Yes	Yes	No	Yes	No	No
FIRE 04 [26]	No	Yes	N.A.	N.A.	N.A.	No	No	Yes	Yes	No
Dim04 [17]	No	N.A.	N.A.	N.A.	Yes	No	No	Yes	Yes	Yes
Dim05 [18]	No	No	N.A.	N.A.	N.A.	No	No	No	Yes	Yes
Yuan06 [78]	No	Yes	N.A.	N.A.	N.A.	N.A.	No	Yes	Yes	No
Yuan06 [82]	No	Yes	N.A.	N.A.	N.A.	N.A.	No	Yes	Yes	No
Ries07 [52]	No	No	N.A.	N.A.	N.A.	N.A.	No	Yes	Yes	Yes
Xu07 [76]	No	No	N.A.	N.A.	N.A.	N.A.	No	No	Yes	Yes
Uddin 08 [84]	No	Yes	N.A.	N.A.	N.A.	N.A.	No	Yes	Yes	No
Mohan 08 [45]	No	Yes	N.A.	N.A.	N.A.	N.A.	No	Yes	Yes	No

## 5.8 Conclusion

In this chapter, we have outlined challenges facing developers of UbiComp applications with regard toward privacy, security and trust. The main contribution for this chapter is the

notion of security, privacy, and trust enhancing services. Privacy and trust adaptively provide protection for users as they enter UbiComp environments which allow for continued transparent use of services without compromising neither the users privacy nor the services ability to provide services. In pervasive computing environments, the deployment for a robust and dynamic security framework should be conditioned by privacy and trust needs so users can be confident by using available services within the environment.

New researches should focus on understanding these services in order to design a global architecture which preserve privacy, provide flexible authentication, enhance context-aware access control, and ensure the enforcement of dynamic authorization. Moreover, authentication and context-aware should be integrated within any trust and privacy-based access control models. Access Control should require user authentication as a prerequisite, should be strong and efficient, and should be integrated within the authentication process. Trust needs to be integrated into the design of any new context-based framework. In addition, trust model based on users' roles, capabilities, behavior, and context factors, etc., should be further investigated and improved.

# Bibliography

- [1] J. Al-Muhtadi, R. Campell, A. Kapadia, M. Mickunas, and S. Yi *Routing Through the Mist: Privacy Preserving Communication In Ubiquitous Computing Environments.*, In Proceedings of the International Conference of Distributed Computing Systems (ICDCS 2002), pp. 65-74, Vienna, Austria.
- [2] F., Almenarez, A., Marin, D., Diaz, and J., Sanchez *Developing a Model for Trust Management in Pervasive Devices* , In Proceeding of the fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM'2006), Italy. pp: 267-271.
- [3] V., Bellotti *Design for Privacy in Multimedia Computing and Communications Environments.* , In Proceedings of the Technology and Privacy: The New Landscape, 1997, pages 63-98
- [4] G., Bojadziev, G., and M., Bojadziev *Fuzzy Logic for Business, Finance, and Management. World Scientific.*, In Proceedings of the Fifth International Conference on Information Technology: New World Scientific.
- [5] M., Blaze, J., Feigenbaum, and J., Lacy *Decentralized Trust Management.*, In Proceeding. of the 1996 IEEE Symposium on Security and Privacy, 1996, pp. 164-173.
- [6] E., Bertino, E., Ferrari, and A., Squicciarini *Privacy Preserving Trust Negotiations.*, In Proceeding of the fourth International workshop Privacy Enhancing Technologies, 2004, pp. 283-301.

- [7] A., Berrached and A., Korvin *Reinforcing Access Control Using Fuzzy Relation Equations*, In Proceeding of the 2006 International Conference on Security and Management (SAM06), 2006, pp: 489-493
- [8] E., Bertino, L., Khan, R., Sandhu, and B., Thuraisingham *Secure Knowledge Management: Confidentiality, Trust, and Privacy*, In IEEE Transactions on Systems, Man, and Cybernetics, Systems and Humans, Vol. 36, No. 3, 2006, pp. 429-438.
- [9] D., Calcutt *Report of the Committee on Privacy and Related Matters*, In Proceedings of (1990). British House of Commons, Committee on Privacy and Related Matters, 1990, Cm 1102, London
- [10] G., Caronni *Walking the Web of Trust.*, In Proceeding of 9th IEEE International Workshops on Enabling Technologies, pp. 153-158.
- [11] R., Campbell, J., Al-Muhtadi, P., Nadldurg, G., Sampemane, and M. Mickunas *Towards Security and Privacy for Pervasive Computing.*, In Proceedings: ISSS, Tokyo, Japan, 2002, pp 1-15
- [12] B., Chandrasekaran, J., Josephson, and V., Benjamins *What are ontologies, and why do we need them?*, In ACM press, IEEE Intelligent Systems and Applications, Vol (14), 1999, pp. 20-26.
- [13] M., Covington, W., Long, S., Srinivasan, A., Dey, M., Ahamad and D.G., Abowd *Securing Context-Aware Applications Using Environments Roles*, In Proceeding of the sixth ACM Symposium on Access Controls Models and Technologies, 2001, pp. 10-20
- [14] G., Chen, and D., Kotz *A Survey of Context-Aware Mobile Computing Research.*, . Tech. Rep. Dartmouth Computer Science Technical Report TR2000-381, 2000.
- [15] L., Cranor, J., Reagle, and S.M., Ackerman *Beyond Concern: Understanding net USERS' attitudes about online privacy.*, In Proceedings of Technical report TR 99.4.3, AT and T Labs Research, April 1999.

- [16] L., Cranor, and R., Wenning *Platform for Privacy Preferences (P3P) Project.*, Technical Report, Retrieved March 4, 2009 from, <http://www.w3.org/P3P/>
- [17] N., Dimmock, A., Belokosztolski, D., Eyers, J., Bacon, D., Ingram, and K., Moody *Using Trust and Risk in Role-Based Access Control Policies*, In Proceedings of the 9th ACM Symposium on Access Control Models and technologies. USA, ACM Press, 2004, pp. 156-162
- [18] N., Dimmock, J., Bacon, D., Ingram, and Moody *Risk Models for Trust-Based Access Control (TBAC)*, In Proceedings of the 3rd Annual Conference on Trust Management, France, 2005, pp. 364-371.
- [19] A.K., Dey *Understanding and using context. Personal and Ubiquitous Computing.*, Journal, volume 5 (1), 2001, pp. 4-7. Retrieved February 23, 2009, <http://www.cc.gatech.edu/fce/ctk/pubs/PeTe5-1.pdf>
- [20] A.K., Dey, and G.D., Abowd *Towards a better understanding of context and context-awareness.*, The CHI 2000 Workshop on the What, Who, when, and How of Context-Awareness. The Hague, Netherlands, Apr. 2000. Retrieved January 8, 2009, from <ftp://ftp.cc.gatech.edu/pub/gvu/tr/1999/92-22.pdf>
- [21] C., Duma, N., Shahmehri, and G., Caronni, G *Dynamic Trust Metrics for Peer-to-Peer Systems.*, In Proceedings of 2nd IEEE Workshop on P2P Data Management, Security and Trust, 2005.
- [22] C., English, P., Nixon, S., Terzis, A., McGettrick, and H., Lowe *Dynamic Trust Models for Ubiquitous Computing Environments*, In Proceedings of UBICOMP2002-Workshop on Security in Ubiquitous Computing, Goteborg, Sweden, 2002.
- [23] A. Feruzan *Context Modeling and Reasoning using Ontologies*, Technical Report, July 2007, Retrieved February 15, 2009, from, <http://www.ponnuki.de/cmaruo/cmaruo.pdf>
- [24] T., Grandison *Trust Management for Internet Application.*, PhD thesis, Imperial College London, 2003.

- [25] M., Gruninger, and J., Lee *Ontology-Applications and Design.*, In Proceedings Communications of the ACM, 2002, Vol(45), No. 2, pp. 39-65.
- [26] T., Huynh, N., Jennings, and N., Shadbolt, *FIRE: An integrated trust and reputation model for open multi-agent systems.*, In Proceeding of the 16th European Conference on Artificial Intelligence, Spain, 2004, pp. 18-22.
- [27] R., Hull, B., Kumar, D., Lieuwen, P.F., Patel-Schneider, A., Shuguet, S., Varadarajan, and A., Vyas *Enabling Context-Aware and Privacy-Conscious User Data Sharing* , In Proceeding of 2004 IEEE International Conference on Mobile Data Management, 2004, pp.187-198.
- [28] I.J., Hong, and A.L., Landay *An Architecture for Privacy-Sensitive Ubiquitous Computing.*, In Proceeding of the Second International Conference on Mobile Systems, Applications and Services (MobiSys 2004), pp. 177-189.
- [29] A., Herzberg, Y., Mass, and J., Michaeli *Access Control Meets Public Key Infrastructure, or: Assigning Roles to Strangers*, In Proceeding 2000 IEEE Symposium on Security and Privacy, May, pp. 2-14.
- [30] U., Hengartner, and P., Steenkiste *Implementing access control to people location information.*, In SACMAT '04, Proceedings of the 9th ACM symposium on Access Control Models and Technologies. ACM Press, 2004, pp.11-20
- [31] Q., He, L., Wu, and P., Khosla *for Personal Control over Mobile Location Privacy* , In IEEE Commun. Mag., vol. 42, no. 5, 2004, pp. 130-136.
- [32] U., Jendricke, M., Kreutzer, and A., Zugenmair *Pervasive Privacy with Identity Management* , In Proceeding of. 1st Workshop Security, UbiComp 2002.
- [33] X., Jiang, and J.A., Landay *Privacy Control in Context-Aware Systems*, In . IEEE Computer Press, Pervasive Computing, 2002, 1(3): p. 59-63.
- [34] A., Konar *Artificial Intelligence and Soft Computing*, In Proceeding of Behavioral and Cognitive Modeling of the Human Brain. New York, CRC Press, 2000.

- [35] L., Kagal, T., Finin, and A., Joshi *Trusted-based Security in Pervasive Computing Environments*, In IEEE Computer Society Press, Vol(34), Issue 12, 2001, pp 154-157.
- [36] A., Kapadia, T., Henderson, I.J., Fielding, and D., Kotz *Virtual Walls: Protecting Digital Privacy in Pervasive Environments.*, In Proceedings of the Fifth International Conference on Pervasive Computing (Pervasive), pp. 162-179, Springer-Verlag, May 2007
- [37] J., Kim, Z., Kim, and K., Kim *A lightweight Privacy Preserving Authentication and Access Control Scheme for Ubiquitous Environment*, In Proceeding of the 10th international Conference on Information Security and Cryptography, ICISC 2007, pp. 37-48.
- [38] L., Kagal, J., Undercoffer, F., Perich, A., Joshi, and T., Finin *Vigil: Enforcing Security in Ubiquitous Environments*, Technical Report: Grace Hopper Celebration of Women in Computing 2002.
- [39] L., Kagal, J., Undercoffer, F., Perich, A., Joshi, and T., Finin *A Security Architecture Based on Trust Management for Pervasive Computing*, In Proceeding of Grace Hopper Celebration of Women in Computing.
- [40] M., Langheinrich *A Privacy Awareness System for Ubiquitous Computing Environments*, In Proceeding of the 4th International Conference on Ubiquitous Computing (UbiComp2002), 2002, Springer-Verlag LNCS, vol. 2498, pp.237-245.
- [41] M., Langheinrich *Privacy by Design-Principles of Privacy-Aware Ubiquitous Systems*, In Proceeding of the 3rd International Conference on Ubiquitous Computing (UbiComp 2001), Springer-Velag LNCS 2201. pp. 273-291
- [42] S., Lahlou, and F., Jegou *European Disappearing Computer Privacy Design Guidelines*, Ambient Agoras Programme Report (IST-2000-25134). Disappearing Computer Initiative, 2004.
- [43] N., Li, and J.C., Mitchell *Datalog with Constraints: A Foundation for Trust-Management Languages*, In Proceeding of the Fifth International Symposium on Practical aspects of Declarative Languages, 2003, pp. 58-73.

- [44] N., Li, W., Winsborough, and J., Mitchell *Distributed Credential Chain Discovery in Trust Management*, In Journal of Computer Security, 2003, vol. 11, no. 1, pp. 35-86.
- [45] A., Mohan, and M., Blough *Attribute Trust-a Framework for Evaluating Trust in Aggregated Attributes via a Reputation System*, In proceeding of the 6th Annual Conference on Privacy, Security and Trust. IEEE Computer Society Press, 2008, pp. 201-212.
- [46] G., Myles, A., Friday, and N., Davies *Preserving Privacy in Environments with Location-Based Applications*, In IEEE Pervasive Computing, IEEE Computer, Press, 2004, pp. 56-64.
- [47] K., Minami, K., and D., Kotz *Controlling Access to Pervasive Information in the "Solar" system*, Dartmouth Computer Science Technical Report TR2002-422, February 28, 2002.
- [48] B., Malek, A., Miri, and A., Karmouch *A Framework for Context-Aware Authentication*, In 2008 IET 4th International Conference on Intelligent Environments. IEEE Computer Society Press, 2008, pp. 1-8.
- [49] NIST *National Institute of Standards and Technologies, About Pervasive Computing*, Technical Report, 2001, Retrieved December 13, 2008, from <http://www.nist.gov/pc2001/about-pervasive.html>.
- [50] OECD *Organization for Economic Co-operation and Development*, Recommendation of the council concerning guidelines on the Protection of Privacy and Transborder Flows of Personal Data, 1980. Retrieved December 14, 2008, from <http://www.oecd.org/document>.
- [51] P.C., Pfleeger *Security in Computing*, In Prentice-Hall, Inc., 1997
- [52] S., Ries *Certain Trust: A Trust Model for Users and Agents*, In Proceeding of the 22nd Annual ACM Symposium on Applied Computing, ACM Press, 2007: 1599-1604.

- [53] I., Ray, E., Bertino, A., Squicciarini, and E., Ferrari *Anonymity Preserving Techniques in Trust Negotiations* , In Proceeding of the 5th International Workshop on Privacy Enhancing Technologies (PET), Cavtat, Croatia, 2005, pp. 93-109
- [54] M., Rehak, M., Foltyn, M., Pechoucek, and P., Benda *Trust Model for Open Ubiquitous agent systems*, , Proceeding of the IEEE/WIC/ACM/ International Conference on Intelligent Agent Technology (IAT), Compiegne University of Technology, France, September 2005. pp: 536-542.
- [55] K., Ren, and W., Lou *Privacy-Enhanced, Attack-Resilient Access Control in Pervasive Computing Environments with Optional Context Authentication Capability* , In Springer Science LLC 2006, Mobile Netw Appl (2007)12:79-92.
- [56] K., Ren, and W., Lou *Privacy Enhanced Access Control in Ubiquitous Computing Environments*, In Proceeding of the 2nd International Conference of Broadband Networks 2005, Vol. 1, pp. 356-365.
- [57] K., Ren, W., Lou, K., Kim, and R., Deng *A Novel Privacy Preserving Authentication and Access Control Scheme for Pervasive Computing Environments* , In IEEE Transactions on Vehicular Technology, 2007, Vol 55, no. 4, pp. 1373-1384.
- [58] F., Stajano *Security for Ubiquitous Computing* , Halsted Press, 2002.
- [59] B., Schilit, N., Adams, and R., Want *Context-Aware Computer Applications* , In Proceeding of the The Workshop on Mobile Computing Systems and Applications, 1994, pp. 85-90.
- [60] B.I.J., Siljee, and I.E. Bosloper *A classification framework for storage and retrieved of context* , In Proceeding of the First International Workshop on Modeling and Retrieval of Context (MRC2004), 2004
- [61] A., Squicciarini, E., Bertino, E., Ferrari, F., Paci, F., and B., Thuraisinghm *PP-Trust-X: A system for Privacy Preserving Trust Negotiations* , In Proceeding of the ACM Transactions on Information and System Security (TISSEC), 2007, Vol 10, Issue 3

- [62] A., Squicciarini, E., Bertino, E., Ferrari, and I., Ray *Achieving Privacy in Trust Negotiations with an Ontology-Based Approach*, In the proceeding of Dependable and Secure Computing, IEEE Transactions, 2006, Vol 3, no. 1, pp. 13-30.
- [63] J., Seigneur, and C., Jensen *Trading Privacy for Trust*, In Proceeding of the 2nd International Conference on Trust Management, 2004, pp 93-107.
- [64] A., Squicciarini, E., Bertino, E., Ferrari, and I., Ray *Trust Negotiations with Customizable Anonymity*, In Proceeding of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-ATW06), 2006, pp. 69-72.
- [65] J., Saltzer, and M., Schroeder *The protection of information in computer systems*, In Proceeding of the IEEE Computer Society Press, vol. 63, number 9, September 1975.
- [66] K.E., Seamons, M., Winslett, and T., Yu *Limiting the Disclosure of Access Control Policies During Automated Trust Negotiation*, In Proceeding of the Network and Distributed System Security Symposium, San Diego, CA, February 2001.
- [67] H., Takabi, M., Amini, and R., Jalili *Enhancing Role-Based Access Control Model through Fuzzy Relations*, In Proceeding of the third International Conference on Information Assurance and Security (IAS07), IEEE, 2007, Computer Society Press, pp: 131-136
- [68] M., Tentori, J., Favela, V., Gonzalez, and M., Rodriguez *Supporting Quality of Privacy (QOP) in Pervasive Computing*, In Proceeding of the Sixth Mexican International Conference on Computer Science. ACM, 2005, Press, pp. 58-67.
- [69] F.A., Westin *Privacy and Freedom*, Technical Report, New York, 1967.
- [70] K., Worna, and L., Gomez *Context-Aware Security and Secure Context-Awareness in Ubiquitous Computing Environments*, In SAP research, XXI Autumn Meeting of Polish Information Processing Society Conference Proceedings, 2005, pp. 255-265
- [71] Y., Wang, and M., Singh *Formal Trust Model for Multiagent Systems*, In proceeding of the 20th International Joint Conference on Artificial Intelligence, Boston, 2007, pp. 1-6.

- [72] Z., Wu, and A., Weaver *Application of Fuzzy Logic in Federated Trust Management for Pervasive Computing*, In Proceeding of the 30th Annual International Computer Software and Applications Conferences (COMPSAC06). IEEE Computer Society Press, 2006, pp: 215-222
- [73] W., Winsborough, and N., Li *Towards Practical Automated Trust Negotiation*, In Proceeding of the 3rd International Workshop on Policies for Distributed Systems and Networks. California. IEEE Computer Society Press 2002, pp. 92-103.
- [74] H., Xia, J., Malcolm, B., Christianson, and Y., Zhang *Hierarchical Trustworthy Authentication for Pervasive Computing*, In Proceeding 4th Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services. IEEE Society Press, 2007, pp. 1-3.
- [75] D., Xiu, and Z., Liu *A Dynamic Trust Model for Pervasive Computing Environments. A Research Paper*, Research Supported by the NSF0406325, 2004. [coitweb.uncc.edu/~zhliu/Research/Papers/asc.pdf](http://coitweb.uncc.edu/~zhliu/Research/Papers/asc.pdf)
- [76] W., Xu, T., Xin, and G., Lu *A Trust Framework for Pervasive Computing Environments*, In the proceeding of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCom 07), IEEE Society Press, pp. 2222-2225.
- [77] T., Yu *Automated Trust Establishment In Open Systems*, Ph.D. Dissertation. University of Illinois at Urbana-Champaign, 2003.
- [78] W., Yuan, D., Guan, S., Lee, Y., Lee, and A., Gavrilov *Finding Reliable Recommendations for trust model*, In proceeding of the 7th International Conference on Web Information Systems Engineering (WISE06), pp. 375-386.
- [79] L.A., Zadhe *Outline of a new Approach to the Analysis of Complex Systems and Decision Processes*, In ." IEEE Trans. on Systems, Man and Cybernetics, 1973. 3(1), pp: 28-44.

- [80] M., Zuidweg *Using P3P in a Web Services-Based context-aware Application Platform*, In Proceeding of 9th Open European Summer School and IFIP Workshop on Next Generation Networks (EUNICE 2003), Hungary.
- [81] T., Zimmer *Toward a better understanding of Context attributes*, In Proceeding of The second IEEE Annual Conference on Pervasive Computing and Communications Workshops (PERCOMMW04) IEEE Computer Society, 2004. (pp. 23).
- [82] W., Yuan, D., Guan, S., Lee, Y., Lee, and H., Lee *Filtering out unfair recommendations Finding for trust model in ubiquitous environments.*, In proceeding of the second International Conference on Information Systems Security (ICISS 06), pp. 258-263
- [83] E., Bertino, E., Ferrari, and A., Squicciarini *Trust-X: A Peer to Peer Framework for Trust Establishment.*, In Proceeding of the IEEE Transaction Knowledge and Data Engineer., vol 16, no. 7, pp. 827-842, April 2004.
- [84] M. Uddin, M. Zulkernine, and S. Ahamed *CAT: a context-aware trust model for open and dynamic systems*, In Proceedings of the 2008 ACM symposium on Applied computing, pp. 2024-2029, 2008.
- [85] P. ABI-CHAR, A. Mhamed, B. EL-Hassan and M. Mokhtari, *Controlling Trust and Privacy in Context-Aware Environments, State of Art and Future Directions*, In Proceeding of the IGI Publishing under Book Title: Trust Modeling and management in Digital Environments: From Social Concept to System Development. Book Edited by Nokia Research Center, Finland, 2009. Chapter Accepted, (To Appear).

## Chapter 6

# A TRUST-BASED SECURE FRAMEWORK

In this chapter we extend our previous work [8] by exploring the security parameters in context-aware computing with a focus on authentication. We proposed a novel authentication framework that utilizes contextual attributes to achieve privacy and authenticity both for user and contextual information. Moreover, we incorporate a trust-based engine that affects the level of trust associated with a user in order to enhance the authentication access request. In addition, we motivate the design of an access control scheme that addresses the context-aware issue for access decisions. We present the configuration mechanisms needed to achieve the proposed framework and then we analyze the security of the protocol. This chapter is based on an accepted chapter book edited by Nokia Research Center, 2009 [7] and on an accepted paper at CRISIS2008 international conference [8]. The architecture's implementation and evaluation are performed using the Handicom Lab's platform at Telecom SudParis, and are to be presented in the coming chapter.

### 6.1 Introduction

The growing evolution of Information and Communication Technology (ICT) systems towards more pervasive and ubiquitous infrastructures contributes significantly to the deployment of services anywhere, at anytime and for anyone. To provide personalized services

in such infrastructures, we should consider both user's privacy including security requirements and authentication mechanism, and context-awareness environment. This can be really achieved owing to context awareness systems which allow us to benefit from sensing and mobile technologies to derive more accurate data about the user such as location, time, etc. While the availability of contextual information may introduce new threats against security and privacy, it can also be used to improve dynamic, adaptive and autonomic aspects of security, and user privacy. These threats exist due to the detection of personal sensitive information such as location, preferences and activities about individuals through sensors available anywhere and at any time. Moreover, the authentication process is a fundamental building block in any system in which entities have to identify themselves in order to access resources and services.

In the previous chapters we looked at ways of providing authentication based security in various scenarios, with focus on which cryptographic techniques to use. However, in this chapter we present our work with different areas, with different security requirements. Our approach enables users to be authenticated using a combination of contextual information, harvested from the environment, and some new cryptography-based techniques.

The remainder of the chapter proceeds as follows. In Section 6.2 we present a review of literature and define the problem statement. Moreover, we present an outline of the previous closely related work. In Section 6.3, we give an outline of our proposed solution and propose needed assumptions that have been made as we developed our model. In section 6.4 and 6.5, we propose our trust-based context-aware authentication model. In section 6.6, we provide a summary of the framework interactions that take place when a user want to invoke a context-aware service. In section 6.7, the security analysis is described, it shows how security and privacy requirements are fulfilled. Finally, the chapter is concluded in section 6.8.

## 6.2 Review of Literature

User authentication, authorization and access control are basic requirements for various services in mobile computing such as Auction, e-Learning, GPS, accessing wireless LAN,

e-Government, etc. However we cannot adapt the traditional mechanisms since they do not consider unique characteristics of pervasive computing (i.e., privacy, context-aware based services, etc.). User privacy is one of the big challenges due to the limited communication range of ubiquitous computing devices, and because there are many invisible computing devices that can collect and analyze the identities, locations and personal information of users without their prior agreement or recognition. There are many approaches to solve privacy and security challenges in Ubiquitous or pervasive environments. However, most of these results fall in the scope of establishing a generic security framework that adapting and identifying all security requirements. Characteristics and limitations of these protocols with recently related researches were discussed in (*chapter 5*).

### 6.2.1 Statement of the Project Problem

One key challenge in pervasive applications is managing security including privacy, authentication, and access control. In traditional approaches, permission to access resources or services is moderated by checking for authentication and access control processes associated with each object. However, this strategy is inadequate for pervasive applications as it does not consider context information. In a pervasive environment, users are mobile and typically access resources (information, services, sensors, etc.) using mobile devices. As a result the context of a user (i.e. location, time, system resources, network state, network security configuration, etc.) is highly dynamic, and granting access to user(s) without taking the his current context into account can compromise security as the users access privileges not only depend on "*who the user is*" but also on "*where the user is*", "*what is the users state*", and "*what are the states of the users environment*".

Traditional authentication and access control mechanisms break down in such an environment and a fine-grained authentication and access control mechanisms that change the privilege of a user dynamically based on context information is required. Although a lot of work has been done in the area of authentication and access control, most of this work is user-centric, where credentials of the user are only considered when granting access permission. Characteristics and limitations of these protocols were discussed in previous chapters (*chapters 3 and 4*). Therefore, the existing researches do not address pervasive

application where context is dynamic and users's privileges must continuously adapt based on the runtime context.

### 6.2.2 Closely Related Work

In this section, we briefly highlight existing researches that has influenced our work with attribute-based authentication, security, and trust.

Authors, in [11], have defined a model that uses contextual attributes to achieve an approach to authentication that is better suited for dynamic, mobile computing environments. They examined the use of trusted platforms to provide assurances for these contextual attributes. Although authors claimed that their model provides a seamless and flexible user experience that can protect privacy and reduce administrative overhead, it does not provides trust and reasoning and there no mention about how to protect privacy (i.e, user, attributes, and data privacy). Marc Langheinrich, [6], introduce a privacy awareness system that allows data collectors to both announce and implement data usage policies. The announced data collections of each services and their policies is delegated by a mobile privacy assistant to a personal privacy proxy residing on the platform, which interact with corresponding service proxies and inquires their privacy policies (Privacy Beacon). Corner et al. [12] describe *Transient Authentication* as a means of authenticating users with devices through a small, short-ranged wireless communications token. This research is limited to the use of location-based context (*i.e., proximity*) as an attribute in authentication. A similar approach is taken by Glynos et al. [14] where they combined traditional authentication with a limited set of contextual information used to identify users. Another similar approaches were taken by [9, 24] where they also have performed an authentication process that is based on a limited set of attributes. Moreover, their architecture do not provide privacy control neither trust management. However, we have presented a more generic approach that allows any attributes to be used for authentication. Creese et al. [13] present a general overview of security requirements for authentication in pervasive computing and discuss how traditional authentication does not fit these requirements. Although they discuss authentication of entities using attributes, they did not present a framework for authentication as we have done. In [10], authors have developed a P3P-based privacy control architecture

for the WASP platform by providing users with means to control their personal data. However, the architecture does not support user and context-aware based authentication neither user trustworthiness evaluation. In [15], authors present a service provision mechanism which can enable effective service provision based on semantic similarity measure with the combination of user profiles and situation context in mobile enabled environment. The paper suggests the combination of user profiles and contextual information to provide a more pervasive service experience in smart assistive environments with mobile device. Behzad et al. [16] propose a framework to construct a context-aware authentication system. The framework is flexible, privacy preserving, and provide context-aware user authentication. However it does not support user trustworthiness evaluation neither user role assignment. Moreover, the framework is designed to be applicable to Ad-Hoc network does not provide users a way to control attributes. In [17], authors propose an authentication scheme for a mobile ubiquitous environment, in which the trustworthiness of a users device is authenticated anonymously to a remote Service Provider (verifier), during the service discovery process. However, the scheme do not provide support for contextual information, and does not support fuzzy private matching. Ren et al. [26] propose a framework to construct a privacy-enhanced authentication system. The framework is flexible, and privacy preserving. However, it does not provide full context-aware user authentication and neither support user trustworthiness evaluation.

### 6.2.3 Work Valuable

Transactions in a mobile environment involve the user, the mobile platform, the specific resource or service being accessed, and the physical environment of both the user and platform. Contextual attributes can describe the user's mobile operating environment, without necessarily disclosing the user's personally identifiable information. In addition, contextual attributes allow for flexible policies and rules that can be applied to a variety of access requests. Chen et al. [25] define four categories of contextual attributes for pervasive environment including: *Computing Context*: network connectivity, costs and nearby resources such as printers, etc. *User Context*: user's profile, location, people nearby, etc. *Physical Context*: lighting, noise level, traffic condition, temperature, etc.

*Time Context*: time of a day, week, month, etc. Traditional approaches to authentication are not well-suited for rich mobile applications because they fail to utilize the contextual information presented by Chen [25]. In the following, we will propose our new approach to achieve authentication using contextual attributes. Our proposed approach is an extension of our previous work [8] by presenting an authentication model using trusted platforms capabilities and contextual attributes. A significant body of the work has emerged around the use of contextual information in computer systems. Context can be used to provide these systems with certain capabilities inherent to human perception and reasoning. Context describes a specific situation by capturing the setting in which an event occurs. These observations have led us to consider the impact of context on security services. In particular, we are interested in how contextual attributes can be used to support and enhance authentication, security, and access control in dynamic mobile environments by presenting a secure framework that provides all these security requirements.

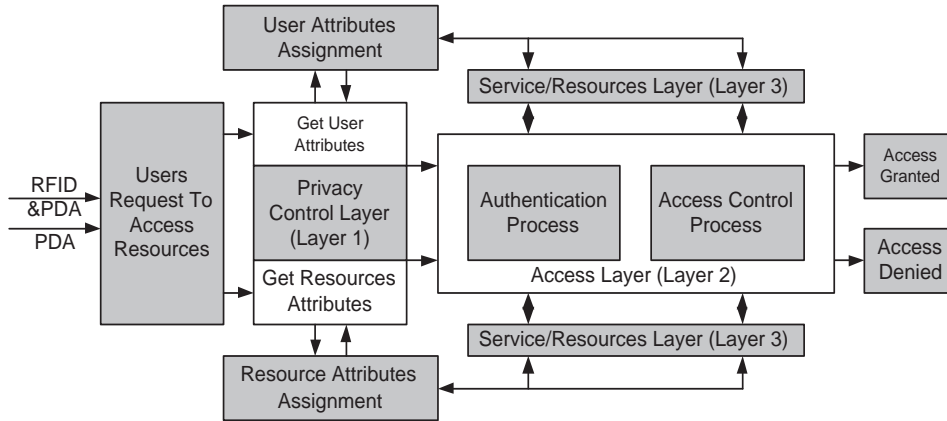
### 6.3 Towards a New Solution

Here, we outline our proposed authentication-based privacy enhancing infrastructure. Our framework is based on a privacy control layer, a context-aware authentication process, a context-aware access control process and the use of attributes-based private set intersection and trust evaluation engines.

Our framework is a layered architecture that discriminates service providers (context consumers), privacy control process (Layer 1), authentication and access control process (Layer 2), and finally service process (Layer 3). The figure below (Figure 6.1) shows the process of granting access to resources with the help of user and attributes. Attributes can contain identity and other contextual information (i.e user's profile).

In our framework, we design an integration scenario where mobile subjects (*i.e users*) carrying embedded devices (i.e., smart phones, PDA, etc.) receive pervasive services according to their identity and real-time context information environments. The cornerstone of our framework is the flexibility to provide authentication and access control for independent and dependent (with special needs) people both at context level and where privacy is preserved. Moreover, our framework provides a distributed infrastructure that allows

Figure 6.1: Context-Aware Framework



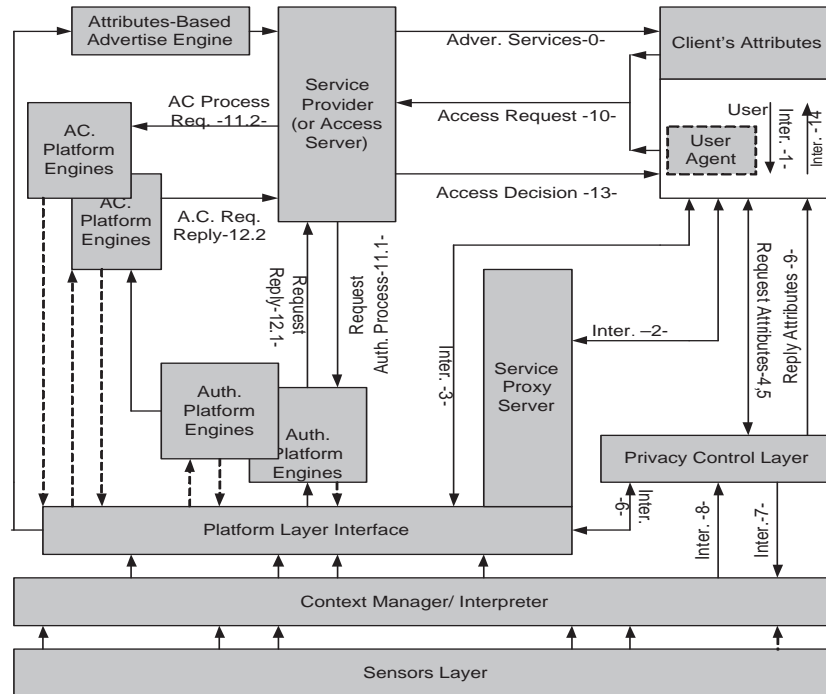
the tracking of the context in a real-time manner. In the following sections, we detail the functionality of these components and describe how they interact with one another. A high-level overview of these logical component and how they interact is given in following figure (Figure 6.2).

Our model is based on contextual information obtained from a distributed network of sensors and it is composed of the following logical components: *Users*, *Sensors*, *Embedded Devices*, *Adaptors*, *Contextual-Attribute Infrastructure* including *Trust Engine*, *Access Control Process*, *Privacy Control Process* and *Authentication Process*. In the following we will detail the functionality of these components.

### 6.3.1 The Entities

Our authentication framework is based on contextual information from the situation in which a service advertisement is made. User authentication is a prerequisite of access control. However, there are few works which actually aims at integrating embedded devices and sensors data into the design of authentication schemes and also few works have been done on designing adequate schemes for people with special needs. Our authentication scheme can be easily integrated to work with different embedded devices such RFID tags or smart card together with PDAs or smart phones to allow users to login. Our authentication scheme uses the following:

Figure 6.2: A High-Level View



-*Badges*: For scenarios where users do not have a PDA or smart phone, every user is given a badge that contains an RFID tag or a smart card.

-*RFID Readers*: Every access point terminal is equipped with an RFID reader. This point make our framework very flexible for scenarios where users are equipped with a RFID tag in order to be authenticated.

-*Smart Card Readers*: Every access point terminal is equipped with an smart card reader. This point make our framework very flexible for scenarios where users are equipped with a smart card in order to be authenticated.

-*Body Network Sensor*: Users with Special needs will have a Body network sensor (*BNS*) woven into their coat, not visible, which reveal data to the context-awareness system. When a user is presented, the *BNS* adapter will translate, for example, a 64-bit tag ID into telling the context server that a user with a specific need tag ID has been detected. Moreover, this 64-bit tag ID will be saved on the user's embedded device in such a way to be used later on for authenticating the user.

-*Client*: In our model, the client ( $C$ ) refers to an entity attempting to access a protected service or resource. This entity would be comprised of both the user and the user agent. For the purpose of our model, the user agent includes the needed software components that allow him to perform a specific function; these components serve an important role in the privacy chain that exists between the user and service being accessed. The client agent is responsible for sending a request for collecting relevant contextual attributes. Contextual attributes can be obtained by the user agent either directly from the environment or from an attribute provider.

-*Attribute Provider*: The attribute provider ( $AP$ ) is the entity that makes contextual information available to the users agents and to the platform.

-*Context TrustWorthy*: The context trustworthy engine ( $CT$ ) is the entity that verify the correctness of the attributes. We make the use of it in order to make our framework more flexible for users holding RFID, and smart card.

-*Private Set Intersection Engine*: A  $PSI$  engine enable two entities, each holding a set of inputs, to jointly identify the intersection of their inputs sets (i.e, shared context), without leaking any additional information about other credentials that each entity might have. Nevertheless, both entities, the prover and the verifier, need to protect their credentials from each other. Moreover,  $PSI$  provide a contextual information authentication based on context-aware confidence level.

-*Trust Engine*: A Trust engine in pervasive computing enable to manage privacy, confidentiality, availability, and controlled access to digital information as it flows through the systems. These approaches could be solved using the concept of fuzzy-based trustworthiness.

-*Context Type*: A context type is defined as a property related to every participant in the system. In simple cases, context type may be a concrete property familiar in everyday life, such as time, location, user ID, etc. In a more complex scenario, context type can also be used to describe users with special needs (i.e. -dependent people with BNS). Based on context type, we define context set as a set of context types by:  $CS = \{CT_1, CT_2, CT_3, \dots, CT_n\}$ , where this context set will be used later for users' authentication mechanism and for defining users' privileges through a context-aware access control mechanism.

### 6.3.2 Assumptions

We now describe the assumptions that were made as we were developing our model for context-based authentication. These assumptions and definitions pertain to the authentication model, system requirements, and infrastructure needed components.

-One limitation in achieving attributes-based authentication is the lack of security assurance for acquiring and reporting contextual attributes in a secure manner. Therefore, and in order to provide better integrity guarantees for reported attributes and context information, we propose that all entities participating in our protocol run on a trusted platform. In our attributes-based model, the client utilizes the properties of a trusted platform to gain access to services being offered by a service provider. Knowing that the client's request originated from a trusted platform provides security assurance to the service provider who will have assurance the client platform has not been tampered and that the integrity of platform services responsible for collecting and reporting contextual information are intact. Our model is not involved in establishing any form of trust between platform's entities that operate on clients' attributes. We assume that this happens via some out of band mechanism and that these entities behave in a trustworthy manner in the sense that they will not reveal clients' identities or disclose any personal information.

-Another key assumption that drove the design of an attribute-based authentication model was that users would be highly mobile and that they may not have users accounts setup on all possible services that are available within the environment. Furthermore, service administrators would find it hard to create users accounts so our model use users' context information to decide whether the entity can access the service or not.

-We assume that the *Trusted Key Generation Center (TKGC)* is capable of setting up system parameters prior to the protocol run. These setting include adding new roles and assigning these roles to users.

-When using temporal attributes, we assume that the clocks of the participating entities are synchronized. This enables entities to check that a service advertisement message or a service reply message is fresh or recent enough.

-Context-providers (i.e sensors and another monitoring devices) are secured from outside attacks. Nevertheless, this area is an active research direction. It also has been the

main focus of security in sensors networks [2, 20]. There are many solutions discussing ways to protect sensors from outside attacks [20, 21, 22, 3].

-Moreover, we assume that our framework will support the dynamic discovery of resources (e.g. sensors or applications) that can be used to check the value of contextual attributes and verify whether a given policy is satisfied, as users move across different pervasive computing environments (e.g. from your office to your car, and to the airport). We consider that this assumption is accomplished during the *Join Phase*.

## 6.4 Context-Based Authentication Scheme

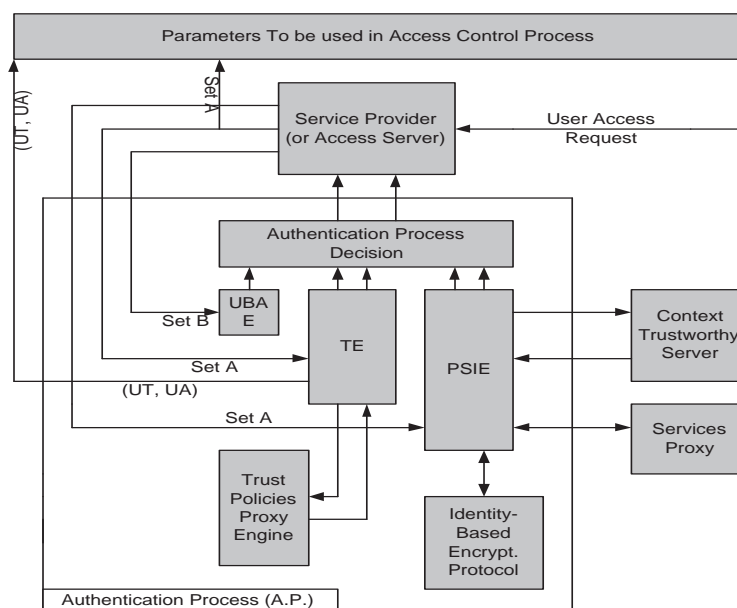
The dynamic nature of a context-aware environment necessitates the need for a very active, flexible authentication mechanism that allows users to securely authenticate and access services with a reasonable level of trust and while privacy is preserved. Our framework consists of the following layers: A Privacy Control Layer (*Layer 1*) for providing users a way for controlling privacy over the reveal of their personal and contextual information. An access layer, (*Layer 2*) which combine authentication process (*SubLayer 2.1*) and access control (*Sublayer 2.2*) process both at context-aware level. The authentication process contains a private set intersection engine and a trust/Risk engine where the trustworthiness parameters values are computed in order to provide a trust-based access to users. Finally, a services/resources Layer (*Layer 3*) for providing and managing services and policies using services proxy server. In the following sections, we detail the functionality of these layers and describe how they interact with one another.

In the following section, we present the access layer architecture scheme. The figure below (Figure 6.3) shows the authentication process architecture. The purpose of access layer is to provide authentication and access control according to user's profile and environment (*attributes-based authentication and access-control*) and then to establish a secure communication link between entities, whilst preserving the privacy of users. Moreover, we will introduce context-aware based user trustworthiness and role's required trustworthiness in order to improve user assignment and role activation.

Our framework is composed of various mechanisms that altogether yield a flexible,

scalable context-aware based authentication. In our model, confidence and trust are defined based on each user's contextual information. First, we introduce the system parameters initialization used for the protocol process. Next, we state the different phases upon which the scheme is based. Finally, we describe the operation of the architecture.

Figure 6.3: The Authentication Architecture Process



Authentication and access control are crucial for information and system security. Most of the identification and authentication schemes for mobile communications are static in nature, and principally dependent on strength of authenticating identifiers for users identity. The acceptance of all the transaction of a user under a single authentication level is vulnerable. The proposed context-aware layer support security and privacy control by using user's situation aware, user's context, and user's environments. Another main purpose of the access layer is also to combine context-aware authentication and context-aware access control. In our opinion, our framework will be a practical application for RFID, smart card, PDAs for pervasive smart environment.

### 6.4.1 The Scheme

**A-Parameters Initialization:** Our infrastructure involves a context-based authentication process, a context-based access control process, a trusted key generation center (*TKGC*), embedded devices *EDs*, Service Providers (*SP*), Inference engines *IEs*, and users denoted by ( $U_i$ ). The trusted Key Generation Center (*TKGC*) chooses two primes order group  $G_1$  and  $G_2$  of prime order  $q$ .  $q$  is a prime which is large enough to make solving discrete logarithm problem in  $G_1$  and  $G_2$  infeasible. The *TKGC* chooses  $G$  as a generator of  $G_1$ , chooses Map-To-Point/Curve function  $H$  and chooses  $e$  where  $e$  is the bilinear pairing map. The *TKGC* computes  $P_{TKGC} = s.G$ , where  $s \in Z_q^*$  is the *TKGC*'s private master key and keep  $s$  secret. We define each user as  $U_i = \langle ID, AK_{ra} \rangle$ , where  $ID$  is a user identity information and  $AK_{ra}$  is a set of assigned keys corresponding to the roles assigned to each user. We defined  $AK$  as  $AK_{ra} = \{K_{IDr_1}, \dots, K_{IDr_n}\}$ . For each user  $U_i$  to be registered, *TKGC* calculates  $Q_i$ , where  $Q_i$  is user's partial public key with  $Q_i = H(ID_i)$ , and determines  $U_i$ 's partial private key  $S_i = s.Q_i$  and calculates  $Q_{SP}$ ,  $Q_{PSI}$  and  $Q_{TE}$  which are the framework entities' partial public key. Moreover, the *TKGC* calculates a user's or an entity's public key [23] as  $P_U = x_u.P_{TKGC} = x_u.s.G$ , where  $x_u \in Z_q^*$  is generated on user's or entity's behavior.

In addition, we define a role as a set of pairs of public and private keys belonging to the role. Each role is represented as  $r = \langle r_{pub}, r_{priv} \rangle$ . When a role  $r_i$  is added to the system, the *TKGC* picks a random  $rpki$  as  $r_i$ 's private key and sets  $RPK_i = rpki.G$  as  $r_i$ 's public key. To assign the role  $r_i$  to a user with an identity  $ID$ , the *TKGC* check the user  $ID$ , computes  $Q_{ID} = H(ID)$ , and generates the user's assigned key  $K_{IDr_i}$  corresponding to  $r_i$  with  $K_{IDr_i} = rpki.Q(ID)$  and where  $rpki$  is the  $r_i$ 's private key.

Finally, *TKGC* sends  $S_i$ ,  $P_U$ ,  $Z$  and the set of  $Q = \{Q_{SP}, Q_{PSI}, Q_{TE}\}$  to the user via a secure channel. The User-Based Authentication Engine *UBAE* manages an stores, for each user  $U_i$  with an *ED*, a record pair consisting of  $\langle Q_i, S_i, s_1, s_2 \rangle$ , where  $(s_1, s_2)$  are the prover's secret. (Table 6.1) shows the mathematical parameters that are to be used in our proposed framework.

In the following, we will propose our model to achieve attribute-based authentication. In our architecture, end-users can interact with the infrastructure (e.g. walking into a room,

Table 6.1: EC Mathematical Notations

Index	Explanation
$TKGC$	The trusted key generation center
$G_1$	An additive group with prime order $q$
$G_2$	An multiplicative group with prime order $q$
$G$	A generator of $G_1$
$P_{pub}$	The public key of $TKGC$ , $P_{pub} = s.G$
$s$	it is chosen from $Z_q^*$ by $TKGC$ , $s$ is kept secret
$ID_i$	The identity of the user $i$ , $ID_i \in \{0, 1\}^*$
$S_i$	The long term private key of user $i$ , $1 \leq i \leq n$
$Q_i$	The long term public key of user $i$ , $Q_i = s.H(ID_i)$ , where $H$ is a Map function
$H_1, H_2$	Hash function
$H$	A map to curve algorithm where an ID is mapped into a point on $G_1$
$e$	$e$ denote a bilinear pairing map
$p, q$	large prime numbers, where $p = 2.q + 1$
$P_1, P_2, P, Q$	Random points over elliptic curve
$a, b$	Random generated private keys
$E$	non-supersingular elliptic curve
$B$	$B \in E(F_q)$ with order $q$
$x(Q)$	$x$ coordinate of point $Q$

entering the subway system using smart phone, PDA, etc). The infrastructure provides a set of resources generally tied to different geographical areas, such as printers, surveillance cameras, campus-based location tracking functionality, and so on. These resources are all modelled as services that can be automatically discovered based on different relevant mechanisms which are out of our band. Our Authentication scheme involves two distinct phases: the *Join Phase*, and the *Mutual Authentication Phase*. We will describe the various interactions that take place between the entities described in our logical system model. We refer our reader to (Figure 6.2) for a comprehensive high level overview of our framework model.

**B-Join Phase:** The purpose of this phase is to automatically support dynamic discovery of services for users through a context-based provision process. In our attributes-based authentication, we aim to have a service provision framework that combines user's profiles and contextual information to select appropriate services to the end users from thousands of

available services. In order to achieve our contribution (**Contribution 4 Achieved (Please check chapter 1 section 1.4 on page 4)**), we have firstly adopted the framework proposed by Qin et al. [15] that automatically provide appropriate services to the right person with the right form with the relevant consideration of contextual information. Therefore, our framework enhances existing service discovery solution by incorporating the use of context information in the matching algorithms, and thus enabling more accurate and relevant results to users. Moreover, we took the assumption that the proposed protocol in [15] is extended to add two new context type fields which will be executed during the provision process. The first context type is related to users with special needs equipped with a body network sensor. This context type is collected by a *BNS* adapter and translated to the provision protocol in order to be proceeded. The second context type is related to a *Meta Classification* process which will be helping in well selecting services. Once, the service provider, *SP*, has initiated the context-aware service provision process, we can go a step forward to start the *Authentication Phase*.

**C-Authentication Phase:** Service discovery typically involves the exchange of service advertisement and service reply messages between the user and service provider. To avoid increasing the communication overheads, we incorporate our extended previous authentication mechanism into these messages (**Contribution 5 Acheived (Please check Chapter 1 section 1.4 on page 4)**). In other words, service discovery and authentication can take place concurrently. We now examine how these messages are constructed to achieve our aim of attributes-based authentication.

$\implies$  Within The First Round, (From: *SP*  $\longrightarrow$  *ED*): Our Attributes-based authentication model will start with a service provider engine advertising available context-aware services to the end user, clients  $C_i$ , as indicated in (6.1).

$$SP \xrightarrow{\text{Advertise Context Aware Services}} C_i \quad (6.1)$$

For example, a location-based service allow providers to advertise its services to any user within a certain acceptable proximity. The advertised service announcement contains the following: A *Universal Resource Locator*, (*URL*), that enable a client  $C_i$  to locate and access the advertised service. *Authentication Requirements (AR)*, allowing clients to

package its access request with the necessary authentication credentials and contextual information. The exchange of traffic between the service provider  $SP$ , the client  $C_i$ , and inference engines is based on an extension for our previous work [8]. For the  $SP$  to construct and send the authenticated services advertisement message, he will be performing the following: The  $SP$  starts the protocol by generating two fresh random nonce  $r_1$  and  $r_2 \in Z_n$ , then he calculates the point  $X$  where  $X = r_1 \times P_1 + r_2 \times P_2$ . Next,  $SP$  constructs the service advertisement message as in (6.2):

$$Adv = (Q_{sp}, (srv_1, srv_2, \dots, srv_n), X) \quad (6.2)$$

Where  $\{srv_1, srv_2, \dots, srv_i\}$  represent the set of available suitable context-aware services defining in the first phase (*Join Phase*). Finally, the service provider encrypts and sends the  $Adv$  message to the embedded device  $ED$ , as given in (6.3).

$$SP \xrightarrow{E_{K_e}(Q_{sp}, URL(srv_1, srv_2, \dots, srv_n), X)} C_i \quad (6.3)$$

In our framework and hereafter, any two entities denoted by  $E_1$  and  $E_2$ , can directly compute a partial private shared key between them without exchanging any previous message. Based on the one's own partial private key and the other party's partial public key, they can directly compute the share key as follows. We denote their partial private key/public key by  $S_{e1} = s.Q_{e1}$ , where  $Q_{e1} = H_1(ID_{e1})$  and by  $S_{e2} = s.Q_{e2}$ , where  $Q_{e2} = H_1(ID_{e2})$ . Now the nodes  $E_1$  and  $E_2$  compute  $K_{e1/e2} = e(S_{e1}, Q_{e2})$  and  $K_{e2/e1} = e(Q_{e1}, S_{e2})$  respectively. And finally the private shared key will be  $K_e$  where

$$K_{E_1/E_2} = H_2(K_{e1/e2}) = H_2[e(Q_{e1}, Q_{e2})^s] = H_2(K_{e2/e1}) = K_{E_2/E_1} = K_e \quad (6.4)$$

This approach is very efficient in terms of communications and computations and this feature makes it very attractive to the environments where the entities capabilities are limited.

$\implies$  Within The Second Round, (From:  $ED \longrightarrow SP$ ): After receiving the advertised service announcement, the client  $C_i$  decrypt the message and retrieve the credentials. Suppose that the client is interested in an advertised service  $srv_i$ , (i.e, request access to perform

an operation  $O$  on service  $srv_i$  from the service provider), he will be performing the following: As  $srv_i$  is a context-based resource,  $C_i$  is promoted to present not only identity credentials but also all the required contextual information and bundle them with the access request that is sent to  $SP_i$ . In our attribute-based authentication model, authentication requirements are dynamic and can vary dramatically from one access to the next. Moreover, we must expect that some attributes will be generated by the user while others by the platform. Our model provides the client with a full control over the reveal of the personal information. The option of collecting contextual information attributes from the platform is done by using a *Privacy Control Layer (PCL)* (**Contribution 6 Achieved (Please check Chapter 1 section 1.4 on page 4)**). In order to retrieve needed attributes to fulfill the access request, the user issues a service request which is handled by the user agent. The user agent does not directly invoke the service. Instead, it retrieves the privacy policy of the service, without revealing any information about the user. The user agent compares the service's policy to the user's preferences and performs the following:

Based on the users preferences:

**-1-:** If there is a preference rule that accepts the privacy policy, then: - Extract the context-dependent preferences from the users extended preferences document. - Store an association between the user, the service and the users context-dependent privacy preferences in the platform and finally a request for contextual information is issued to the *PCL*.

**-2-:** If there is no accepting rule, or there is a rule that indicates that the user should be alerted, then the service will not be invoked. The user is prompted for further evaluation of the policy.

Whenever a request for contextual information arrives at the privacy control layer, *PCL* should performs the following actions:

**-1-:** Check for an association record between the service that is requesting the contextual information and the user about whom information is requested. If this association does not exist, try to contact the users agent and ask it to store an association record.

**-2-:** Retrieve and evaluate the context-dependent preferences referenced in the association. **a:** If the context -dependent preferences evaluate to true, then retrieve the requested information from the context interpreter and return the information to the user agent. **b:** If the context-dependent preferences evaluate to false, then refuse the request for contextual

information.

When The *PCL* is introduced in the infrastructure, the access request itself is altered to include information that was provided by the *PCL*. Context-Aware providers will publicize to their users information such as positions, roles, activities, etc. The validity of these data could be verified by introducing *Context Trustworthy Engine, CTE* in the framework. This is the role of the authentication process, using the *CTE*, to validate these data before starting the authentication process. After receiving relevant reply message from the *PCL*, the user agent retrieves the set of contextual information received from the attribute provider(s) through the *PCL*, and performs the following: The queried *ED* selects the role or the corresponding set of roles denoted by  $SR = \{r_1, r_2, \dots, r_h\}$ . Generates the message  $Q$  and calculates the signature  $Sig_Q$  on  $Q$  with  $Q = S_i|SR|p_{er}$  and where  $p_{er}$  is the permission that the user wants to enforce. The  $Sig_Q$  is denoted by  $\langle U, V \rangle$ . In addition, *ED* generates two fresh random nonces  $f$  and  $a$ , where  $f \in_R Z_2^t$  and  $a \in Z_q^*$ , she calculates  $T_{ED}$ , where  $T_{ED} = a.G$ . For a static context-less system, the user computes  $(R_x, T_x)$ , where  $(R_x, T_x)$  is the signature pair over the user's private key  $S_i$ . This  $(R_x, T_x)$  will be replacing the couple  $\langle U, V \rangle$  in equation (6.6) for the protocol process run. Finally, the client will package all the collected attributes encrypted (i.e., user's profile and environment's attributes) with needed information in order to be sent to the service provider for authentication process. Let assume that a user,  $U_i$ , has received the request set of context-data  $D$  from the privacy control layer. Therefore, the set  $A$ , given as in (6.5), denotes all the attributes that user  $U_a$  may present to set her rules in the authentication process.

$$A = \{A_{C_i}, A_{AP_i}\} = \{a_1, \dots, a_i, b_1, \dots, b_j\} = \{ca_1, \dots, ca_i, ca_{i+1}, \dots, ca_j\} \subseteq D \quad (6.5)$$

Where  $D$  is the reference set that contains all the attributes a user may hold or the context data received and  $ca$  represent the collected context-aware data. Finally, the client packages the final required set of context and attributes that the service provider may use for authentication process and construct the message as described in (6.6).

$$C_i \xrightarrow{E_{K_e}(Q_{C_i}, (srv_i), E_{K_{U/PSI}}(ca_1, ca_2, \dots, ca_i, ca_{i+1}, \dots, ca_j)), T_{ED}, f, \langle U, V \rangle} SP_i \quad (6.6)$$

Hereafter, these attributes are mapped into integer numbers  $ca_i$  for  $i = 1, 2, 3, \dots, l$ ; that is  $ca_1$  is a number representing *Name*,  $ca_2$  is a number representing *Location*, and so on. Our model is very flexible in that the service provider engine may accept or refuse a subset of attributes in  $A$  corresponding to different level of confidence. If the user can present all attributes in  $A$  required by service provider in order to access for identification, a full confidence will be achieved, otherwise the confidence level will be depending both on PSI's reasoning process and on the user's requirements by computing user's trustworthiness and role's required trustworthiness. In the next section, we will demonstrate how our scheme could be combined with *Timed Fuzzy Logic* [18] in order to set a threshold under uncertainty and to account for changes in context-data.

$\implies$  Within The Third Round, (From  $SP \longrightarrow IEs$ ): The service provider now has an authentication package, containing the requested context attributes, that was provided by the client. The first step requires the  $SP$  to decrypt the encrypted message and retrieve the data in order to determine the source and authenticity of these attributes provided by both  $U_i$  and  $AP$ , and later on to complete authentication process. Once the service provider has retrieved the data set from equation (6.6), the authentication process will be performed as follows: The service provider send the encrypted set  $A$  where  $A = \{ca_1, ca_2, \dots, ca_j\}$  to both  $PSIE$  and  $TE$  engines, and send  $B = \{\langle U, V \rangle, f\}$  to  $UBAE$  engine.

The service provider's platform is composed of the two main processes. The authentication process and the access control process. Each of these processes contains different relevant engines that they interacted altogether provide a flexible, and scalable context-aware authentication framework. For the authentication process (Figure 6.3), we have the following engines: a Private Set Interaction Engine ( $PSIE$ ), a Trust Engine ( $TE$ ), and a User-Based Authentication Engine ( $UBAE$ ). We also have an Identity Based Encryption Engine ( $IBEE$ ) that will be responsible for setting a shared secret key for secure future communications. This  $IBE$  protocol will be interacting with the  $PSI$  in order to calculate the shared secret key. Moreover, the  $PSI$  engine will be interacting with the  $CTE$  engine to accomplish the attributes verification process. Therefore, Our authentication process decision will be based on the output of these several engines (**Contributions 3 and 7 Achieved (Please check chapter 1 section 1.4 on page 4)**). The description of these engines and their interacting process will be explained in the coming section.

$\implies$  Within The Fourth Round, (From:  $IE_s \longrightarrow SP$ ): Upon receiving the encrypted messages from the service provider, the  $PSI$  start the attributes verification process. To verify the source of  $AP$ 's attributes, we have introduced the *Context Trustworthy Engine* ( $CTE$ ) which is responsible for verifying all attributes provided by  $AP(s)$  and other contextual information provided by the client (i.e., case of an RFID or a smart card and a client with special need). The interactions (6.7) and (6.8) show the  $PSI$  requesting the  $CTE$  to verify the validity of the attributes  $ca_j$ .

$$PSI \xrightarrow{E_{K_e}(Q_{SP_i}, (ca_1, ca_2, \dots, ca_j))} CTE_i \quad (6.7)$$

and

$$PSI \xleftarrow{E_{K_e}(Q_{CTE_i}, (Verification\ result))} CTE_i \quad (6.8)$$

Once  $PSI$  determines the verification process of these attributes provided on behalf of the client, it passes the authentication credentials and attributes to the relevant engines that will complete the processing of the client's access request. Each engine will start its own process as follow:

**Description of The  $PSI$  Engine:** One new component that will be added to our architecture is the notion of Private Set Intersection Engine ( $PSIE$ ).  $PSI$  are cryptographic techniques allowing two or more parties, each holding a set of inputs, to jointly identify the intersection of their inputs sets (i.e, shared context), without leaking any information about credentials that each entity might have. Nevertheless, both entities, the prover and the verifier, need to protect their credentials from each other. Moreover, any entity awaiting to be authenticated by a server has to establish enough confidence in it and be able to present the required attributes (**Contribution 3 Achieved (Please check Chapter 1 section 1.4 on page 4)**). Therefore, the conditions that the server sets for authentication become extremely valuable, as they determine the reasoning mechanisms in the authentication protocol (**Contribution 7 Achieved(Please check Chapter 1 section 1.4 on page 4)**). To keep a high level of security, the server needs to keep those attributes private. For this purpose, we make use of the Private Set Intersection ( $PSI$ ). Once, The  $PSIE$  receive and extract/decrypt the set  $A$  of attributes and upon the sender request's selected  $srv_i$ , the  $PSIE$  will initialize a

PSI protocol over the two sets  $A$  and  $S_{srv_i}$ . Where  $S_{srv_i} = \{S_{srv_i1}, S_{srv_i2}, \dots, S_{srv_ij}\}$  represent the needed set of contextual information defined by the service  $srv_i$  administrator deployment. The  $S_{srv_i}$  set reside on a Services Proxy Server  $SPS$ , and the PSI protocol will be initialized between  $PSI$  engine and  $SPS$ . There are many  $PSI$  protocols in the literature. We can adopt the one that was chosen by [16, 18] since it has a provision for approximate matching, referred to as *Fuzzy Private Match*. The  $PSI$  Inference engine performs two kinds of tasks: First, it gives a level of confidence when a user is on an authentication process. It makes use of authentication contextual information to assign the confidence level. Second, it evaluates a Fuzzy Logic Matching protocol queries from applications about whether a certain entity is allowed to access a certain resources. It makes use of applications specific contextual information, the credentials of the entity, and entity's contextual information to decide whether an entity is authenticated and has access to resources. For convenient readerships, we recommend our readers who want to go deeper in the theory of *Fuzzy Private Matching Protocol* and getting acquainted with the principles of  $PSI$  theory to refer to (*Appendix A.2*). Moreover, the  $PSI$  engine will be also interacting with the identity Based encryption protocol to calculate the secret shared key. This step will be discussed as follow:

**Description of The Identity-Based Encryption Protocol** The  $IBE$  removes the need to set and exchange certificates as the message can be encrypted based on the identity of the entities. The identity can be defined as a location, name, email address, time,... or a combination of them. The combination of them could be refereed to the context data. For convenient readership, we recommend our readers to refer to [5]. In the following, we will describe the details of how  $PSI$  interacts with  $IBE$  protocol in order to calculate the shared secret key. From the  $PSI$ , let  $A \cap S_{srv_i}$  be the intersection set of  $A$  and  $S_{srv_i}$  defined above.

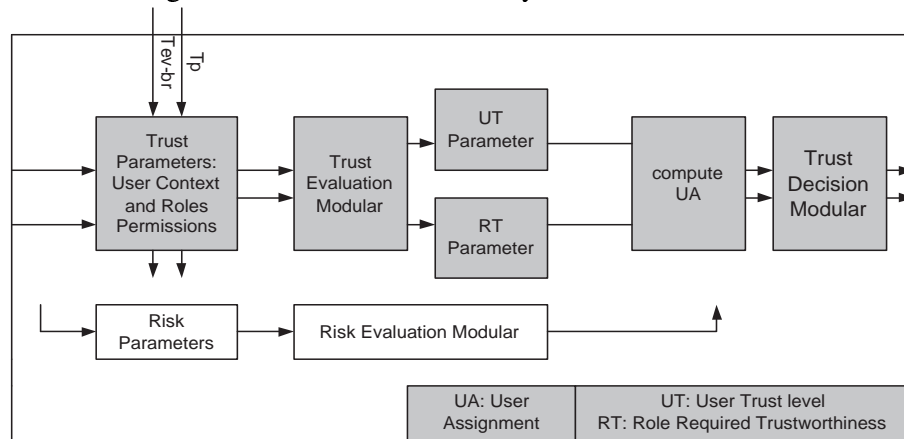
$$A \cap S_{srv_i} = \{d_1, d_2, d_3, \dots, d_i\} \quad (6.9)$$

where  $d_i$  denotes the context that are shared between the user and the service provider. Finally, the  $IBE$  will calculate and send  $T_{SP}$  to  $PSI$  engine with  $T_{SP} = (\sum d_i).G$

**Description of the Trust Engine:** Another new component that will be added to our architecture is the notion of Trust Engine ( $TE$ ). To trust pervasive computing, we must be

able to manage privacy, confidentiality, availability, and controlled access to digital information as it flows through the systems. In the following, we will describe the Trust Layer architecture. The figure below (Figure 6.4) shows the Trust Layer design architecture. Our ultimate goal is to provide a trust model that is flexible and adaptive for applications scenarios and environments. This approach could be solved using the concept of fuzzy-based trustworthiness (**Contribution 7 Acheived (Please check Chapter 1 section 1.4 on page 4)**).

Figure 6.4: The Trust/Risk Layer Architecture -L2-



In this section, a dynamic trust model is formally introduced to incorporate trust strategies in order to first build up the user's and role's required trustworthiness level and then the User Assignment  $UA$  trustworthy value. There are several ways and approaches to design trust models. The component-based approach is chosen for our model design because it can be implemented in a distributed way and be extended easily and transparently (i.e., To include later the Risk Assessment Engine). During a real-time trust management process in pervasive computing environments, the trust information may be from different resources at any time. Therefore, the our adopted trust model is designed to be able to evaluate the trust information concurrently. Using this approach, the trust engine derives the *level* trustworthiness of a user  $UT$  and role's required trustworthiness  $RT$  by using users attributes and roles permission, respectively. The user assignment  $UA$  level is performed based on the trust level  $UT$  in comparison with the trust level  $RT$ . However, our trust model is based

on the trust policies, the environment contextual information, and the users roles permissions. As a cognitive process, trust is complex and fuzzy. That is, for a special context, we can not easily make a decision about whether to trust an entity or distrust it. Therefore, Our *Trust* evaluation engine is adopted as a combination from [1, 4] where trust model is provided by integrating trust into a fuzzy logic-based trusted decision upon building the trustworthiness's prediction. For convenient readership of this work, we will describe the trust model process here: Trust establishment can be thought as a process that identifies or verifies the principal's claim against the trust evidence. Trust evidence,  $T_{ev}$ , are further classified into the following categories: credentials, the context of the environments, and behavior records. We denote  $T_{ev} = \{T_c, T_{ce}, T_{br}\}$ . We define user trust level,  $UT$ , of the user by a Function  $F$ , as given in (6.10)

$$UT = F_{res}(T_p, T_{ev}) = F(T_p, T_c, T_{ce}, T_{br}) \quad (6.10)$$

Where  $F_{res}$  is the function of the trust level of the client to access the resource and  $T_p$  is the set of trust policies for the resources. In our definition, The trust level of the user,  $UT$ , for accessing the resource in the system is determined by evaluating the trust evidence against the trust policies for the resource and the user assignment.  $UA$ , is evaluated based on  $UT$  in comparison with  $RT$ . For simplicity, we will consider  $T_{ev} = T_{attributes} = T_a$  and finally  $UT = F(T_a)$ .  $F$ ,  $UT$ , and  $RT$ , parameters could be calculated using the formal mathematical equations from [1]. We urge our readers who want to go deeper in the theory of *Trust Evaluation* to refer to (*Appendix A.1*). Once these parameters are calculated, the trust decision modular will evaluate the user assignment  $UA$  based on  $UT$  in comparison with  $RT$ , and will package the final result in order to be sent to the *Authentication Process Decision*.

**Description of the UBA Engine:** Moreover, upon receiving the encrypted signature pair message  $E_{K_e}\langle U, V \rangle$  from the service provider, the *UBA* engine will decrypt the message, then verify the signature pair, if it is valid, then the *UBA* engine accept, and the pair  $(s_1, s_2)$  associated with the authenticated *ED* is extracted from the database server, and encrypted using the Weil-Pairing-based encryption algorithm. Finally, the user based authentication engine packages the encrypted message  $E_{K_e}(s_1, s_2)$  with the evaluated result

in order to be sent to the authentication process decision.

The authentication process decision will take the decision based on its different engines evaluation and package the final output result and send it encrypted to the service provider.

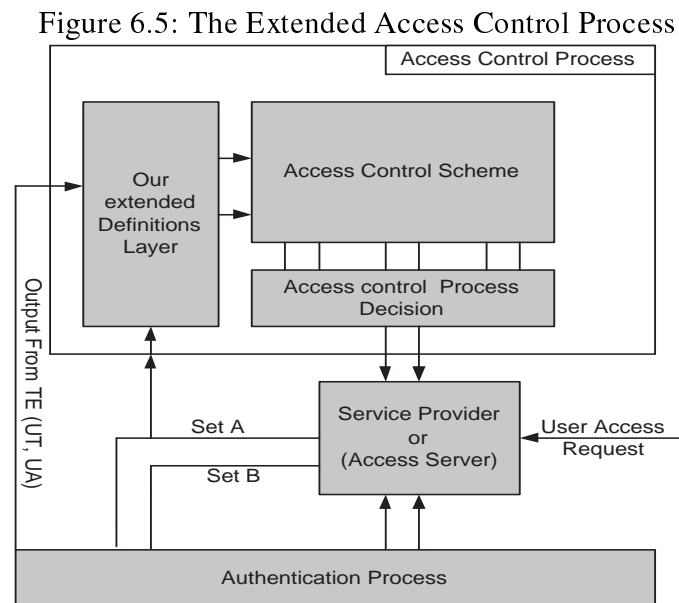
$\implies$  Within The Fifth Round, (From:  $SP \rightarrow ED$ ): Upon receiving the message from the authentication process decision, the service provider first decrypt the message and then evaluates the output. If the result is false, he will denied access request to resources, otherwise, if true (i.e., *the user is authenticated, the user trustworthiness parameters are acceptables, and the confidence level is acceptable*) the service provider extracts the pair  $(s_1, s_2)$  and then computes

$$y_i = (r_i + (f \times s_i))(\text{mod } n) \quad (6.11)$$

for  $i = 1$  and  $2$  and starting packaging the following data ( $T_{SP}$ , and  $y_i$  for  $i = 1$  and  $2$ ) in order to be sent later to the  $ED$ . Meanwhile, as the final decision will be evaluated based on both authentication and access control process decisions, the user's access request is also subject to context-aware access control rules which will be discussed in the following:

**Context-Based Access Control Process** A key challenge in ubiquitous environment is the design of an effective active access control schemes [1] that can adequately meet the security challenges represented by the system's ability to capture security relevant contextual information, such as time, location, user's profile, or environmental state available at the time the access request are made and to incorporate these information in its access control process. We specify and integrate our own context-aware access control rules definitions to further enhance the security of our proposed authentication-based framework scheme (**Contribution 8 Achieved (Please check Chapter 1 section 1.4 on page 4)**). Moreover, the context directly affects the level of trust associated with a user, and hence the authorizations granted to him. Therefore, we introduce the user trustworthiness and role's required trustworthiness parameters into the design the context-based access control by incorporating them within the development of the context constraints (**Contribution 7 Acheived (Please check Chapter 1 section 1.4 on page 4)**). Conditions on the access control to solve the semantic problem is to check the trust engine parameters  $UT$

and  $UA$ , if they satisfy the condition, the user will be subject to authorization rules and policies based on the available presented attributes. We believe that the introduction for the rules definitions is necessary for providing an adequate authorization decision for any *Service\_Access\_Request* and to accomplish a secure authentication process. In the following figure, (Figure 6.5), we show our extended access control scheme with the rules definitions.



In the following, we describe needed rules definitions for a dynamic context-aware access control infrastructure to fulfill the framework's security requirements:

**Rule Definition 1: *Dynamic Adjustment*** In our approach, we believe that any pervasive model should dynamically adjust role assignments and permission assignments based on presented context information. Therefore, we consider DRBAC concept [19] where each user is assigned a set of roles and the context information is used to decide which role is active at a time. User will access the resource with the active role. Moreover, each role is assigned a set of permission, where the context information will be used to decide which permission is active for that role. The systems-based context for resources should be taken into consideration, and the security policy for the resources should be able to define a permission transition for a current role.

**Rule Definition 2:** *Context Type:* A context type is defined as a property related to every participant in a service. In simple scenario, context type may be a concrete property familiar in everyday life, such as time or location, etc. However, in a more complex scenario, we believe that context type should be extended to describe more attributes such as user's capability and/or willingness (i.e., case of people with special need equipped with a hidden body network sensor). We define such context type by  $CT_c$ . Therefore, based on a complete users' context types  $CT_i$  we can define that each resource  $r_i$  has its own context set  $CS_{r_i}$ , which is defined as follows:

$$CS_{r_i} = \{CT_1, CT_2, \dots, CT_c, \dots, CT_n\} \quad (6.12)$$

In any access control design to be integrated within our framework, we define two sets of context types, passive and active sets. While the authentication process will be subject to only the active set, the access control decision will be subject to the two sets.

**Rule Definition 3:** *Context Constraint:* We define our context constraint as a regular expression that is capable of specifying any complex context related constraint to introduce all kinds of security requirements. In general a context set is defined as follows: *Context Constraint* :=  $CC := Clause_1 \cup Clause_2 \dots \cup Clause_i$  where *Clause* :=  $Condition_1 \cap Condition_2 \dots \cap Condition_j$  and where *Condition* :=  $\langle CT \rangle \langle OP \rangle \langle VALUE \rangle$ , where  $CT \in CS_{r_i}$ ; *OP* is a logical operator in the set  $\{>, \leq, <, \geq, \neq, =\}$ , and *VALUE* is a specific value of *CT*. Therefore, we suggest that should be extended to accommodate user trustworthiness *UT* and user assignment trustworthiness *UA* as a new clause. The new context constraint will be as follows:

$$CC := Clause_1 \cup Clause_2 \dots \cup ((UT \geq VALUE) \cap (UA \geq VALUE)) \quad (6.13)$$

As an illustration, suppose we have a context set  $CS = \{\text{Time, Location, Authentication Level}\}$ , and we have a partial security rule such as a patient data can be accessed from within the hospital between 8am and 5pm with a trust level of a password; otherwise a higher level of trust is required.

**Rule Definition 4:** *Authorization Policy:* We define an authorization policy as a quadruple,  $AP = \langle S, \langle P, O \rangle, CC \rangle$  where *S* is the subject in this policy, which

could be a user or a set of roles.  $P$  is the mode of operation defined by READ, APPEND, DELETE, UPDATE, WRITE,  $O$  is a data object and  $CC$  is a context constraint defined according to *definition 3*.

**Rule Definition 5:** *Resource\_Access\_Request*: The *Resource\_Access\_Request*, denoted by  $RAR$ , is defined as a quadruple  $RAR = \langle U_i, \langle P_i, O_i \rangle, RC_i \rangle$  where  $U_i \in UserSet$ ,  $P_i \in PermissionSet$ ,  $O$  is the data object requested, and context  $RC_i$  is a runtime context set of values for every context type in the context set  $CS$ .  $RC_i$  is defined according to *Definition 2* and captured dynamically at the time of the access request.

**Dynamic Context Evaluation:** Finally, the access control decision for any service access request  $RAR = \langle U_i, \langle P_i, O_i \rangle, RC_i \rangle$  is granted only if there exists an authorization policy  $AP = \langle S, \langle P, O \rangle, CC \rangle$ , such that  $U_i \in S$ ,  $\langle P_i, O_i \rangle = \langle P, O \rangle$ , and  $CC$  evaluates to true under  $RC_i$  (that is, when all CTs in constraint  $CC$  are replaced with their **available presented values** in  $RC_i$ , then the resulted Boolean expression is true).

$\implies$  Finally, the service provider evaluate the final access request decision (the one from authentication process and the other from the access control process) and packages the results with the relevant data ( $T_{SP}$ ,  $y_i$  for  $i = 1$  and  $2$ ) and send it to the user with the embedded device. The  $ED$  computes

$$\left( \sum (y_i \times P_i) + f \times Z \right) \quad (6.14)$$

and then checks that if  $(\sum (y_i \times P_i) + f \times Z)$  is equals to  $X$ , if so the  $ED$  accepts and extract the shared secret key in order to be used for encrypting future communications, else rejects.

After the above messages,  $T_{ED}$  and  $T_{SP}$  are exchanged, the reader and the user can agree and compute the secret shared key

$$K_{SP/ED} = e(Q_{ED}, P_{ED})^b \cdot e(x_{sp} \cdot S_{sp}, T_{ED}) \quad (6.15)$$

and

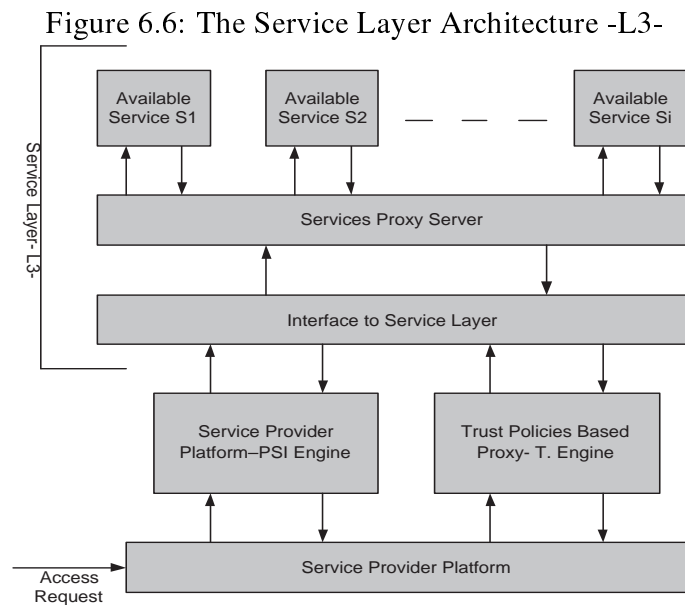
$$K_{ED/SP} = e(Q_{SP}, P_{SP})^a \cdot e(x_{ed} \cdot S_{ed}, T_{SP}) \quad (6.16)$$

respectively. We denote by  $K = K_{SP/ED} = K_{ED/SP}$ . Hence, the key  $K$  is a shared between the entities. To ensure forward security, we can use a the new shared key  $K_h$  after applying a hash function to  $K$ . Once the protocol run completes successfully, both parties may use the  $K_h$  to encrypt subsequent session traffic in order to create a confidential communication channel. In the following we will present a brief verification regarding the similarity of the shared key equations:

$$\begin{aligned}
 K_{SP/ED} &= e(Q_{ED}, P_{ED})^b \cdot e(x_{sp} \cdot S_{sp}, T_{ED}) \\
 &= e(Q_{ED}, x_{ed} \cdot s \cdot G)^b \cdot e(x_{sp} \cdot S_{sp}, a \cdot G) \\
 &= e(x_{ed} \cdot s \cdot Q_{ED}, b \cdot G) \cdot e(x_{sp} \cdot s \cdot Q_{SP}, a \cdot G) \\
 &= e(x_{ed} \cdot S_{ed}, T_{SP}) \cdot e(Q_{SP}, P_{SP})^a \\
 &= K_{ED/SP}
 \end{aligned} \tag{6.17}$$

## 6.5 Service Layer Process

In the following, we will briefly describe the Services Layer architecture. The figure below (Figure 6.6) shows the Service Layer design architecture.



Our Service Layer specify a series of requirements in terms of proxies, engines and

actuators that clients must meet in order to be eligible to receive their service. Our service layer architecture contains a service proxy layer which will be used to retrieve contextual information, services policies and trust policies parameters. These parameters will be used by our authentication and access control processes to perform the framework functionality and decision making. Additionally, we assume that the service layer hold the logic and processing power needed for executing the services it offers.

As a summary, a context-based access is developed and it can thus be granted to both known and unknown agents. The integration for the IEs engines, the extension for the context-based access control definitions, and the development of IBE engine form the core of our context-based authentication framework where every request is authenticated and filtered in order to remove any unauthorized actions. After filtration the service provider can evaluate the request and create an appropriate response depending on the contextual information.

## 6.6 Framework Interaction Summary

In the following, we will give a summary of the interaction that take place when a user want to invoke a context-aware service. We consider these interactions occur after the user have received the advertisement message and wants to invoke a service. Figure 6.2 shows the following interaction steps:

0. The purpose of this step is to provide users' dynamic discovery of services through the context-based provision process.
1. The user tells the user agent what context-aware service it should invoke.
2. The user agent obtains the privacy policy of the context-aware service.
3. The user agent compares the policy to the users preferences, and if the policy is acceptable, it registers the users context-dependent preferences in the platform.
- 4 & 5. The user agent invokes the context-aware service through the privacy control layer and request the service's contextual information.
6. The privacy control layer checks whether the user about whom the context-aware service is requesting information has registered context-dependent preferences, and evaluates these preferences.

7. If the context-dependent preferences are satisfied, the privacy layer passes the request for the contextual information to the context interpreter.

8. The context interpreter processes the request and returns the requested information to the privacy layer.

9. The privacy layer returns the contextual information to the user agent.

10. The user agent packages all the needed data that fulfill the access request and sends the data package to the Service provider (or access server).

11. The service provider (or access server) processes the data package and forward each, the authentication process and the access layer, the corresponding set of data.

12. The authentication and access control processes evaluate the data through their built-in engines and reply with a relevant access request decision to the service provider (or access server).

13. The service provider evaluates the access request decisions and returns the final request decision to the user agent.

14. The user agent evaluates the request decision and displays the context-aware services result to the user.

## 6.7 Security Analysis and Discussion

Our proposed architecture is considered to provide privacy and anonymity for users. In the following, we evaluate our architecture regarding the security and privacy requirements.

-Mutual Authentication: Considering the fact that the digital signature pair  $(U, V)$ , created by the *ED*, is verified by the Back-end server. Considering that the pair  $(s_1, s_2)$ , sent by the back-end server, is recalculated by the reader under  $(y_1, y_2)$  and verified by the *ED*. Therefore, our proposed architecture guarantees the secure mutual authentication between the embedded device *ED* and the back-end server.

-Passive attack: Suppose an attacker performs a passive attack, then the session will terminate with both legitimates parties accepting. That is, the two parties successfully identify themselves to each other. And regarding the fact that the exchanges messages between the reader and the *ED* are generated from random nonce which are generated with every new session, so it is infeasible that an attacker computes any useful information

including the  $ID_i$  of a user  $U_i$ . Therefore the architecture resists against the passive attack.

-Man in the middle attack (or active attack): Suppose that an attacker intercepts  $X$  and replaces it with  $X'$ , the attacker then receives  $f$  and  $(U, V)$  from the  $ED$ . He would like to replace the pair with  $(V', V')$ , as before. However, and unfortunately for the attacker, he can not compute the value of the new pair because he does not know the users credentials and parameters and because the transmitted messages are meaningless. Therefore the proposed scheme thwarts the man in-the-middle attack.

-Perfect forward secrecy: Each run of the protocol computes a unique  $x$ , a unique Signature pair  $(U, V)$  and a unique pair  $(y_1, y_2)$ . In addition the transmitted messages are meaningless as they are generated for each new session using new random nonce. Thus, the architecture is secure against perfect forward secrecy.

-Data Confidentiality: Since our architecture provides secure mutual authentication between the  $ED$  and the system and since the information transmitted between the  $ED$  and system is meaningless, thus, our architecture provide data confidentiality and the user privacy on data is strongly protected.

- $ED$  Anonymity and Location Privacy: During the authentication processes, a signature algorithm is used to produce the signature pair  $(U, V)$ . The pair  $(U, V)$  and  $f$  that are transmitted between the  $ED$  and  $R$  are randomized and anonymous since they are updated for each read attempt. Thus, our architecture provides user anonymity and location privacy is not compromised.

-Unauthorized Reader Detection: Our Proposed architecture is based on the insecure communication channel between  $R$  and back-end server. The unauthorized reader  $R'$  is detected and prevented by the back-end server  $DB_{ID}$  using the weil pairing based encryption algorithm between the reader and the back-end server, and by verifying the pair  $(y_1, y_2)$  by the legitimate user or  $ED$ . Thus, our scheme protects against Unauthorized reader.

## 6.8 Conclusion

We identified security and privacy threats that may arise during access services in a pervasive computing environments; we also derived corresponding security and privacy requirements. We presented our attributes-based authentication scheme, using Trusted Computing

Functionality, which preserves user privacy. The scheme also satisfies all the identified security requirements. To a user and service provider, security and privacy are both desirable, but they are potentially conflicting requirements, and it is challenging to achieve them both. However, this is achieved by our attributes-based authentication scheme presented here, enabling secure access to services while privacy is preserved. In the coming chapter, we implement/integrate the proposed model into the existed platform.

# Bibliography

- [1] T. Hassan, A. Morteza, and J. Rasool *Trust-Based User-Role Assignment in Role-Based Access Control*, In Proceeding of the IEEE/ACS International Conference on Computer Systems and Applications (AICCSA), pp. 807-814, 2007.
- [2] P. Adrian, S. John, and W. David *Security in Wireless sensor Networks*, In Proceeding of the ACM Communications, 47(6):53-57, June 2004.
- [3] W. David, *Resilient Aggregation in Sensor Networks*, In Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks, pp. 78-87, ACM Press, 2004.
- [4] D., Xiu, and Z., Liu, *A Dynamic Trust Model for Pervasive Computing Environments. A Research Paper*, Research Supported by the NSF0406325, 2004. [coitweb.uncc.edu/zh-liu/Research/Papers/asc.pdf](http://coitweb.uncc.edu/zh-liu/Research/Papers/asc.pdf)
- [5] B. Dan, and F. Matthew, *Identity-Based Encryption from the Weil Pairing*, In Proceeding of the SIAM Journal of Computing, 32(3):586-615, 2003.
- [6] L. Marc, *A Privacy Awareness System for Ubiquitous Computing Environments*, In Proceeding of the 4th International Conference on Ubiquitous Computing, (UbiComp) pp. 237-245, 2002.
- [7] P. ABI-CHAR, A. Mhamed, B. EL-Hassan and M. Mokhtari, *Controlling Trust and Privacy in Context-Aware Environments, State of Art and Future Directions*, In Proceeding of the IGI Publishing under Book Title: Trust Modeling and management in Digital Environments: From Social Concept to System Development. Book Edited by Nokia Research Center, Finland, 2009. Chapter Accepted, (To Appear).

- [8] P. ABI-CHAR, A. Mhamed, B. EL -Hassan and M. Mokhtari, *Towards a Robust Privacy and Anonymity Preserving Architecture for Ubiquitous Computing*, In Proc. of the Third International Conference on Risks and Security of Internet and Systems (CRISIS08). Tozeur, Tunisia, IEEE Computer Society Press, October 28-30, 2008, pp. 125-132.
- [9] B. Jakob, K. Rasmus, P. Michael, *Context-aware user authentication: Supporting proximity-based login in pervasive computing*, In Proceeding of International conference on ubiquitous computing No5, Seattle WA, vol. 2864, pp. 107-123, 2003.
- [10] Z. Martijn, *A P3P-Based Privacy Architecture For A Context-Aware Service Platform*, Master thesis, University of Twente, Netherlands, August 2003. Available at <http://asna.ewi.utwente.nl/education>
- [11] J.M. Covington, M. Sastry, and D.J. Manohar, *Attribute-Based Authentication Model for Dynamic Mobile Environments*, In Proceeding of the Third International conference on Security in Pervasive Computing (SPC), York, UK, pp. 227-242, 2006.
- [12] M.D. Corner, and B.D. Noble *Protecting Applications With Transient Authentication*, In Proceeding of the First International Conference on Mobile Systems, Applications and Services, pp. 57-70, 2003.
- [13] S.J. Creese, M.H. Goldsmith, and B.R. Zakiuddin, *Authentication in Pervasive Computing*, In Proceeding of the First International Conference on Security in Pervasive Computing (SPC), 2003.
- [14] D. Glynos, P. Kotzanikolaou, and C. Douligeris, *Preventing IMpersonation Attacks in MANET with Multi-Factor Authentication*, In Proceeding of the Third International Symposium on Modeling and Optimization in Mobile Ad-hoc, And Wireless Networks, pp. 59-64, 2005.
- [15] W. Qin, Z. Daqing, M. Mounir, S. Yuanchun, and D. Kejun, *Combining User Profiles and Situation Context for Spontaneous Service Provision in Smart Assistive Environments*, In Proceeding of the 5th international conference on Ubiquitous Intelligence and Computing, pp. 187-200, 2008.

- [16] M. Behzad, M. Ali, and K. Ahmed, *A Framework for Context-Aware Authentication*, In Proceeding of 2008 IET 4th International Conference on Intelligent Environments, pp. 1-8, 2008.
- [17] L. Adrian, and M.J. Chris, *Ninja: Non Identity Based, Privacy Preserving Authentication for Ubiquitous Environments*, In Proceeding of 9th International Conference on Ubiquitous Computing, (UbiComp2007), Lectures Notes in Computer science, Springer 4717, pp. 73-90, 2007.
- [18] J.F. Michael, N. Kobbi, P. Benny, *Efficient Private Matching and Set Intersection*, In Proceeding of the Advances in CryptologyEurocrypt'2004', vol. 3027 of Lectures Notes in Computer Science, Springer-Verlag, pp. 1-19.
- [19] G. Zhang, and P. Manish, *Context-Aware Dynamic Access Control for Pervasive Applications*, In Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference, (CNDS'04), USA, 2004.
- [20] P. Adrian, S. Robert, W. Victor, E.C. David, and J.D. Tygar, *SPINS: Security Protocols for Sensor Networks*, In Proceedings of the Seventh Annual ACM International Conference on Mobile Computing and Networking, MobiCom2001, pp. 189-199, 2001.
- [21] P. Bartosz, S. Dong, and P. Adrian, *SIA: Secure Information Aggregation in Sensor Networks*, In Proceedings of ACM on Sensor Networks, pp. 255-265, 2003.
- [22] S. Naveen, S. Umesh, and W. David, *Secure Verification of Location Claims*, Technical Report UCB/CSD-03-1245, EECS Department, Berkeley, 2003.
- [23] S. Wang, Z. Cao, and H. Bao, *Efficient Certificateless Authentication and Key Agreement (CL-AK) for Grid Computing*, In Proceeding of the International Journal of Network Security, vol.7, No.3, pp. 342-347, 2008.
- [24] C. Wang, L. Fang, and Q. Wang, *Zero-Knowledge-Based User Authentication Technique in Context-aware System*, In Proceeding of the International Conference on Multimedia and Ubiquitous Engineering, pp. 874-879, 2007.

- [25] G. Chen, and D. Kotz, *A survey of Context-aware Mobile computing research*, In Publication of Dartmouth College, Computer Science Departement Technical Report TR2000381, 2000.
- [26] K., Ren, and W., Lou, *Privacy-Enhanced, Attack-Resilient Access Control in Pervasive Computing Environments with Optional Context Authentication Capability*, In Springer Science LLC 2006, Mobile Netw Appl (2007)12:79-92.

# Chapter 7

## FRAMEWORK IMPLEMENTATION

In this chapter, we detail the prototype implementation of the framework. We start by a brief description of the platform existed at Telecom SudParis Lab. Next, we move for presenting our new platforms' extensions that are needed to provide more privacy, security and trust. Moreover, we present the new platform re-design, we present the implementation phases that were done and we conclude with an example scenario.

### 7.1 Scope Of The Prototype

As a proof of concept, a prototype for the architecture was implemented. The following sections describe the implementation phases of this prototype. The focus of the prototype lies on the implementation and evaluation of our context-aware authentication-based process including the access layer and access control layer within the proper module in the available platform. The prototype focuses on the notion of the context-aware authentication process explained in previous chapter. The scenarios to be handled by the prototype are based on user-initiated interaction, and the focus is on the interaction with the platform. As a consequence, interaction between the user and the context-aware service is not part of the prototype. The provisioning of a service to these users was considered achieved and is not part of the prototype. For the prototype, the user agents behavior was limited to evaluating privacy control, user's credentials and current contextual information, comparing them to the existed services policies and the associated context-aware preferences in the platform.

## 7.2 Preliminary

The main goal of the prototype is to serve as a proof for our framework concept. For this purpose, several characteristics were considered most important: -(i) Evaluation of context for user class -(ii) Evaluation of the access layer class, and finally -(iii) Evaluation of the access control layer class. With these priorities in mind, **Java** was chosen as the programming language for the prototype. During the last years, Java has been one of the technologies with a fastest growth. Cryptographic capabilities were first added to the Java SE platform and then extended to other platforms such as Java Card. Elliptic Curve Cryptography (*ECC*) represent one of the most interesting techniques for protecting sensitive information nowadays and it is represented in all major part of our work. According to [2], among the independent implementations developed outside the Java standardization bodies, *Bouncy Castle* and *IAIK* outstand above the rest. Both of them provide high quality implementations and can be used for ECC applications and other cryptographic deployments. As a drawback, and due to Java nature, we have developed and employed new classes and interfaces to the ECC Java-based package in order to use all needed cryptographic operations and procedures related to the functionality of our framework. As a programming environment, **Eclipse** was used. Eclipse is an open source community, whose projects are focused on building an open development platform comprised of extensible frameworks, tools and runtime for building, deploying and managing software across the lifecycle.

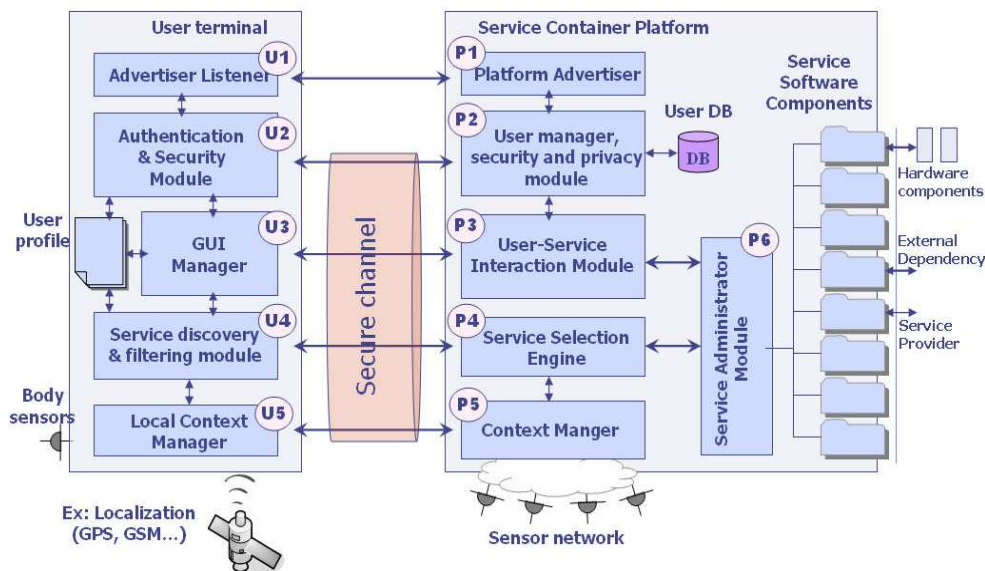
## 7.3 Platform Extension

In this section we will present the proposed platform at Telecom SudParis Lab. We will briefly describe the different components of the proposed architecture, the internal modules and the different interactions that may occur during the system running. Moreover, we will describe the new platforms' extensions that need to be integrated in order to enhance privacy, security, and trust. However, our new architecture provides both user-based and context-aware authentication. Finally we conclude this section by showing the final platform new re-design.

### 7.3.1 Previous Platform

The service provision architecture proposed at Telecom SudParis institute covers two main entities:-(1) The user through the software client used on his terminal.- (2) The environment providing a service container platform. Each entity contains internal modules. The following figure, (Figure 7.1), illustrates different components of the proposed architecture. This figure shows these internal modules and the different interactions that may occur during the system running.

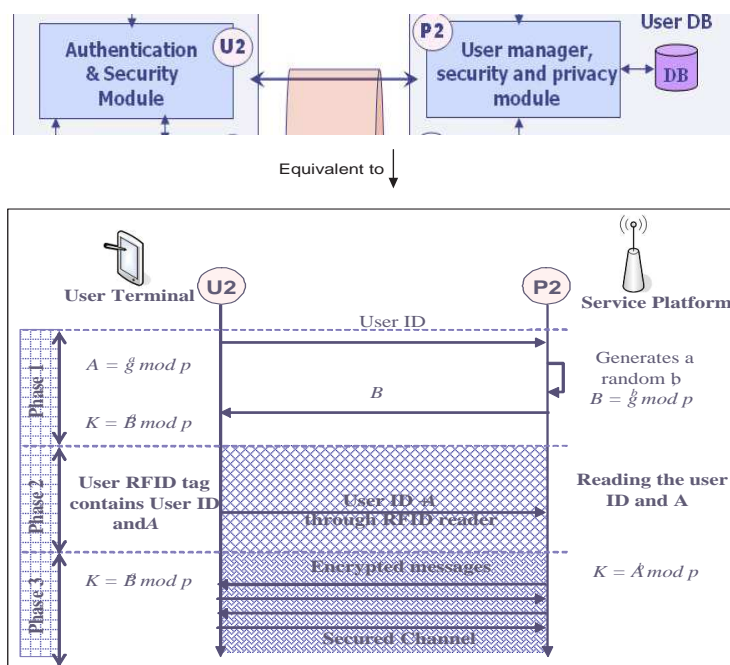
Figure 7.1: Proposed Service Provision Architecture



The two main entities are highlighted. On the left side of the figure, the user is represented by the user terminal software client. On the right side, the environment is represented by the service container. Each one of them is composed of several modules. Almost, each module in one side has its equivalent or its interlocutor in the other side aiming at a full interaction between both entities. User terminal modules are abbreviated to U1, U2, U3, U4 and U5. Service platform modules are abbreviated to P1, P2, P3, P4, P5 and P6. These modules from each side are to be briefly detailed as follow:-(i) P1 corresponds to the Platform Advertiser module whereas U1 is its equivalent module in the user side; Advertiser Listener. The major role of the interaction between those two modules is to establish

the first link between the user and the service platform. According to authors, security aspects are not considered in this first phase because it concerns all potential users connected to the network. -(ii) After discovering the service platform, the user might be assured of the integrity and the confidentiality of data exchanged with the service platform. The role of U2 and P2 is to establish such secured channel. Different security aspects could realized through secret keys which has to be generated online between U2 and P2 just after advertisement process. According to authors, the most suitable key management method was Diffie-Hellman. Figure 7.2 represents the U2 and P2 processes. -(iii) P3 and U3 modules

Figure 7.2: U2 and P2 Process



are responsible of the presentation layer of provided assisted through the selected services. After receiving the list of selected services from U4 module, U3 contacts P3 to perform the presentation of those services.-(iv) U4 and P4 are modules responsible of service discovery process. They work closely with previous modules (P5 and U5) in order to determine whose services are suitable for the user.-(v) P5 module deals with environment context extracted from the ambient sensor network. U5 deals with user context. It is about user information like his location (or information coming from local body sensors).

### 7.3.2 New Platform: Extensions and New Design

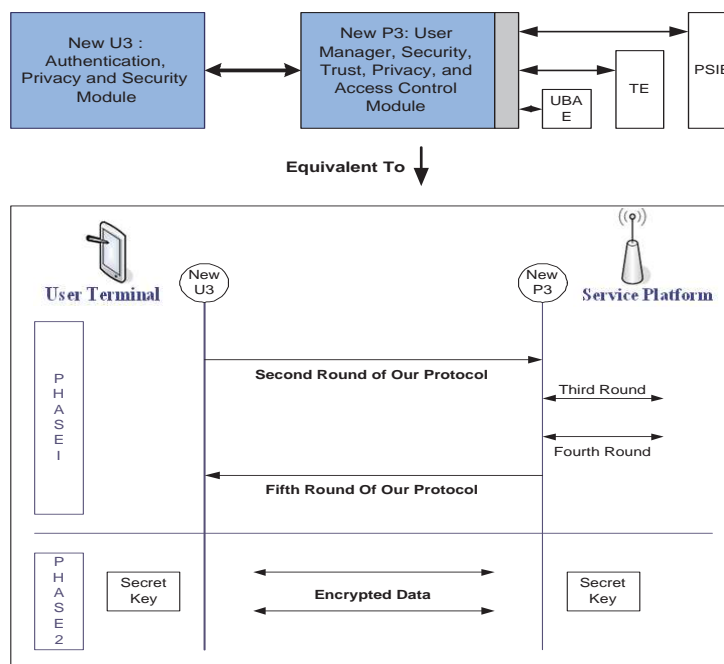
The main modules that need to be extended are: (U1, P1), (U2, P2), (U4, P4), (U5, P5), and finally (U6, P6) modules. Minor extension have been made to (U1, P1), (U5, P5) and (U6, P6) modules, whilst major extension where done to (U2, P2) and (U4, P4) modules. The module (U3, P3) could be used with its functionality. In the following we will describe the extension and modification that were done to each of the modules.

**(U'1, P'1) Module:** P'1 will correspond to the Platform Advertiser module. Different from P1, P'1 will publish all available context-based services to the user in process. We will assume that P'1 has the ability to publish services based on the available captured user's contextual information. In order that our assumption to work properly, we have extended the old (U1&P1) module to be able to communicate with the new U'4 and P'4 module. U'1 will be its equivalent module in the user side; Advertiser Listener. The major role of the interaction between those two modules is to establish the first link between the service platform and the user. According to authors, security aspects are not considered in this first phase because it concerns all potential users connected to the network and were provided in U2, P2 module. However, these points were proved not to be achieved with the old platform. However, after discovering the service platform using the new design, the user might be assured of the integrity and the confidentiality of data exchanged with the service platform by using U'3 and P'3 module.

**(U'2, P'2) Module:** According to our approach, U'2 and P'2 are the new modules responsible of contextual information based service discovery process. They work closely and in conjunction with other modules, (U'1 and P')(U'5 and P'5), in order to determine whose services are suitable for the user based on captured context attributes.

**(U'3, P'3) Module:** According to our approach, the role of U'3 and P'3 module is to provide user's authentication while security and privacy are preserved and to establish a secure channel through providing a robust key agreement process. Different security aspects could realized through secret keys which has to be generated online between U'3 and P'3 just after advertisement process. According to authors, the most suitable key management method was Diffie-Hellman. However, a new authentication and key agreement protocol

was implemented. This protocol is a part of our complete framework that has been integrated within the platform. Figure 7.3 shows the new extended module ( $U'3$ ,  $P'3$ ) as it will be integrated within the existed platform.

Figure 7.3:  $U'3$  and  $P'3$  Process

On the left side of the figure, the user is represented by the the new  $U'3$  user terminal client. On the right side, the  $P'3$  is represented by new processes containers that specify the access and the access control layers. Each one of them is composed of several modules. In the following, we will represent the developed extensions to these modules. Moreover, we will show how our integrated framework can improve the security and usability of the platform. We now start giving the set of algorithms to evaluate the new ( $U'3$ ,  $P'3$ ) module. When typesetting texts with many algorithms, many writers have questions regarding how they will present the algorithm to his readers. Showing them in the real well-established language (as our case, like C++ and JAVA) may be a good choice, but often, these languages require too many synthetic details to be written and such details distract the reader from the essence points of the algorithm. For this reason, we have agreed to present our algorithms in

a language that is similar to well-known languages (like C++, Java) but with more flexibility, in what is known as pseudo-code. The following subsections define the set of extension needed to configure the new  $(U'3, P'3)$  module for a context-aware authentication-based access.

**User Agent:** In this section, we formally define the user' algorithms or classes denoted by USER-ACCESS algorithms (Algorithm 1 and 2) respectively and presented as pseudo code algorithms. These formal algorithms rely on the components defined as follow:  $Adv$  is the advertised message sent by the  $SP$  to start communication, a role  $r$ , a service  $srv$  with a permission  $p$ , and finally the user private key  $S_i$ . The user's algorithms work as follows:

---

**Algorithm 1 : USER-ACCESS (Phase[I]: Start of Second Round)**

---

**INPUT:**  $S_i, r, p, srv$

**OUTPUT:**  $T_{ED}, \langle U, V \rangle, f$

Get Attribute set:  $C = getAttributeSet$ , which returns the set  $C$  of context

Generates: A fresh random number  $f \in_R Z_2^t$ , A fresh random number  $a \in Z_q^*$

Computes:  $T_{ED}$ , and  $Q$

Computes: Signature over  $Q$  denoted by  $SigQ = \langle U, V \rangle$

**RETURN**  $T_{ED}, C[], f, \langle U, V \rangle$

---



---

**Algorithm 2 : USER-ACCESS (Phase[I]: End of Fifth Round)**

---

**INPUT:**  $y_i, T_{SP}$

**OUTPUT:** Decision,  $K_{ED/SP}$

Computes:  $SUM = (\sum(y_i \times P_i) + (f \times Z))$

Checks:  $Access = GetDecision(SUM, X)$

**if** ( $Access == False$ ) **then**

$Decision = Access - Denied$

**RETURN** Decision

**else if** ( $Access == True$ ) **then**

$Decision = Access - Granted$

Calculates:  $K_{ED/SP}$

**RETURN**  $K_{ED/SP}$

**end if**

---

**Access Layer Agent:** In this subsection, we formally define the extensions (listed as

algorithms) that were integrated in the platform (in previous (U2&P2) module). These extensions include the integration of a PSI engine (Algorithm 3), of an IBE protocol (Algorithm 4), and finally of a trust engine (Algorithm 5). The formal PSI algorithm rely on several components defined as follow:  $A$  is a contextual information set retrieved from the service access request issued by the user, and  $S_{srvi}$  is set of attributes for the pair  $(r, srv)$ . The values of these attributes are specified by the system security officer (SSO). Based on the service access request, the system determines the applicable authentication access policy for the requested service. This policy will be based on a set of constraints on the role and service name, and evaluated in conjunction with the available presented contextual information. One main achievement is to provide both user-based and context-aware authentication.

The PSI-ALGORITHM process is defined as follows. This algorithm is implemented as a class that provides one method: `getContext(String userId, String contextType)`. This function takes a user name and a context type as parameter.

---

**Algorithm 3** : ACCESS LAYER, PSI Engine (Phase[I]: Round 3 & 4)

---

**INPUT:** Set  $A$  Where  $A = \{ca_1, ca_2, \dots, ca_j\}$ , Set  $S_{srvi}$  Where  $S_{srvi} = \{S_{srvi1}, S_{srvi2}, \dots, S_{srvi j}\}$   
**OUTPUT:**  $d_i, PSI - Decision$  where  $PSI - Decision = \{ALLOWED, N - ALLOWED\}$

GET  $A$ , GET  $S_{srvi}$

$PSI - P = \langle A, S_{srvi} \rangle$ ; Initialize PSI Protocol  $PSI - P$  over  $A$  and  $S_{srvi}$

Calculates  $d_i = \{A \cap S_{srvi}\}$

GET  $PSI - Decision$

**if** ( $PSI - Decision = ALLOWED$ ) **then**

    CALL IBE ALGORITHM (4)

    CALL TRUST ALGORITHM

**else if** ( $PSI - Decision = N - ALLOWED$ ) **then**

    CALL TRUST ALGORITHM

**end if**

**RETURN**  $d_i, PSI - Decision$

---

**Access Control Agent:** In this section, we formally define the access control's algorithm denoted by AC-ALGORITHM (Algorithm 7). The formal algorithm relies on the

---

**Algorithm 4 :** ACCESS LAYER, IBE Protocol (Phase[I]: Round 3 & 4)

---

**INPUT:** The set  $d_i$ , where  $d_i = A \cap S_{srvi}$  denotes the shared set of context,  $G$ .

**OUTPUT:**  $T_{SP}$ .

Calculates  $T_{SP} = (\sum d_i).G$

**RETURN**  $T_{SP}$ .

---



---

**Algorithm 5 :** ACCESS LAYER, Trust Engine (Phase[I]: Round 4)

---

**INPUT:**  $T_P, T_{ev}, T_p$ , A User Assignment Threshold  $UAT$ .

**OUTPUT:**  $UTL, RRT, UA$ , Trust Decision  $TD$ .

Calculates  $UT$  or  $TL$

Calculates  $RT$

Calculates  $UA = Comp\langle TL, RT \rangle$

**if** ( $UA \geq UAT$ ) **then**

$TD = AccessToBeGranted$

**else if** ( $UA < UAT$ ) **then**

$TD = AccessToBeDenied$

**end if**

**RETURN**  $UTL, RRT, UA$ , and  $TD$

---



---

**Algorithm 6 :** ACCESS SERVER, UBA Engine (Phase[I]: Round 3)

---

**INPUT:** The signature pair  $\langle U, V \rangle, f$ .

**OUTPUT:**  $\langle s_1, s_2 \rangle$  associated the user requesting access, Return *Decision* where  $Decision = \{Verified \text{ or } UnVerified\}$

Get  $\langle U, V \rangle$

$Decision = Verify\langle U, V \rangle$

**if** ( $Decision == Verified$ ) **then**

GET  $\langle s_1, s_2 \rangle$

**RETURN**  $Decision, \langle s_1, s_2 \rangle$

**else if** ( $Decision == Unverified$ ) **then**

**RETURN**  $Decision$

**end if**

---

components defined as follow:  $C$  is a context array. The USER-ALGORITHM works as follows.

---

**Algorithm 7** : ACCESS LAYER, AC Engine (Phase[I]: Round 4)

---

**INPUT:**  $role, srvi, CA$  Where  $CA$  is the user's request context Array

**OUTPUT:**  $Decision D$ , where  $D \in \{YES, NO, PENDING, N/A\}$

```

GET  $CL = GETCLAUSES(role, srvi)$ 
GET  $A = GETATTRIBUTES(role, srvi)$ 

for  $j = 1$  To Length( $CL$ ) do
  SET  $Clause = CL[j]$ 
   $ACCESS = GETDECISION(CLAUSE, A, CA)$ 
  if ( $ACCESS == FALSE$ ) then
    RETURN  $Decision = NO$ 
  end if
end for
if ( $ACCESS == TRUE$ ) then
  RETURN  $Decision = YES$ 
else
  RETURN  $Decision = PENDING$  or  $N/A$ 
end if

RETURN

```

---

**( $U'4, P'4$ ) Module:** According to our approach,  $U'4$  and  $P'4$  are the new modules responsible of the presentation layer of provided assisted through the selected services. After receiving the list of selected services from  $U'3$  module,  $U'4$  contacts  $P'4$  to perform the presentation of those services.  $P'4$  interacts with the service administrator  $P'6$  to extract the presentation layer of each service.  $U'4$  is responsible of the adaptation and the personalization of the display of services on the user terminal. That's why it is closely interacting with additional element like user and device profile

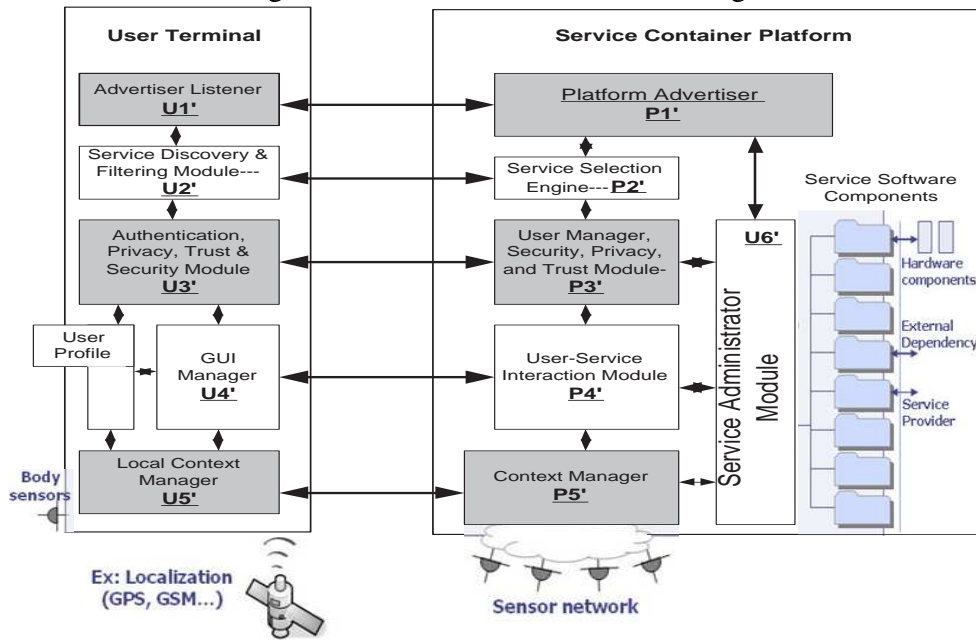
**( $U'5, P'5$ ) Module:**  $P'5$  module deals with environment context extracted from the ambient sensor network. Those sensors should be not intrusive enough to not infer the user life style and simple enough to be easily deployed.  $U'5$  deals with user context. It is about user information like his location (or information coming from local body sensors). When need be,  $U'5$  sends information about user through the secured channel established during

interaction between  $U'3$  and  $P'3$ .

$(U'6, P'6)$  **Module:** All service administration and management are done through this module. We distinguish two kinds of interaction with this module: 1- System-System Interaction: this module interacts with other modules or the service platform. Both  $P'2$  and  $P'4$  query this module in order to interact with available services. Indirect interaction with  $U'4$  is done through  $P'4$  in order to perform User-Service-Interaction. 2- Administrator-System Interaction: this is another human machine interaction involving the human in the administration process. In fact, services have to be installed at the beginning of the life cycle, updated when they are modified and also removed if need be. That includes a special administrator interface.

Finally, the new re-design is well illustrated in figure 7.4. This figure illustrates the new different components of the proposed architecture. It shows these internal modules with the new order and the different interactions that may occur during the process running.

Figure 7.4: New Architecture Re-Design



## 7.4 Platform: Implementation And Evaluation

The following section details the implementation phases that were done and the evaluation steps. In our work we consider Elliptic Curve Cryptography (ECC) because of the high level of security it provides with small key sizes. ECC is ideal for use on constrained environments such as pagers, personal digital assistants, smart phone, PDA, etc. However, the platform does not include a Java based ECC package allowing accessing all the power that *ECC* can provide. Therefore, the first problem definition was to well select the proper ECC package and to deploy it. As problem solution, several research works have been done regarding the available libraries and cryptography modules that can be used for the development of cryptosystems. Finally, the main conclusion derived, according to [2], where *Bouncy Castle* and *IAIK* outstand above all the independent implementations packages developed outside the Java standardization bodies. As second problem definition, was the ability to define, develop, and employ new classes and interfaces to the integrated ECC package in order to fulfill all needed cryptography operations and procedures related to our framework process (i.e, bilinear pairing, etc.).

### 7.4.1 Implementation Setting

The elliptic curve operations defined above real numbers are slow and inaccurate. To make cryptographic operations on elliptic curve faster and more efficient, the curve cryptography is defined over finite fields. The equation of the elliptic curve on a prime field  $F_p$  is  $y^2(\text{mod}p) = x^3 + ax + b(\text{mod}p)$ , where  $4a^3 + 27b^2(\text{mod}p)$  not equal to 0. Here the elements of the finite field are integers between 0 and  $p - 1$ . All the operations such as addition, subtraction, division, multiplication involves integers between 0 and  $p - 1$ . This is modular arithmetic and is defined in chapter 2. The prime number  $p$  is chosen such that there is finitely large number of points on the elliptic curve to make the cryptosystem secure. SEC specifies curves with  $p$  ranging between 112 – 521 bits [3]. The domain parameters and the key size should be chosen so as to provide sufficient cryptographic security [1, 4]. Apart from the curve parameters  $a$  and  $b$ , there are other parameters that must be agreed by both parties involved in secured and trusted communication using ECC. These are domain parameters. There are several standard domain parameters defined by [3, 5, 6].

The domain parameters for Elliptic curve over  $F_p$  are  $p$ ,  $a$ ,  $b$ ,  $G$ ,  $n$  and  $h$ .  $p$  is the prime number defined for finite field  $F_p$ .  $a$  and  $b$  are the parameters defining the curve  $y^2(\text{mod}p) = x^3 + ax + b(\text{mod}p)$ .  $G$  is the generator point  $(x_G, y_G)$ , a point on the elliptic curve chosen for cryptographic operations.  $n$  is the order of the elliptic curve. The scalar for point multiplication is chosen as a number between 0 and  $(n - 1)$ .  $h$  is the cofactor where  $h = \#E(F_p)/n$  and  $|E(F_p)|$  is the number of points on an elliptic curve. In addition, the Bouncy Castle Crypto packages [7] are a Java implementation of cryptographic algorithms including `org.bouncycastle.crypto`, `org.bouncycastle.math.ec` (Support for EC) and they were developed by the Legion of the Bouncy Castle. The *Bouncy Castle*'s latest version supports different EC-based signatures and Key agreement protocols including *ECDSA*, *ECDH*, standard Diffie-Hellman key agreement protocol, etc. As a drawback, this version does not include new ID-based protocols neither bilinear pairing variant. As a solution, we have developed these new classes and cryptographic tools to fulfill the needed requirements. The *Bouncy Castle* package, namely `org.bouncycastle.math.ec`, consists of the following classes:

**ECConstants Class:** Which provides the numbers 0 to 4 as BigIntegers.

**ECCurve Class:** Which represents the base for an elliptic curve in the Weierstrass normal form.

**ECFieldElements Class:** Which represents an element in the Galois field that is used.

**ECPoint Class:** Which represents the base class for representing points on the elliptic curve and implements the arithmetic of this curve.

All these classes are implemented based on  $F_p$  that represents elliptic curves defined over a prime field  $F_p$ . However, we urge our reader to visit <http://bouncycastle.org> for more information.

## 7.4.2 Implementation and Testing Processes

In this section, we will present the JAVA based *Implementation* and *Testing* phases that have been integrated and tested within the platform. We will start by giving a brief overview regarding the ECC package installation process then we move to describe the user and access layer agents both in static and dynamic processes. For our different set

of experiments, we generated a series of access request using different set of rules, roles, policies. The figure below (Figure 7.5) represent the different ECC-based classes that were installed and integrated within the Java-based Eclipse platform.

Figure 7.5: ECC Packages Installation Classes

```

import org.bouncycastle.math.ec.ECCurve;
import org.bouncycastle.math.ec.ECPoint;

import java.math.*;
import java.util.Hashtable;
import java.util.Random;
import fr.dgac.ivy.*;
import java.io.*;
//import org.osgi.framework.ServiceReference;
import org.w3c.dom.*;
import org.xml.sax.*;
import javax.xml.parsers.*;
import javax.xml.transform.*;
import javax.xml.transform.dom.*;
import javax.xml.transform.stream.*;

private ECCurve.Fp curve;
private ECPoint p1;
private ECPoint p2;
private BigInteger s1 ;
private BigInteger s2 ;
private BigInteger Zx ;
private BigInteger Zy ;
private String path;
private int h;
private String q = "104729";
private String n = "78517";
private String a = "6074";
private String b = "1445";
private static ECPoint G;
private static ECPoint S;
private static ECPoint Z;

```

The first two requests (*Static Phase*) involved a policy that granted or denied accesses regardless of environmental and contextual information. All other requests (*Dynamic Phase*) involved access check that made use of more complex policies and modules including trust engine, contextual information rules, etc. However the following two figures represents the *ASM\_AccessServer* (Figure 7.6) and the *ASM\_SecClient* (Figure 7.7) Java packages respectively that have been integrate within the *Eclipse* platform both in static and dynamic phases.

**Static Phase:** In this section, the protocol described in chapter 4 has been integrated in the platform. This protocol represent a static phase as no contextual information are integrated neither trust or privacy modules. This phase represent an improvement over the already built-in security framework (old framework) by the mean of providing integrity, confidentiality, robust shared key, access control. Two scenarios have been conducted.

The first scenario (Figure 7.8) represents a legitimate user attempting to access resources. In our proposed protocol, the user is authenticated whenever  $D$  is equal to  $X$ . As  $D$  and  $X$  are points on the elliptic curve, therefore  $D$  and  $X$  are represented by  $(S_x, S_y)$  and by  $(X_x, X_y)$  respectively. Finally the user is authenticated whenever  $S_x$  is equal to  $X_x$  and  $S_y$  is equal to  $X_y$ . From Figure 7.8, it is well noticed that  $D$  and  $X$  are equal and

Figure 7.6: ASM AccessServer Java Implementation

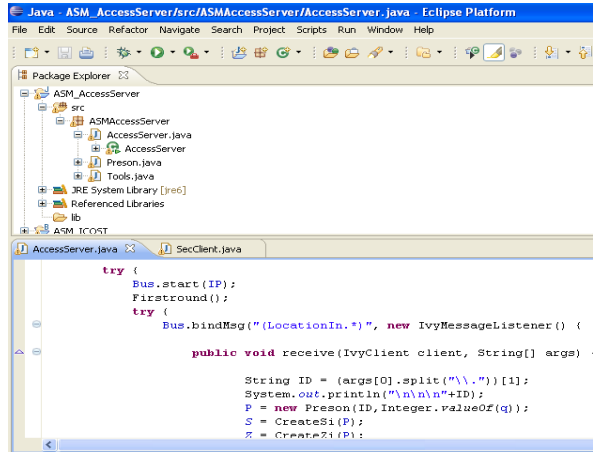
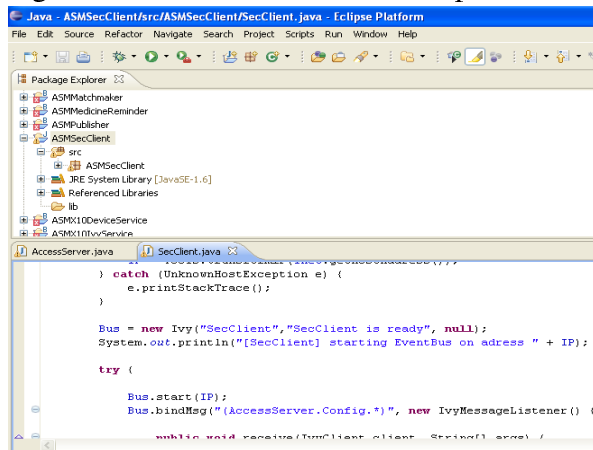


Figure 7.7: ASMSecClient Java Implementation



therefore the user is authenticated.

The second scenario ((Figure 7.9)) represents an unauthorized user attempting to access resources. It is well noticed that the values of  $D$  are not equal to the values of  $X$  and therefore the user is not authenticated.

**Dynamic Phase:** The framework presented in Chapter 6 has been integrated in the Eclipse platform. This framework contains the protocol of Chapter 6 with the addition of the contextual information, trust and privacy modules. In our implementation, policies are defined through the eXtensible Markup Language (XML). XML is used to specify access policies, roles definitions, etc. Figure 7.10 shows an XML policies prototype.

Figure 7.8: User Access Granted

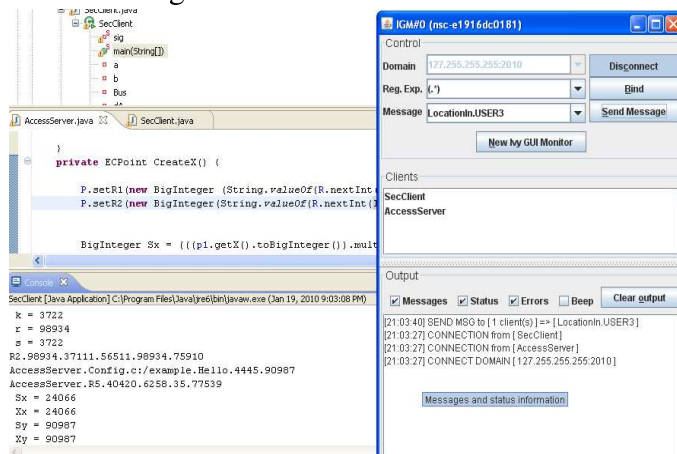
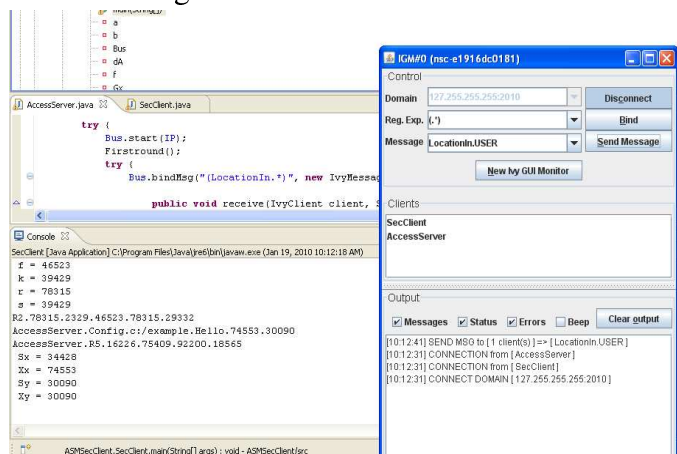


Figure 7.9: User Access Denied



XML provides an efficient structure for storing the policy that is generated and enforced by our security services. The experiments were conducted on a cluster of workstations using dual-2.20 GHZ Intel processors, running windows XP Professional. Figure 7.11 represents the steps to be done by the access server. Depending on the user based contextual information, the access server will select the corresponding context based services and deliver them to the user. In our following example, these corresponding services are services *S0*, *S1* and *S3*.

Figure 7.12 illustrates a user with the option of selecting a service from the available context-based services. In our case, the user has the ability to select a service form these

Figure 7.10: Policy Specification in XML

```

<?xml version="1.0"?>
<services>
  <service name="S0">
    <trust value="0.7"/>
    <time inf="6" sup="15"/>
    <age inf="100" sup="1000"/>
  </service>
  <service name="S1">
    <trust value="0.7"/>
    <time inf="8" sup="13"/>
    <age inf="18" sup="1000"/>
  </service>
  <service name="S2">
    <trust value="0.5"/>
    <time inf="0" sup="24"/>
    <age inf="100" sup="1000"/>
  </service>
  <service name="S3">
    <trust value="0.4"/>
    <time inf="0" sup="13"/>
    <age inf="20" sup="1000"/>
  </service>
  <service name="S4">
    <trust value="0.1"/>
    <time inf="0" sup="24"/>
    <age inf="0" sup="1000"/>
  </service>
</services>
-----
-----

```

three services. As illustration, the user select the service namely *S3*.

Figure 7.13 and 7.14 show that the user has been granted access the service. As demonstration, the user's contextual information fulfill the service's access requirements and therefore the user will be allowed to access the server.

Another scenario has been illustrated where the access request has been denied. Figures 7.15 and 7.16 represent the user's service selection and the access decision respectively. The user select the service namely *S0*. In this scenario, the user's contextual information does not fulfill the service access requirements and therefore the user will not be allowed to access the service.

## 7.5 Implementation Outcomes

It is our belief that the following main goals have been accomplished after implementing the new framework. The outcomes after the implementation are summarized as follow:

**Provide Privacy:** Our new implementation design provide users with a mean to control the reveal of their personal and contextual information. This feature was not implemented in the old implementation design. However, by introducing and integrating the privacy control layer, we have enabled users to express their preferences in a set of preference-rules (called a ruleset), which can then be used by their user agents to make automated or semi-automated decisions (using services privacy policies) regarding the reveal of personal

Figure 7.11: Resources Access In Dynamic Environments

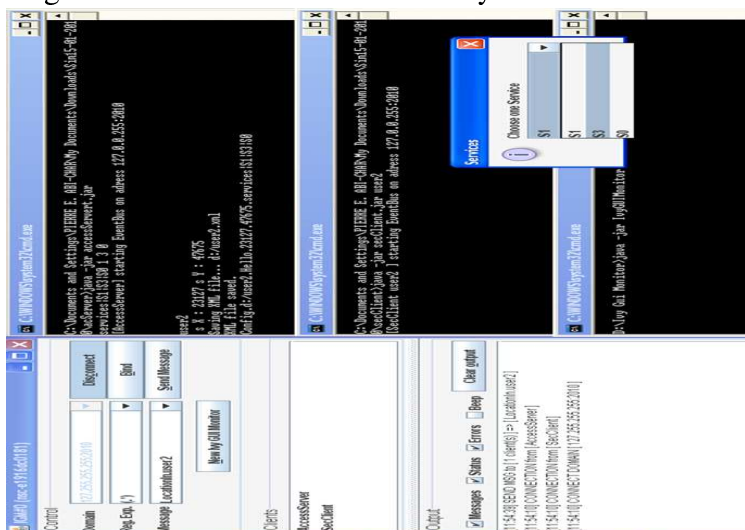
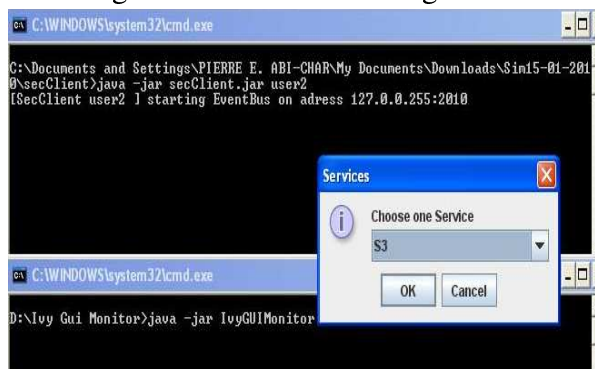


Figure 7.12: User Selecting Service



and contextual information.

**Provide User & Context-Based Authentication:** Our new implementation design provide users with both user and contextual information based authentication. This process is very interested in context-aware service. The old design was just done to provide user based authentication by using the Diffie Hellman protocol with an RFID tag. However, the new design has been achieved by using the *Private Set Intersection* technique.

**Provide Trust:** Our new implementation design provide users with a trust-based context-aware authentication process. The old implementation design does not provide trust-based

Figure 7.13: User Access Granted-1-

```

C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\PIERRE E. ABI-CHAR\My Documents\Downloads\Sin5-01-201
@AccessServer>java -jar accessServer.jar
services!S1!S3!S0 1 3 0
[AccessServer] starting EventBus on adress 127.0.0.255:2010

user2
s X : 23127 s Y : 47675
Saving XML file... d:/user2.xml
XML file saved.
Config.d:/user2.Hello.23127.47675.services!S1!S3!S0

28512
8074
1445
1
928
568
47
527
54974
2813
741
1012
8974
1445
50929
k = 46431
e = 58803
s = 46431
R2.58803.66235.50929.58803.36591.user2.Sparam!S3;age=25;time=8;trust=0.5
AccessServer.R5.36762.100514.78232.89012.user2.ACCESS_GRANTED
X = 51272
Y = 23127
Z = 47675
XY = 47675

```

Figure 7.14: User Access Granted-2-

```

C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\PIERRE E. ABI-CHAR\My Documents\Downloads\Sin5-01-201
@AccessServer>java -jar accessServer.jar
services!S1!S3!S0 1 3 0
[AccessServer] starting EventBus on adress 127.0.0.255:2010

user2
s X : 23127 s Y : 47675
Saving XML file... d:/user2.xml
XML file saved.
Config.d:/user2.Hello.23127.47675.services!S1!S3!S0

SecClient.R2.58803.66235.50929.58803.36591.user2.Sparam!S3;age=25;time=8;trust=0.5
Sparam!S3;age=25;time=8;trust=0.5
100 < 25 < 1000
trust =true OK = true chek = true
R5.36762.100514.78232.89012.user2.ACCESS_GRANTED

```

access and neither a dynamic decision-making process. However, our new proposed design provide authentication and access control decisions based on trust measures. The new process has been achieved by using fuzzy logic operations and rules. Using the fuzzy logic concept, we can define and form a formal decision-making process to calculate user trustworthiness and role's required worthiness parameters.

**Derive Robust Shared Key:** Our new implementation design provide users with the ability to derive a very robust shared key. The derivation of the shared key in the old design has been achieved using the Diffie Hellman process. However their process does not provide a robust shared key derivation that can resist key attack. Instead, our new design provides users the ability to derive more robust shared key. This new process has been achieved by using new elliptic curve techniques including the *Bilinear Pairing* protocols.

Figure 7.15: User Selection Service

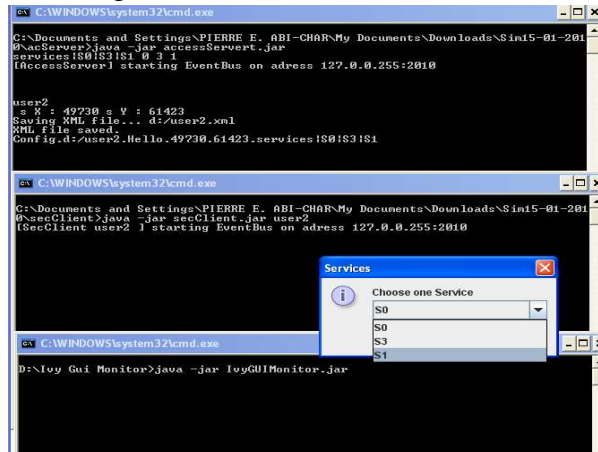
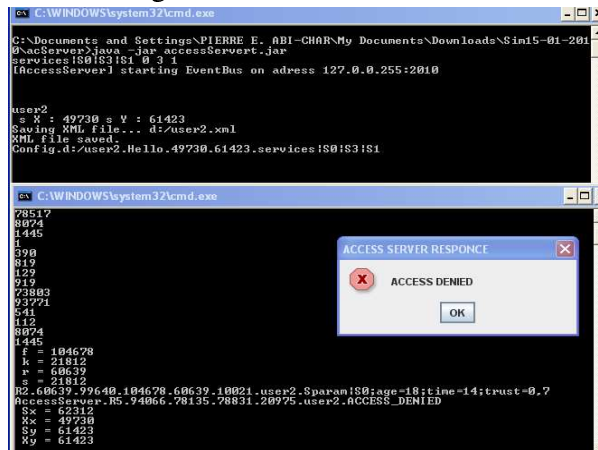


Figure 7.16: Access Decision



As a summary, our new implementation design provide users with a dynamic and flexible context-aware secure authentication framework where integrity, confidentiality, trust and privacy are integrated and provided. This approach has been achieved by the integration of these new security engines listed in *Chapter 6*.

## 7.6 Conclusion

In conclusion, this chapter present an implementation for our context-aware based authentication framework. We implement several extensions for the involved entities in order

to improve security (i.e, efficient context-aware authentication) and usability. The prototype we implemented indicated that the architecture seems viable. To further evaluate our architecture, however, more implementation and evaluation are necessary. However, the architecture description and implementation with the study case testing and evaluation are to be presented as an international scientific journal for the ACM Transactions on Information and System Security (TISSEC), available at <http://tissec.acm.org/>. Reviewing, editorial and notification are all in a turn around time of 6 months.

## Bibliography

- [1] H. Orman, and P. Hoffman, *Determining Strengths For Public Keys Used For Exchanging Symmetric Keys* , In Proceeding of the Network Working Group, Request For Comments 3766, BCP 86, 2004.
- [2] M. Gayoso, L. Encinas, and C. Avila, *Elliptic Curve Cryptography: Java Platform Implementations*, In Proceeding of the International Conference on Information Technologies (InfoTech-2009), vol. 1, 2009.
- [3] SEC Certicom, *SEC 2: Recommended Elliptic Curve Domain Parameters*, In Proceeding of Standards for Efficient Cryptography, Version 1.0, September 2000, Available at [http://www.secg.org/download/aid-386/sec2\\_final.pdf](http://www.secg.org/download/aid-386/sec2_final.pdf)
- [4] S.B. Wilson et al. *Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax*, In Proceeding of the Network Working Group, Request For Comments 3278, 2002.
- [5] S.B. Wilson et al. *Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)*, In Proceeding of the Network Working Group, Request For Comments 4492, 2006.
- [6] L.Zhu et al. *Elliptic Curve Cryptography (ECC) Support for Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)*, In Proceeding of the Network Working Group, Request For Comments 5349, 2008.
- [7] Bouncy Castle, *Bouncy Castle Crypto Packages*, Available at <http://www.ucsc.cmb.ac.lk/People/tnk/java/BCDocs/overview-summary.html>

## Chapter 8

# CONCLUSIONS AND FUTURE WORK

This dissertation has described a context-aware based authentication framework that is appropriate for pervasive computing environments and context-aware applications. In this final chapter, we summarize our research contribution and briefly describe some areas that merit future research.

### 8.1 Research Summary

The dynamic nature of mobile computing, pervasive computing environments, and especially contextual information has motivated the need for a secure and robust security framework that can operate effectively in such environment. An examination of existing security solutions, including a variety of authentication and access control schemes, has revealed that current approaches can not provide security protection for both context aware pervasive computing users and applications. Most notably, current authentication and access control can not meet the demands of a context-aware application because they fail to provide any extensive use of contextual information. To meet the need for a robust mobile computing, we firstly have developed several protocols for mobile computing using advanced cryptography techniques and later concluding by combining authentication and access control to eliminate the case for identifying user's ID in a separate process. Moreover, to meet

the needs for the context-aware computing environments, we have developed a trust based context-aware authentication framework that integrates context-awareness with automated reasoning to perform authentication and access control and while privacy is controlled using a privacy control layer. This framework is an extension to the work accomplished for the mobile computing. Moreover, we have developed this framework to be able to adapt with trust management and where privacy is always preserved.

Finally we have discussed the design and an implementation of our context-aware security architecture. Moreover, an implementation has been also studied. We have implemented a prototype for our security architecture and demonstrated its effectiveness. The implementation phase has been developed using the platform at Telecom SudParis (Ex. INT), Evry, Paris.

## **8.2 Summary of Contributions**

In this survey, we have outlined challenges facing developers of UbiComp applications with regard toward privacy, security and trust. The main contribution for this chapter is the notion of security, privacy, and trust enhancing services within context-aware environments. Privacy and trust adaptively provide protection for users as they enter UbiComp environments which allow for continued transparent use of services without compromising neither the users privacy nor the services ability to provide services. In pervasive computing environments, the deployment for a robust and dynamic security framework should be conditioned by privacy and trust needs so users can be confident by using available services within the environment.

## **8.3 Future Work**

The study of security in pervasive computing environments has a large scope and we obviously have not addressed all of the relevant issues. We will now discuss some issues that need further investigation and studies in order to be deployed.

We need to develop a methodology for investigating trust modeling (i.e. trust management and trust negotiations), trust value establishment, trust propagation, trust synthesizing, etc. By incorporating all these trust strategies into a formal model, this trust model can adapt to the different application scenarios, environment changes, and trust evidence types. It further resolves the limitations of current existing trust models in handling different trust management requirements. These researches and studies should focus on integrating privacy policies and references that can be automatically retrieved and interpreted by users' agents, who accept or reject services according to user's stated preference policy. Beside trust, Privacy-based enhanced access control should be more explored and extended to support privacy preferences. These designs should be incorporated into distributed and dynamic environments. Moreover, any new methodology should be based on a dynamic hybrid model that should integrate risk analysis, risk management and reputation mechanisms. In addition, trust model based on users' roles, capabilities, behavior, and context factors, etc., should be further investigated to improve the hybrid model. Other primary goal scheme is to increase quality of privacy, (QoP), by giving users more time to react adequately to dangerous situations. The concept of Quality of Privacy (QoP) allows balancing the trade-off between the amount of privacy a user is willing to concede and the value of the services that can be provided by UbiComp applications. This concept should be explored in any new trust and risk-based framework in pervasive computing.

## 8.4 Conclusion

This dissertation has studied the challenges involved with providing authentication and access control within context-aware environments. We have presented the requirements for and an implementation of our architecture that provide trust and privacy for resources and users in these smart environments. Our architecture provides a robust attribute-based authentication model that can express complex security policies by integrating with a dynamic access control. We have studied our security models and have provided a prototype implementation experience to demonstrate the effectiveness of our approach.

# Appendix A

## Basic Fuzzy concepts and Definitions

A fuzzy set is any set that allows its members to have different grades of membership (membership function) in the interval  $[0,1]$ . Fuzzy set  $A$  on  $U$  is completely defined by its membership function  $A : U \rightarrow I$ , Where  $I$  denotes the unit interval  $[0,1]$ . A fuzzy set is usually represented by  $A = \sum A(x)/x$  where  $A(x)$  is a member of the set and  $x$  is its membership degree. Moreover, we denote by  $x \wedge y = \min\{x, y\}$ , and  $x \vee y = \max\{x, y\}$ . Maximizing set of  $A$  and  $B$  is the fuzzy set  $M$  that consists all supports from  $A$  and  $B$ . Membership degree of each support equals the ratio of the support itself to the maximum support of  $A$  and  $B$ . For any two finite sets  $X$  and  $Y$ , we denote by  $R$  as the fuzzy relation from  $X$  into  $Y$ . The relation  $R$  is the fuzzy subset of the cartesian product  $X \times Y$  and it is represented by a matrix with all coefficients in the interval  $[0,1]$  and where for all  $x \in X$  and  $y \in Y$ ,  $R[x,y]$  represents the membership degree of  $(x, y)$  in  $R$ . We denote by  $[A\alpha R](y) = \sup_{x \in X} \{\min\{A(x), R(x, y)\}\}$ , where  $Y$  includes a finite set of value that can be assigned to  $B$ ,  $R$  is the fuzzy subset relation, and  $A$  is a set of attributes for a given user. The operator  $\alpha$  is called the *sup – min composition* of fuzzy set  $A$  and fuzzy relation  $R$ . We define the fuzzy implication operator  $\beta$  as:  $a\beta b = \sup\{c | 0 \leq c \leq 1, a \wedge c \leq b\}$ . We denote by  $GMD(A)$  the greatest membership degree of  $A$ .

## A.1 Trust and Trustworthiness

We define two parameters related to the concept of trust and trustworthiness. The first parameter is user trustworthiness ( $UT$ ) which means how much a user in system is reliable and how much we trust him. The second parameter is role's required trustworthiness ( $RT$ ) which determines the amount of trust is required by a user to play the role in system. After computing a user trustworthiness ( $UT$ ) and a role's required trustworthiness ( $RT$ ), user assignment ( $UA$ ) is performed based on the trust level of the user  $UT$  in comparison with the required trust level of the role  $RT$ .  $UT$  and  $RT$  are computed using users attributes and roles permissions respectively. We define a set of user  $U = u_1, u_2, \dots, u_n$  where  $u_i$  identifies a user of the system, and a set of roles  $R = r_1, r_2, \dots, r_n$  where  $r_j$  represents a role in the organization. The procedure will start as follow:

**Step 1:** Compute the user trustworthiness  $UT_i$  for user  $u_i$  and the role's required trustworthiness  $RT_j$  for role  $r_j$  where  $UT = A\alpha R$  and  $RT = Per\alpha R$ .  $A$  represents set of attributes for the user  $u_i$ ,  $Per$  represents the set of permission for user  $u_i$  with the role  $r_j$ . The fuzzy relation  $R$  is calculated by establishing a correspondence between different set of attributes and  $UT$ .  $R = \cap R_k$ , where  $R_k = A_k\beta UT_k$  and  $k = 1, 2, \dots, n$ .

**Step 2:** Compute  $(UT_i \wedge R)$  and  $(RT_j \wedge R)$ .

**Step 3:** In user assignment ( $UA$ ) relation, a user  $u_i$  can be assigned to role  $r_j$  if and only if  $GMD(UT_i \wedge R) \geq GMD(RT_j \wedge R)$ . In role activation, a user  $u_i$  can activate a role  $r_j$  if and only if  $GMD(UT_i \wedge R) \geq GMD(RT_j \wedge R)$ .

## A.2 Private Set Intersection

Any entity awaiting to be authenticated by  $U_a$  has to establish enough confidence in  $U_a$  and be able to present the required attributes. To keep a high level of security,  $U_a$  needs to keep those attributes private. For this purpose, we make use of the Private Set Interaction ( $PSI$ ), a cryptography tool that finds the commons between two set without revealing other attributes. Suppose that the user  $U_b$  is in possession of a set of context data denoted by  $B$ , where  $B = \{b_1, b_2, \dots, b_n\}$ . The user  $U_b$  wants to authenticate himself to the entity  $U_a$ . Upon his request,  $U_a$  initializes a  $PSI$  over the two sets  $A$  and  $B$ . Set  $A$  denotes

all attributes that  $U_a$  may use to set her rules for the authentication process.  $A$  is represented as  $A = \{a_1, a_2, \dots, a_l\}$ . The *PSI* protocol runs as follow: The User  $U_a$  chooses the secret-key parameters for a semantically-secure homomorphic encryption scheme, and publishes the public keys and corresponding parameters. Then, she calculates the coefficients of the polynomial  $P(Z) = \sum \alpha_i Z^i$  of degree  $l$  with roots of  $\{a_1, a_2, \dots, a_l\}$  and  $i = 0, 1, \dots, l$ . She encrypts each of the  $(l + 1)$  coefficients  $\alpha_0$  by the semantically-secure homomorphic encryption ( $\varepsilon$ ) and sends to  $U_b$  the resulting set of ciphertexts  $\psi$ , where  $\psi = \{\varepsilon(\alpha_0), \varepsilon(\alpha_1), \dots\}$ . Having received  $\psi$ , the user  $U_b$  uses the homomorphic properties of the encryption function to evaluate the polynomial  $P(\cdot)$  on each of his inputs (context data); that is for all  $j = 1, 2, \dots, m$ ,  $U_b$  computes  $\xi_j = \varepsilon(rP(b_j) + b_j)$  where  $r$  is a number selected at random each time. The user  $U_b$  then sends these values back to  $U_a$ , where she can decrypt the modified ciphertexts. It is simple to check that decryption of  $\xi_j$  returns  $b_j$  if  $b_j$  is a root of  $P(\cdot)$  and is in common  $b_j = a_j$  between two users, otherwise it returns a random number  $rP(b_j) + b_j$ . In this protocol, only  $U_a$  finds the common entries between the two sets, while no information about other entries of  $U_b$  are revealed. Therefore, it perfectly serves our purpose to privatize the authentication requirement of  $U_a$  defined in  $A$ , while it finds the matching attributes.

# Appendix B

## VITA

Pierre E. ABI-CHAR was born on September 9, 1974, in Halat, Lebanon. In 1998, Pierre E. ABI-CHAR received a Master Diploma in *Physics* from the faculty of Sciences at Lebanese University, Beirut Lebanon. He received an M.S. degree in Computer and Communications Engineering, *CCE*, from University of Balamand (UOB) in 2000. Moreover, he received a Diplome d'Etudes Approfondies, *DEA*, degree in Network Security in 2004 from University Pierre & Marie Curie-France in collaboration with France Telecom (ENST), Lebanese University Lebanese, Faculty of Engineering (Lebanon), and Institut National de la Recherche Scientifique (INRS-Telecommunication, Canada).

Mr. ABI-CHAR research efforts focusing on building secure, scalable, and dependable systems for pervasive computing environments led to his doctoral dissertation entitled: "Privacy-Preserving Authentication Architecture for Pervasive Computing Environments". Mr. ABI-CHAR received his Ph.D. from department of Reseaux Et Service De Telecommunication at Telecom Sud-Paris (Ex. Institut De Telecommunication) and from University Pierre & Marie Curie, *Paris-6*.

Mr. ABI-CHAR research interests span the fields of Applied Cryptography, Network and Computer Security, Applied Number Theory, Information and Coding Theory, Wireless Security, Context-Aware Security Technologies, Access Control, Privacy, Trust Management and Risk Assessment.