# USER-CENTRED AND CONTEXT-AWARE IDENTITY MANAGEMENT IN MOBILE AD-HOC NETWORKS

Abdullahi Arabo MEng, MBCS

A thesis submitted in partial fulfilment of the requirements of Liverpool John Moores University for the degree of Doctor in Philosophy

December 2011

Any maps, pages, tables, figures graphs,or photographs, missing from this digital copy, have been excluded at the request of the university.

## Synopsis

A thesis presented on the security issues of Identity Management in Mobile Ad-hoc Networks, considering the issues and utilisation of contextual information, user-centricity and user control in a new User-centred and Context-aware Identity Management (UCIM) Framework.

**ABSTRACT**

The emergent notion of ubiquitous computing makes it possible for mobile devices to communicate and provide services via networks connected in an ad-hoc manner. These have resulted in the proliferation of wireless technologies such as Mobile Ad-hoc Networks (MANets), which offer attractive solutions for services that need flexible setup as well as dynamic and low cost wireless connectivity. However, the growing trend outlined above also raises serious concerns over Identity Management (IM) due to a dramatic increase in identity theft. The problem is even greater in service-oriented architectures, where partial identities are sprinkled across many services and users have no control over such identities.

In this thesis, we review some issues of contextual computing, its implications and usage within pervasive environments. To tackle the above problems, it is essential to allow users to have control over their own identities in MANet environments. So far, the development of such identity control remains a significant challenge for the research community. The main focus of this thesis is on the area of identity management in MANets and emergency situations by using context-awareness and user-centricity together with its security issues and implications. Context-awareness allows us to make use of partial identities as a way of user identity protection and node identification. User-centricity is aimed at putting users in control of their partial identities, policies and rules for privacy protection. These principles help us to propose an innovative, easy-to-use identity management framework for MANets. The framework makes the flow of partial identities explicit; gives users control over such identities based on their respective situations and contexts, and creates a balance between convenience and privacy. The thesis presents our proposed framework, its development and lab results/evaluations, and outlines possible future work to improve the framework.

## TABLE OF CONTENTS


ABSTRACT ..... ii
TABLE OF CONTENTS ..... iii
ACKNOWLEDGMENTS ..... x
DEDICATION ..... xi
CHAPTER ONE ..... 2
INTRODUCTION ..... 2
1.1 BACKGROUND ..... 3
1.2 PROJECT AIMS AND OBJECTIVES ..... 4
1.3 PROJECT SCOPE ..... 5
1.4 NOVEL CONTRIBUTIONS OF THIS PROJECT ..... 6
1.5 PROJECT ACHIEVEMENTS ..... 8
1.5.1 BOOK CHAPTERS ..... 9
1.5.2 JOURNALS ..... 9
1.5.3 CONFERENCES ..... 10
1.5.4 CITATIONS ..... 11
1.6 METHODOLOGY ..... 13
1.7 THESIS ORGANISATION ..... 14
1.8 SUMMARY ..... 17
CHAPTER TWO ..... 18
THE DEVELOPMENT OF UBIQUITOUS COMPUTING ..... 18
2.1 A BRIEF HISTORY OF COMPUTER NETWORKS ..... 21
2.2 MOBILE AD-HOC NETWORKS (MANETS) ..... 23
2.3 AN OVERVIEW OF NETWORK SECURITY ..... 26
2.4 COMPUTER SECURITY ..... 32
2.5 CONCEPT OF UBIQUITOUS COMPUTING ..... 34
2.6 SUMMARY ..... 41
CHAPTER THREE ..... 42
NETWORK SECURITY AND IDENTITY MANAGEMENT ..... 42
3.1 IDENTITY MANAGEMENT (IM) ..... 42
3.2 USER CENTRICITY IN MANETS ..... 52
3.3 CONTEXT-AWARENESS IN MANETS ..... 53
3.4 Context-Aware Monitoring in System-of-Systems (SoS) ..... 59
3.5 SOCIAL IMPLICATIONS ..... 62
3.6 SUMMARY ..... 64
CHAPTER FOUR ..... 65
LIMITATIONS OF CURRENT IDENETITY MANAGEMENT SCHEMES ..... 65
4.1 USER CENTRICITY ..... 65
4.2 DYNAMIC POLICY ..... 66
4.3 CONTEXT-AWARENESS ..... 67
4.4 AIMS AND OBJECTIVES ..... 67
4.5 REQUIREMENTS ..... 69
4.5.1 REQUIREMENTS FOR ANONYMOUS COMMUNICATION ..... 69
4.5.2 REQUIREMENTS FOR IDENTITY MANAGEMENT ..... 71
4.5.3 USER REQUIREMENTS ..... 72
4.6 SUMMARY ..... 72
CHAPTER FIVE ..... 75

UCIM DESIGN FOR MANets ..... 75
5.1 PROPOSED FRAMEWORK ..... 75
5.1.1 CONTEXTUAL INFORMATION ..... 77
5.1.2 PERSONAL IDENTITY MANAGER ..... 78
5.1.3 PRIVACY MANAGER ..... 79
5.2 USER CENTRICITY ..... 79
5.2.1 SECURITY POLICIES ..... 80
5.2.2 CONTEXT BASED ACCESS CONTROL (CBAC) ..... 81
5.3 LIGHTWEIGHT ..... 84
5.3.1 HYBRID EUCLIDEAN METRIC ..... 84
5.3.2 CONTEXT ONTOLOGY ..... 88
5.3.3 INFORMATION GRANULARITY ..... 89
5.4 DYNAMIC POLICY SPECIFICATION ..... 90
5.4.1 PARTIAL IDENTITY ..... 95
5.5 CONTEXTRANK ..... 99
5.6 SUMMARY ..... 105
CHAPTER SIX ..... 106
UCIM IN MANETS IMPLEMENTATION ..... 106
6.1 SCENARIOS ..... 109
6.1.1 SOS COMPOSITION SCENARIO (CRISIS MANAGEMENT) ..... 110
6.1.1A THE BOUNDARY CHECKING PROBLEM ..... 115
6.1.2 IDENTITY MANAGEMENT IN SOS ..... 120
6.1.3A OUTDOOR EXPERIMENT USING MESH NETWORK TESTBED ..... 122
6.1.3 DATA MISHANDLING AND PROFILE BUILDING ..... 127
6.1.4 DEVICE CONFIGURATION ..... 132
6.1.5 USER INTERFACE (UI) CONFIGURATION ..... 134
6.2 MATTS AND COMPOSITION CLIENT APPLICATION ..... 135
6.3 CONTEXTRANK APPLICATION ..... 140
6.3.1 SCENARIO ..... 140
6.3.2 OVERVIEW ..... 142
6.4 DEMO WITH POLICY INTERFACE APPLICATION ..... 143
6.4.1 DYNAMIC POLICY SPECIFICATION INTERFACE ..... 143
6.4.2 POLICY XML FILE ..... 145
6.4.3 THE PROPERTY INTERFACE ..... 147
6.4.4 THE POLICY LOGIC ..... 150
6.4.5 THE POLICY CREATOR ..... 153
6.5 SECURITY MODULE FOR HOME GATEWAY FRAMEWORK ..... 156
6.6 SUMMARY ..... 160
CHAPTER SEVEN ..... 161
EVALUATION ..... 161
7.1 ANALYSIS OF PROPOSED FRAMEWORK ..... 161
7.2 SIMULATION AND LAP TEST RESULTS ..... 166
7.2.1 SIMULATION ..... 166
7.2.2 PORTABLE DEVICES ..... 168
7.2.3 DEVICES COMMUNICATION ..... 169
7.2.4 SECURITY ANALYSIS ..... 170
7.2.5 REFLECTION ON AIMS AND OBJECTIVES ..... 178
7.3 SUMMARY ..... 178

CHAPTER EIGHT ..... 179
CONCLUSION AND FUTURE WORK ..... 179
8.1 CONCLUSIONS ..... 179
8.2 FUTURE WORK ..... 181

LIST OF FIGURES

Figure 1 Development of Computing ..... 18
Figure 2 Mainframe to Ubiquitous Computers [14] ..... 19
Figure 3 ISO OSI 7-layer Reference Model ..... 28
Figure 4 Information Security Overview [27] ..... 31
Figure 5 Hyper Cycle ..... 34
Figure 6 Identity and Authentication [29] ..... 43
Figure 7 Strategy Map Adapted from [46] ..... 44
Figure 8 How Identity Information Is Stolen Adapted from [46] ..... 45
Figure 9 MANet Security Overview ..... 46
Figure 10 Positioning Technologies ..... 62
Figure 11 UCIM Framework ..... 76
Figure 12 Relevant contextual information ..... 78
Figure 13 Privacy Policy Protocol Sequence Diagram ..... 84
Figure 14 XML Schemas ..... 89
Figure 15 Information Granularity ..... 90
Figure 16 Identity and Partial Identities of Ababa ..... 96
Figure 17 Identity Attributes ..... 97
Figure 18 Profile Type and Context Information ..... 100
Figure 19 UCIM Implemented Modules ..... 107
Figure 20 SoS Composition Scenario ..... 110
Figure 21 Sending Properties of Node/Device ..... 112
Figure 22 Sending XML Files ..... 114
Figure 23 Establish Connection ..... 115
Figure 24 A composed system of trusted internal nodes, with no external connectivity. ..... 116
Figure 25 A composed systems with a connection to the outside. ..... 116
Figure 26 Sending Files between Organisations once connected to the Network using a PDA. ..... 118
Figure 27 Initial Network ..... 118
Figure 28 Analysed Network ..... 119
Figure 29 XML Property File ..... 124
Figure 30 AWDS Topology Viewer screen shot showing node connectivity ..... 124
Figure 31 Mobile ad-hoc network in a 100 square metre area using five netbooks ..... 125
Figure 32 MATTS Interface for IM in SoS ..... 125
Figure 33 Profile Building ..... 128
Figure 34 Data Mishandling ..... 128
Figure 35 Dynamic Policy Demo Tool - with the file deleted due to a policy violation ..... 130
Figure 36 Transfer File-Policy Fulfilled ..... 131
Figure 37 Delete File-Policy Violated ..... 131
Figure 38 Step-by-step Device Configuration ..... 133
Figure 39 Flow chart for configuration device ..... 134
Figure 40 Device Connection ..... 138
Figure 41 Send XML File ..... 140
Figure 42 ContextRank Sequence Diagram ..... 142
Figure 43 User Policy ..... 145
Figure 44 Example of different property types ..... 148

Figure 45 Property Interface Main Menu ..... 149
Figure 46 A Fully Populated Property Set ..... 149
Figure 47 Machine Generated Parser Code ..... 151
Figure 48 Hand Written Code ..... 152
Figure 49 PDA based policy interface (text free entry based on property file) ..... 154
Figure 50 Web Based Policy Creation Interface ..... 154
Figure 51 Web Based Policy Creation Interface for Sub-Functions Level 1 ..... 155
Figure 52 Configure UI ..... 156
Figure 53 Gateway Peer Overlay Network ..... 157
Figure 54 Policy Manager ..... 158
Figure 55 Device Policy List ..... 159
Figure 56 Mobile ad-hoc Network in a 100 Square metre area using five netbooks ..... 167
Figure 57 Property Interface for Defining Node Properties ..... 167
Figure 58 MATTS Test bed Interface ..... 168
Figure 59 UCIM deployed in Various Portable Devices ..... 169
Figure 60 Google+ Circles ..... 176
Figure 61 Google+ Profile and Privacy Settings ..... 177

LIST OF ACRONYMS

| | |
|---|---|
| AI | Artificial Intelligence |
| AWDS | Ad-hoc Wireless Distribution Service |
| CBAC | Context Based Access Control |
| CI | Context Information |
| CIA | Confidentiality, Integrity and Availability |
| CID | Corporate Identity |
| CoI | Community of Interest |
| CP | Context Provider |
| CPU | Central Processing Unit |
| CR | Context Requestor |
| CS | Context Server |
| CSMA/CA | Carrie Sense Multiple Access/Collision Avoidance |
| DSTL | Defence Science and Technology Laboratory |
| EMD | Emergency Medical Dispatch |
| ESA | Enhance Situation Awareness |
| ESN | Electronic Serial Numbers |
| FTP | File Transfer Protocol |
| GloMo | Global Mobile |
| GPS | Global Positioning System |
| GUI | Graphical User Interface |
| HCI | Human Computer Interaction |
| ICT | Information Communication Technology |
| IdPs | Identity Providers |
| ID | Identity |
| IETF | Internet Engineering Task Force |
| IM | Identity Management |
| IMEI | International Mobile Equipment Identity |
| IMMANets | Identity Management in Mobile Ad-hoc Networks |
| IMSI | International Mobile Subscriber Identity |
| IP | Internet Protocol |
| ISO | International Standards Organization |
| MATTS | Mobile Agent Topology Test System |

| | |
|---|---|
| MANets | Mobile Ad-hoc Networks |
| MEID | Mobile Equipment Identifiers |
| MoD | Ministry of Defence |
| NHS | National Health Service |
| NTDR | Near-Term Digital Radio |
| NTP | Network Time Protocol |
| OS | Operating System |
| OSI | Open Systems Interconnection |
| P2P | Peer-to-Peer |
| PARC | Palo Alto Research Centre |
| PCBone | Pervasive Computing Bone |
| PC | Personal Computer |
| PDA | Personal Digital Assistant |
| PEG | Parsing Expression Grammar |
| PID | Personal Identity |
| PRNET | Packet Radio Networks |
| PKI | Public Key Infrastructure |
| RBAC | Role Based Access Control |
| RFID | Radio Frequency Identification |
| SID | Social Identity |
| SIM | Subscriber Identity Module |
| SMTP | Simple Mail Transfer Protocol |
| SoS | System of Systems |
| STVN | Segment-Tree Virtual Network |
| SSA | Secure Situation Awareness |
| SURAN | Survivable Adaptive Radio Networks |
| TCP | Transfer Control Protocol |
| TDMA | Time Division Multiplex Access |
| UCIM | User-centred and Context-aware Identity Management |
| UDID | Unique Device Identifier |
| UDP | User Datagram Protocol |
| UI | User Interface |
| XML | Extensible Mark-up Language |

**ACKNOWLEDGMENTS**

I would like to take the opportunity to express my thanks and gratefulness to the people who have helped and supported me throughout my PhD study. Without their support and contribution, successful completion of my research would not have been possible.

First and foremost, I will like to thank my parents for their encouragements and putting me into school from my early days. I would like to express my gratitude to my first supervisor, Professor Qi Shi, for his invaluable contributions, continuous support, encouragement, guidance and invaluable suggestions during this research. I would also like to say special thanks to my second supervisor, Professor Madjid Merabti, for his numerous helpful discussions and his valuable time to help me in various stages of my project, in particular, giving valuable suggestions and comments on my work.

Additionally I am deeply appreciative to Dr. David Llewellyn-Jones for his sincere guidance and encouragement throughout my research, which has been invaluable and unsparing. Consequently I do want to show my deep appreciation to him.

My acknowledgement will be incomplete without expressing my thanks to the research administrators, namely Tricia Watson, Lucy Tweedle and no other than the lovely Carol Oliver, the PA to the School Director. Their moral and friendly approach has really helped me to feel at home, and they have also spoiled me a bit☺. Moreover, the other researchers, administration staff and technicians in the School of Computing and Mathematical Sciences at Liverpool John Moores University have also played an important role throughout my study, and I would like to express my appreciation to their support over the past years.

## DEDICATION

My Family

# CHAPTER ONE

## INTRODUCTION

*"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it"* – Mike Weiser 1991.

We once assumed that Personal Computers (PC) would be the main medium of using the Internet and consumption as a device within all sectors of the economy. But currently in most markets the mobile Internet is overtaking the fixed Internet phenomenon. With the continuous growth and development of computer networks, the notion of ubiquitous computing coined by Mark Weiser has received increasing attention. This notion also leads to the proliferations in usage of Mobile Ad-hoc Networks (MANets). However, this evolution faces a barrier in many ways. On the one hand, people want to construct a ubiquitous network to make the best use of computers. On the other hand, they must secure their network and protect their identity information as well as the need to be in full control of their information, in order to deal with a number of security threats from malicious entities. One solution for this is to provide a framework that will offer the users with such abilities in an efficient, dynamic and lightweight format, allowing users to be in full control of the system so as to minimise or eliminate relevant security threats.

The main focus of this thesis is a combined study of identity management, context-awareness and user-centricity together with their security issues and implications in MANets and emergency situations where different systems need to interact in an ad-hoc manner. The emergent notion of ubiquitous computing makes it possible for mobile devices to communicate and provide services via networks connected in an ad-hoc manner. The use of contextual information in ad-hoc environments can extensively expand the adaptation and usage of such applications. Context is information that can be used to characterize situations or an entity that is considered relevant in the interaction process of a user or an

application. Context-awareness allows us to make use of partial identities as a way of user identity protection and node identification. This is coupled with user-centricity aiming at putting users in control of their partial identities, policies and rules for privacy protection. These principles help us to propose an innovative, easy-to-use identity management framework for MANets. The framework makes the flow of partial identities explicit, gives users full control over such identities based on their respective situations and contexts, and creates a balance between convenience and privacy. The thesis presents the development of the proposed framework and its methodologies, and outlines some possible future work to improve the framework.

This chapter is organized as follows. First, the topic of the thesis and background information is presented. Second, the aims and objectives of the project are stated. Third, the novel contributions of the approach hypothesized in the thesis are summarised. Fourth, we present the project achievements in terms of research publications that consist of book chapters, journal and conference papers. Fifth, a brief summary of our methodology is presented. Sixth, an overview of the chapters of the thesis is provided. Finally, the chapter is summarized.

## 1.1 BACKGROUND

With the emergence and development of wireless networks, the notion of "Ubiquitous Computing" coined by *Mark Weiser [1]* has received increasing attention. One of the fundamental building blocks for such ubiquitous computing applications is MANets, and is increasingly used to support mobile and dynamic operations such as emergency services, disaster relief and military networks. MANets can be defined as a platform or a set of nodes that can move freely and establish a transient self-configuring wireless network. A MANet offers a temporary network without relying on any predetermined network infrastructure, and communicates in a self-organising manner. Moreover, MANets play crucial roles in many application areas such as surveillance, marketing and military [2].

While bringing huge benefits to these applications, they also raise serious privacy/security concerns, more specifically on the protection of users' private information and identity.

Users currently rely on numerous forms of identities to access services via MANets. The inconvenience of processing and using these identities creates significant security vulnerabilities as well as significant user discomfort, including the disclosure of personal information. These growing trends have raised serious concerns over identity management (IM) due to a dramatic increase in identity theft [3, 4]. IM in this context is about managing relevant digital identities of users and ensuring that they have fast, reliable and secure access to distributed resources and services via MANets within ubiquitous computing environments.

Ubiquitous computing has the capability of providing computational environments that facilitate the provision of information through the use of "invisible interfaces" and allowing limitless sharing of information. If developed properly, ubiquitous computing could offer an invaluable support for many aspects of our society and its institutions. However, neglecting the above mentioned security or privacy issues and aspects such as proper integration of contextual data, use of efficient user control and centricity, and the adaptation of relevant access control policies can present a great likelihood that the end products will resemble an Orwellian nightmare [5].

### 1.2 PROJECT AIMS AND OBJECTIVES

The main goal of this project is to develop the interoperability among different IM techniques and to propose a User-centred and Context-aware Identity Management framework (UCIM) for fulfilling such interoperability in MANets. The key objectives of the research are to make the proposed UCIM efficient, user-centred, context-aware and lightweight so as to meet the specific needs of MANets. Context-awareness allows us to make use of partial identities as a way of user identity protection and node identification. User-centricity is aimed at putting users in control of their partial identities, policies and rules for privacy protection.

Efficiency and lightweight on the other hand allow the application to be deployed on devices with less memory and processing requirements in order to increase the adaptability and usability of UCIM.

UCIM should possess a number of capabilities. Particularly, it should be able to manage multiple identities for different MANet situations and perform negotiations with other peers about necessary identity information to be used for identification in relation to given security policy settings and situation awareness. UCIM should also be able to provide users with friendly control over their own identity information and security in terms of mobility across different MANets. Moreover, it should be able to separate users from the complexity of the technical implementation and operation issues of IM in MANets while allowing users to focus on policy aspects of IM. This ability is essential, as users typically do not want to learn detailed security techniques.

### 1.3 Project Scope

In this section we briefly highlight the scope of our work by pointing out what it is all about and what we are not covering.

The thesis covers the following contents:

- Knowledge and information within our field of research
- Our design of mechanisms for the privacy protection of user information
- Our definition of user profile types needed for different application scenarios
- Our design and implementation of a new dynamic policy specification language
- Our adaptation of XML to make the proposed framework lightweight
- Our design and implementation of a new algorithm for context filtering based on user policies and contextual information

- Our development and evaluation of a new identity management framework, which is context-ware, lightweight and user friendly.

It is worth pointing out that, although our main focus is on ad-hoc networks, the solutions provided in chapter six, the general methodologies proposed in this thesis and the issues covered above are also applicable to other network environments such as P2P. The solutions presented in the thesis include an example on how we apply our solution to a P2P environment. Additionally, the proposed framework includes an element of a centralised component. This is mainly for the purpose of events of critical importance, which have a need for ad-hoc devices to be authenticated for retrieving some information from a single trusted source.

The thesis does not cover the contents below:

- Acquiring contextual information using sensors or other means
- Developing communication protocols for the dissemination of contextual information among devices

The above two points imply that our work is based on the assumption that necessary contextual information is available to individual devices when needed.

### 1.4 NOVEL CONTRIBUTIONS OF THIS PROJECT

Our main novel contributions include a new methodology (i.e. a new framework with a user policy definition ContextRank), utilisation of an existing technology in a new way (i.e. Identity and Social Identity Theory), and improvement of other methodologies (e.g. hybrid metrics based on energy metrics, and Context Based Access Control (CBAC) based on Role Based Access Control (RBAC)). The details of these contributions are summarised below:

- **New Framework UCIM:** Our framework represents contextual information and user profiles using XML semantic representation. It facilitates the realisation of a more balanced solution to IM in MANets to cater for the

desirable features of privacy, user-centricity, context-awareness and user friendliness in a systematic and consistent manner. At present, based on our knowledge there is no such framework available for MANets.

- **User-Centred:** Our framework gives total control of the system to users in terms of profile disclosure as well as the usage and definition of rules and policies. We introduced an improved version of the Role Based Access Control concept, i.e. Context Based Access Control (CBAC), where the context constraint of a profile type is used to restrict which users within the environment will be able to request and view other users' contextual information. The usage of portable user profiles within our framework enables users to be in total control of their information usage and be able to do so with minimal technical knowledge. This mechanism is highly portable and consistent with the mobility feature of ubiquitous networks.

- **Dynamic Policy Specification:** This allows users to dynamically specify policies using predicates and functions. Policies are built from predicates (which are just inequalities containing properties and values) and sub-functions, joined together using logical operators. The use of sub-functions eliminates the problem of tackling complex mixtures of conjunctions and disjunctions. Policy can be easily translated and used in many other applications. The methodological design can be implemented for any scenario and is adaptable. Policies are aimed to be stored in a tree structure in memory or in XML files in a way that is lightweight and portable. The design is modular, meaning that new modules can be plugged in to improve functionality. The technique could be easily integrated into current existing solutions, *e.g.* Ponder, to act as client applications and pass on created policies to the Ponder engine for processing. Policies can also utilize users' partial identities. Users can modify policies, which will take effect dynamically and with minimal interruption of the system.

- **ContextRank:** This algorithm is designed to function in two main ways: either used as a filtering algorithm considering only the relevant contextual information for a user, or integrated with user defined policies from a Dynamic Policy creation module (to be presented in Chapter five) or any other policy in filtering context. It is designed to take in the criteria under which contextual

information can be filtered, and to produce an output for the users who meet the required criteria. The expected inputs for the algorithm are: an array or a list of all available contextual information within the range of a given user and one or more policy files. The expected outputs are: a list of relevant contextual information for the user, access to users' data or profile information and possibly a request for further information if required.

- **Lightweight:** This is achieved via the use of XML representation of contextual information and new resource-efficient schemes, including protocols and mechanisms that are conceived for the formation and distribution of context information and the negotiation of identity information. A novel hybrid Euclidean Metric based algorithm has been devised to determine how to access or request contextual profiles from other nodes by using a restrictive hybrid metric that measures the balance of system resources such as energy levels, CPU usage and distances between nodes. Accordingly, the framework is also designed in a way that the user effectively influences the resource usage by introducing the concept of user-centric design to reduce the additional resources consumption on networks. The use of identity and social identity theory in influencing relevant context information to be displayed to the user will also help to limit the overhead of the framework on devices.

### 1.5 PROJECT ACHIEVEMENTS

Our framework UCIM is the first framework that is proposed for identity management in MANet within ubiquitous computing environments by utilising both contextual information and user-centricity. It keeps the special requirements of ubiquitous computing in mind throughout its design and implementation. The methods used in UCIM set a new direction for future research and development. Practically, it ensures the feasibility and realization of identity management in ubiquitous computing and provides means of creating and modifying policies via the use of a user interface rather than hard coding such policies.

The outcomes of our research have generated the following book chapters, journal and conference papers amongst other technical reports and documentations:

### 1.5.1 BOOK CHAPTERS

- Arabo, A., Shi, Q., and Merabti, M., ContextRank: Begetting Order to Usage of Context Information and Identity Management in Pervasive Ad-hoc Environments, in Emerging Pervasive and Ubiquitous Aspects of Information Systems: Cross-Disciplinary Advancements, in Information Science Reference (an imprint of IGI Global), Symonds, J., Editor. 2011. p. 275-297
- Arabo, A., Shi, Q., and Merabti, M., Dynamic Device Configuration in Ubiquitous Environments, in Global Security, Safety, and Sustainability Communications in Computer and Information Science, Tenreiro de Magalhaes, S.J., Hamid; Hessami, Ali G. (Eds.), Editor. 2010, Springer. p. 263-273.
- Arabo, A., Shi, Q., and Merabti, M., Situation Awareness in Systems of Systems Ad-hoc Environments, in the 5th International Conference on Global Security. Safety and Sustainability 2009, CCIS, H. Jahankhani, A.G. Hessami, and F. Hsu (Eds.). Springer-Verlag Berlin Heidelberg, 2009. p. 27-34.

### 1.5.2 JOURNALS

- Arabo, A., Shi, Q., and Merabti, M., Privacy Preserving Identity Management in Pervasive Ad-hoc and Context Sensitive Environments. Journal of Ambient Intelligence and Humanized Computing (AIHC), 2012 in press.
- Arabo, A., Shi, Q., and Merabti, M., Context-Aware Identity Management in Pervasive Ad-hoc Environments. International Journal of Advanced Pervasive and Ubiquitous Computing, 2009. **1**(4): p. 29-42.
- Arabo, A., Shi, Q., and Merabti, M., A Framework for User-Centred and Context-Aware Identity Management in Mobile ad hoc Networks (UCIM). Ubiquitous Computing and Communication Journal -Special issue on New Technologies, Mobility and Security, 2009. **NTMS - Special Issue**
- Arabo, A., Shi, Q., and Merabti, M., Ubiquitous Secure Cash Withdrawal. International Journal of Database Theory and Application, 2009. **1**(2): p. 55-64.

### 1.5.3 CONFERENCES

- Arabo, A., Kennedy, M., Shi, Q., Merabti, M., Llewellyn-Jones, D., and Kifayat, K., Identity Management in System-of-Systems Crisis Management Situation. The 6th IEEE International Conference on Systems of Systems Engineering, 2011, Albuquerque, New Mexico, USA. p. 37-42.
- Kifayat, K., Arabo, A., Drew, O., Llewellyn-Jones, D., Shi, Q., Merabti, M., Waller, A., Craddock, R., and Jones, G., State-of-the-Art in System-of-Systems Security for Crisis Management. The Fourth Annual Layered Assurance Workshop (LAW 2010), Dec 2010, Texas, USA. 2010.
- Arabo, A., Shi, Q., and Merabti, M., Data Mishandling and Profile Building in Ubiquitous Environments. IEEE International Conference on Privacy, Security, Risk and Trust. 2010, IEEE Computer Society, Minneapolis, Minnesota, USA. p. 1056-1063.
- Zhou, B., Drew, O., Arabo, A., Llewellyn-Jones, D., Kifayat, K., Merabti, M., Shi, Q., Craddock, R., Waller, A., and Jones, G., System-of-Systems Boundary Check in a Public Event Scenario. The 5th IEEE International Conference on Systems of Systems Engineering, 2010, Loughborough, UK **(Won Best Paper).**
- Muhammad, A., Arabo, A., Merabti, M., Shi, Q., and Askwith, B., A Secure Gateway Service for Accessing Networked Appliances. The Fifth International Conference on Systems and Networks Communications, 2010, Nice, France, IEEE Computer Society.
- Arabo, A., Shi, Q., and Merabti, M., Dynamic Policy Specification in Ubiquitous Environments. The 11th Annual PostGraduate Symposium on The Convergence of Telecommunications, Networking and Broadcasting, 2010, Liverpool, UK.
- Arabo, A., Shi, Q., and Merabti, M., User-Centred Identity Management in Mobile Ad-hoc Networks. The 10th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (PGNet 2009), 2009, Liverpool, UK.
- Arabo, A., Shi, Q., and Merabti, M., Towards a Context-Aware Identity Management in Mobile Ad-hoc Networks (IMMANets). The IEEE 23rd

International Conference on Advanced Information Networking and Applications Workshops (AINA-09), 2009, Bradford, UK.

- Arabo, A., Shi, Q., Merabti, M., and Llewellyn-Jones, D., Identity Management in Mobile Ad-hoc Networks (IMMANets): A Survey. The 9th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (PGNet 2008), 2008, Liverpool, UK.
- Zhou, B., Arabo, A., Drew, O., Llewellyn-Jones, D., Merabti, M., Shi, Q., Waller, A., Craddock, R., Jones, G., and Yau, A.K.L., Data Flow Security Analysis for System-of-Systems in a Public Security Incident. The 3rd Conference on Advances in Computer Security and Forensics (ACSF 2008), 2008, Liverpool, UK.
- Arabo, A. Secure Cash Withdrawal through Mobile Phone/Device. International Conference on Computer and Communication Engineering (ICCCE 2008), 2008, Malaysia.

### 1.5.4 CITATIONS

Based on our knowledge, some of our publications and work have been cited by the European Network and Information Security Agency (ENISA) in their April 2010 report on Mobile Identity Management, a keynote speaker of the International Conference on Security and Identity Management (SIM) 2009 and other publications.

- En-Nasry, B. and Kettani, M.D.E.-C.E., Towards an Open Framework for Mobile Digital Identity Management through Strong Authentication Methods in Secure and Trust Computing, Data Management, and Applications, Communications in Computer and Information Science. 2011, Springer. p. 56-63.
- Merabti, M., Kennedy, M., and Hurst, W., Critical Infrastructure Protection: A 21st Century Challenge. 2011 International Conference on Communications and Information Technology (ICCIT), 2011. p. 1-6.
- ENISA, Mobile Identity Management, Papadopouli, M. (Editor), http://www.enisa.europa.eu/act/it/eid/Mobile%20IDM/at.../fullReport,

Accessed in April 2010.

- Jagannathan, S., Social Network and Identity Management. International Conference on Security and Identity Management (SIM) (Keynote Speaker), Ahmedabad, India, 2009.
- Carlos Rodríguez-Domínguez ; Kawtar Benghazi ; Manuel Noguera ;José Luis Garrido: Redefinable Events for Dynamic Reconfiguration of Communications in Ubiquitous Computing. The First International Workshop on Data Dissemination for Large Scale Complex Critical Infrastructures, 2010, p17-22 (ACM publication).
- Saleh, Z. and Alsmadi, I., Using RFID to Enhance MobileBanking Security. International Journal of Computer Science and Information Security (IJCSIS), 2010. **8**(9): p. 176-182.
- Quah, R., Securing Emergency State Data in A Tactical Computing Environment. NAVAL Postgraduate School Monterey, 2010, California, USA.
- Ngo, H.H., Dandash, O., Le, P.D., Srinivasan, B., and Wilson, C., Formal Verification of a Secure Mobile Banking Protocol in Advances in Networks and Communications in Computer and Information Science. N. Meghanathan et al., Editors. 2011, SpringerLink. p. 410-421.

We have also been requested to summit book chapters from several publishing institutions such as IGI Global on a book titled "Emerging Pervasive and Ubiquitous Aspects of Information Systems: Cross-Disciplinary Advancements" and an INTECH new book project under the working title "Theory and Applications of Ad-hoc Networks", 978-953-7619-X-X.

Our work has also been recognised by both conference organising committees and journal editors. Hence, I have been invited as a Technical Program Committee member of several important IEEE conferences such as IEEE Consumer Communication & Networking Conference (CCNC) short papers 2009- 2011; International Conference on Electronic Devices, Systems and Applications (ICEDSA) 2010-2011; IEEE Symposium on Industrial Electronics & Applications

(ISIEA) 2009-2011 and IEEE Applied Power Electronics Colloquium (IAPEC) 2011, and a paper reviewer for the International Journal of Network Security from 2009. I have also been a season chair for IEEE International Conference on Privacy, Security, Risk and Trust (PASSAT) 2010.

## 1.6 METHODOLOGY

The process for achieving the project aims and objectives is divided into the following phases: requirement analysis and specification, design, implementation and evaluation. The first phase, requirement analysis and specification, is based on identifying current works within the domains of MANets and IM, and capturing necessary requirements for the proposed framework. This includes an extensive literature survey to obtain the comprehensive knowledge of existing IM systems, MANets, lightweight privacy-enhanced mechanisms, key management, efficient trust management, context-aware systems, user-centric concepts and mechanisms. An awareness of the issues and limitations of these available techniques are then established. To tackle the issue of identities, personal and device attributes are investigated to derive a set of partial identities suited to MANet-based applications, which help to specify privacy policies for access to identity information. From these studies, a set of requirements for the proposed framework are defined and documented.

The design phase of the project involves transforming the requirements identified in the first phase into effective solutions for their realisation, including the design of data structures, system architectures, interfaces and components. Before addressing the data structure issue, methods for collecting information required for IM in MANets have been explored. This helps us to build context-awareness from a mixture of different information such as locations, time, proximity selection, automatic contextual reconfiguration and context-triggered actions. For outdoor location information we suggest obtaining such information via the use and exploration of GPS technology, and provide a means that will allow the possibility of using active badges when required. As many small and low-cost devices do not necessarily have GPS, the method proposed by *Bulusu* [6] using a connectivity-

based localisation technique is explored. Since GPS does not support data communication [7] in such situations, our framework provides the means of allowing devices to track locations by listening to beacons from a cell-based station or query a local database for their current locations. Users can choose to advertise their current locations with respect to their privacy policies. Other low-level context information such as time, nearby objects, network bandwidth and orientation is also examined with reference to necessary quality context indicators [8].

The design phase also selects other suitable existing techniques for IM, key management, efficient trust establishment and lightweight privacy enhancement in relation to the requirements set out in the first phase. These techniques are then tailored, extended and integrated together with the context-aware solutions to produce necessary schemes, mechanisms and protocols, which collectively form the proposed framework.

In the implementation and evaluation phases, simulation techniques and tools [9] such as NS-2 [10], GTNet [11] and OPNET [12] are examined to find an effective way to implement the framework designed. However, we decided to broaden our in-house developed simulator for testing the security of composed systems-of-systems to the needs of our framework and provide means of integrating the new simulator to work with other simulators when time permits. The main reason for using a simulation tool for the implementation is due to its cost-effectiveness for the framework evaluation and subsequent refinement as well as the limitation of resources needed for a real implementation. The simulated framework is assessed based on case studies to determine its compliance with the requirements specified in the first phase. We also try to test the framework on available portable hardware devices within the laboratory.

### 1.7 THESIS ORGANISATION

***Chapter two:*** It reviews the issues related to the development of ubiquitous computing from its inception by Weiser, where the three main phases: the

Mainframe era, PC (Personal Computing) era and the Ubiquitous Computing era are presented. A brief history of computer networking is covered while presenting its future direction. The chapter also looks at the layered network in relation to the development of network security. It continues with several related areas in which smaller and cheaper computer chips are embedded into many appliances from a greeting card to a smart home so that people's daily lives can be closely connected to computers and beneficially become ever more convenient. Finally, along with the benefits, vulnerabilities of ubiquitous computing are discussed. Security is one of the major concerns for any computer network, including ubiquitous computing.

***Chapter three:*** In this chapter, we will narrow down the related work and provide more insight into the areas provided in chapter two. The chapter first introduces network security and identity management. It then further presents the historical issues (e.g. development, attacks and frameworks) of IM and an introduction to MANets in relation to IM. An overview of research issues in these areas is provided. The chapter also classifies users' personal identity into sub-identities to make it easier to protect user identities and define policies, which will be further expanded in the chapter five of the thesis. The chapter explores the importance of user-centricity and context awareness for both MANets and IM by highlighting how these two techniques can help to provide more security to users and networks. Other benefits include the efficiency of resources used, its impact on helping to create a lightweight framework, its ability to help improve service provisions, etc. Context reasoning and representation have also been introduced in this chapter.

***Chapter four:*** In this chapter, we will summarise the weaknesses of the related work identified in chapters two and three. The weaknesses identified are mainly focused on IM, including the issues of defining user profiles, user control and centricity, the usage of contextual information, and allowing users to define policies dynamically. These provide a justification for proposing a new framework to solve the weaknesses. For example, existing frameworks and solutions for IM are mainly aimed at wired networks, which do not meet the requirements of ubiquitous ad-hoc environments, and also they neglect the important role of users

and are not lightweight. From these identified weaknesses we will present our motivations for the thesis. Building from the identified limitations, the chapter also presents the aims and objectives of our project for the development of the proposed UCIM framework and provides the set of requirements for UCIM.

***Chapter five:*** Based on the identified limitations, project aims, objects and requirements in chapter four, this chapter presents the overall design of UCIM showing what modules or components are needed to create the framework. The chapter further looks into the details of each module by focussing mainly on how they meet the specified requirements of the framework and how they overcome the limitations summarised in chapter four. Some desirable features such as portability, heterogeneity, lightweight, dynamic policy creation and context filtering are also explained in detail with regard to the proposed framework.

***Chapter six:*** In this chapter we provide the proof of concept of UCIM. The proposed UCIM framework is implemented by extending our in-house built simulator, and developing new mobile and other relevant applications that provide the functionality of UCIM. The implementation is supplemented by scenarios that are aimed at helping readers understand the reason behind such implementations and methods used. The implementation is deployed on various portable devices and desktop computers with web based interfaces for other modules of UCIM. The chapter further presents the results of the implementation and deployment using screenshots. It also presents some of the code used in the implementation of UCIM to aid readers' further understanding and show how we have managed to turn our proposed algorithms and mechanisms in chapter five into a workable framework and solution so as to fulfil the project goal.

***Chapter seven:*** This chapter takes us back to the chapter one of the thesis by re-visiting the aims and objectives of the project as presented in both chapter one and five respectively. The chapter further provides the evaluation of the implemented UCIM with respect to the project aims and objectives as well as the requirements set out in chapter five, including a security analysis and test results. The chapter

also provides justifications on how UCIM is able to rectify the limitations of the existing work summarised in chapter four and most importantly fulfil the main aims and objectives of the project.

***Chapter eight:*** This chapter presents our conclusions and future work.

## 1.8 SUMMARY

In this chapter we have presented an overview of the content of the thesis. We have highlighted the fact that computing applications are becoming ubiquitous, which has created serious threats to users' privacy and security. Trivial embedded systems with abilities of computing and communication are becoming widely available and spreading everywhere for the purposes of sensing, control and information display. Hence, the need for the privacy and security protection of users is an inevitable and critical issue. These must be planned effectively to meet the future large-scale implementation and deployment of ubiquitous computing applications. Current IM frameworks are not fit for such an environment due to the resource constraints and heterogeneous infrastructure of ubiquitous computing. Therefore, this thesis provides a novel solution to the problem: UCIM for users' identity protection in MANets. This framework utilises flexible and adaptive system architecture to provide resource-efficient security protection against malicious activities, and gives users full control of their identity information.

In the next chapter, the history of computer networks and the trend towards ubiquitous computing will be introduced.

## CHAPTER TWO

## THE DEVELOPMENT OF UBIQUITOUS COMPUTING

In this chapter, a brief introduction to the history of computer networks is presented. Then our focus is directed to the continuous growth and development of computer and network technologies, including introducing the early stage of the information era – ubiquitous computing and highlighting its possible future directions. Figure 1 depicts a graphical representation or summary of the development of computers, which is classified into three main phases: the mainframe era, PC (Personal Computing) era and ubiquitous computing era. These also summarise the main contents and points of the discussion in this chapter.

Since the first version of mainframes, a number of decades have passed by. It is only in the 1960's that the notion of computers, which was supposed to work alone and process programs locally, set a new milestone and initiated a change pattern for this concept. The new concept came about allowing a set of computers to be connected together to allow remote access to computer resources. Since then, the world has witnessed one of the greatest miracles in human history – the Internet. Following this, the notion of ubiquitous computing was explored by Mark Weiser in the CS lab at Xerox PARC [13].

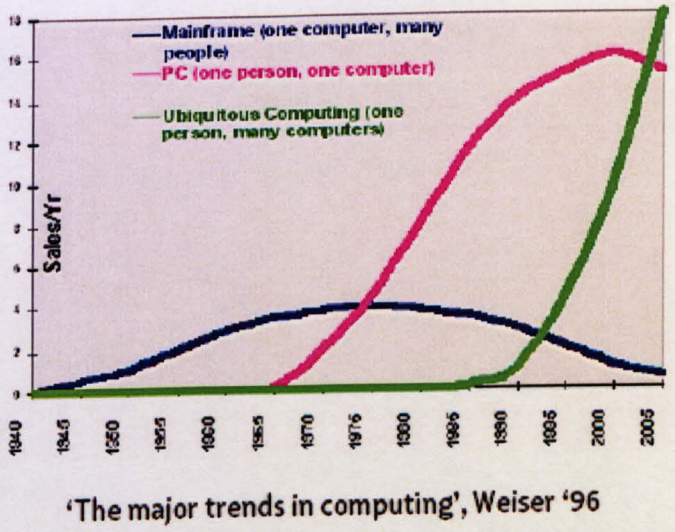


Figure 1 Development of Computing

When analysing such developments within computer technology over the last century, a clear trend of events and shifts can be seen. Over the years as time goes by, when mapping the relation of how many people use how many computing devices, it can be seen that it goes from many-one in the 1960's to many-many in 2010. A summary of this trends has been presented in Figure 2 [14].

Figure 2 Mainframe to Ubiquitous Computers [14]

The three main developments or waves of computing have been summarised by Weiser as: The first wave of computing from 1940 to about 1980 was dominated by many people using one computer. The second wave, still peaking, has one person and one computer in uneasy symbiosis, staring at each other across the

desktop without really inhabiting each other's worlds. The third wave, just beginning, has many computers serving each person everywhere in the world, which is called 'ubiquitous computing' [1].

Weiser further states that: "The defining words [for the third wave in computing] will not be "intelligent" or "agent", but rather "invisible" and "calm" and "connection"" [1].

The development of ubiquitous computing itself has gone through various phases. The first phase has been where hundreds of computing devices ranging from the size of a memo pad to wall-size boards have become available for users in different ways, i.e. iPads, mobile, body sensors etc. The means of connection between devices are via wireless networks with abilities such as shared meeting applications and location-based services. Weiser envisioned the scenarios where "embedded computers [...] will bring other worlds to us in new ways - sometimes in ways so unobtrusive we will not even notice our increased ability for informed action". Weiser described the kind of tune in a future alarm clock: "the kind of tune [it] plays to wake me up will tell me something about my first few appointments of the day. A quick urgent tune: 8 am important meeting. Quiet, reflective music: nothing until noon". Hence, devices "can be suggestive without being intermediating". Ubiquitous computing would allow us to focus on those issues that are really important, interesting and challenging.

In order to turn Weiser's dream to reality, research organisations and industry need to be more focused on developing the required techniques, hardware and software with the careful consideration of security measures that will motivate users to adapt this new wave of interaction and gain its benefits. This research will include several areas related to such issues as: security protocols, policy definition interfaces, user-centricity, context-awareness, low cost devices, low power devices, identity management frameworks and techniques, mobility, applications for small devices, effective user interface, etc. The above issues are vital in order to encourage users to acclimatize new developments. As in today's growing world of

threats most users will only settle in new technologies when they are satisfied that they are in full control of their personal and identity information.

The concept of ubiquitous computing has other related research areas such as the concept of calm technology, which extends the notion of ubiquitous computing and uses its principles to create technology that utilizes both the centre and periphery of a user's attention. Another area is augmented reality that makes heavy usage of the principles of ubiquitous computing to improve users' perception of computation by enhancing physical objects (such as a desk), using computer generated sensory inputs (such as sounds or graphics) to provide a direct or indirect view of a physical, real-world environment. Such technological advancement allows better interaction and usage of the physical objects. However, these areas are not related to our work.

In the following sections we will turn to the issues of computer networks and security.

## 2.1 A BRIEF HISTORY OF COMPUTER NETWORKS

The initial stage of computers was stand-alone mainframe computers with each occupying an entire room. As time went by with the developments of more portable mainframe computers, the industry realised that there would be an advantage if these supercomputers could be connected in a way to allow them to talk to each other, i.e. sharing information. From this, the notion of computer networks was born. Hence, the term 'network' in computer science can be seen as a means of interconnecting computer systems by making use of transmission technologies.

The early development of computer networks came about in the late 1960's with the main aim of connecting researchers to remotely accessible expensive computer resources that individual research centres or institutions could not afford. This came about via the ARPA (Advanced Research Projects Agency) project

ARPAnet [15], which produced the first prototype of modern networks. Its connection speed at that time was 50 kbits/s, but ARPAnet brought a fundamental change from centralized to distributed computing and incorporated features of reliability and robustness, e.g. multiple links and distributed routing. The ARPAnet project, initiated in 1969, is sometimes referred to as the grandfather of the Internet. The network was designed as a computer version for a nuclear bomb shelter to protect the flow of information between military installations by creating geographically separated computers capable of exchanging information via the use of the Network Control Protocol (NCP). The initial connection consisted of four computers from the UCLA (University of California Los Angel) Research Lab, Stanford Research Institute, UC (University of California) Santa Barbara and the University of Utah. The first data exchange within the network was between UCLA and Stanford Research Institute. In the first attempt to log into the Stanford's computer by typing "login", researchers in UCLA managed to crash the network when they typed the letter 'g'.

From then on, the ARPA development led to an array of new hardware and protocols, and this eventually emerged as the Internet. The Internet is defined as a set of networks connected by routers that are configured to pass traffic among any computers attached to a network in the set. Initially the Internet had only a few hundred computers and a few dozen sites. Today, hundreds of millions of computers, small portable devices and thousands of networks worldwide are connected together.

Another analogy on the development of the Internet as seen by Frank Casanova, who was a Director of Apple Computer Inc. is [16]:

"The concept of computers as things that you walk up to, sit in front of and turn on will go away. In fact, our goal is to make the computer disappear. We are moving towards a model we think of as a 'personal information cloud'. That cloud has already begun to coalesce in the form of the Internet. The Internet is the big event

of the decade. We'll spend the next 10 years making the network as it should, making it ubiquitous."

The development of the Internet brought about a new industry. Companies like Cisco, IBM and Microsoft continuously work out new products on networking hardware, computers and relevant software. Today, the Internet has become a new phenomenon that networks are an important part of everyday activities. Through the Internet, we can do shopping at home, finish a degree without going to a university, make friends with people from anywhere of the world, etc. In many ways, it changes the way we live. Currently, it is even possible to produce specially designed objects (tables, chairs, cloths, shoes, etc.) from a computer and print them using a 3D printer. In [17], Weinberg wrote "the ability to reproduce physical objects in a small workshop and at home is potentially just as revolutionary as the ability to summon information from any source onto a computer screen".

## 2.2 Mobile Ad-Hoc Networks (MANets)

A MANet is a collection of mobile nodes forming a network on demand without the assistance of any centralized structures. These networks can be well used and suitable for environments where either the infrastructure is lost or the deployment of an infrastructure is not very cost-effective. In today's growing use of technology, MANets are used in a number of environments for various reasons. For example, the military can employ them to track an enemy tank as it moves through a geographic area covered by the networks. Your local community can use an ad-hoc network to detect your car moving through an intersection and to check the speed and direction of the car. In an environmental network, you can find out temperatures, atmospheric pressures, amounts of sunlight, and relative humidity at a number of locations.

Historically, the whole life cycle of ad-hoc networks could be categorized into the first, second, and third generation ad-hoc network systems. Existing ad-hoc network systems are considered the third generation [18].

The first generation goes back to 1972. At the time, they were called PRNET (Packet Radio Networks) [19]. PRNET were used on a trial basis to provide different networking capabilities in a combat environment.

The 1980's provided us with the second generation of ad-hoc networks. When the ad-hoc network systems were further improved and implemented as a part of the SURAN (Survivable Adaptive Radio Networks) program.

In the 1990's, the concept of commercial ad-hoc networks arrived with notebook computers and other viable communication equipment. At the same time, the idea of a collection of mobile nodes was proposed at several research conferences.

The IEEE 802.11 subcommittee had adopted the term "ad-hoc networks" and the research community had started to look into the possibility of deploying ad-hoc networks in other areas of application.

In the intervening time, the work was going on to advance the previously built ad-hoc networks. GloMo (Global Mobile Information Systems) and the NTDR (Near-term Digital Radio) [20] are some of the results of these efforts. GloMo was designed to provide an office environment with Ethernet-type multimedia connectivity anywhere and anytime in handheld devices.

The requirements of MANets represent a spectrum of network challenges. During the last few years, almost every aspect of MANets has been explored to some level of details. Yet, more questions have arisen than been answered [21]. The major open problems are listed as:

- *Identity Management* - How to protect users' identity information such as text, address books, emails, personal information, etc.

- *Autonomous* - No centralized administration entity is available to manage the operation of the different mobile nodes.
- *Dynamic topology* - Nodes are mobile and can be connected dynamically in an arbitrary manner. Links of the network vary timely and are based on the proximity of one node to another node.
- *Device discovery* - Identifying relevant newly moved in nodes and informing about their existence need dynamic updates to facilitate automatic optimal route selection as well as other contextual information.
- *Bandwidth optimization* - Wireless links have significantly lower capacity than the wired links.
- *Limited resources* - Mobile nodes rely on battery power, which is a scarce resource. Also storage capacity and power are severely limited.
- *Scalability* - It can be broadly defined as whether a network is able to provide an acceptable level of service even in the presence of a large number of nodes.
- *Limited physical security* - Mobility implies higher security risks such as peer-to-peer network architecture or a shared wireless medium accessible by both legitimate network users and malicious attackers. Eavesdropping, spoofing and denial-of-service attacks should be considered.
- *Infrastructure-less and self-operated* – A self-healing feature demands that a MANet should be able to realign itself for its protection when some nodes move out of its range.
- *Poor transmission quality* - This is an inherent problem of wireless communication caused by several error sources that result in a degradation of received signals.
- *Ad-hoc addressing* - Standard addressing schemes should be developed.
- *Network configuration* - The whole MANet infrastructure is dynamic and is the reason for the dynamic connection and disconnection of variable links.

- *Topology maintenance* - Updating the information of dynamic links among nodes in MANets is a major challenge.

The issues mentioned above still forms the fundamental research issues today. In our work and sections to follow we try to address some of the issues by focusing more on the security aspects. Having introduced the history of computer networks, the next section will look at security issues within computer networks.

### 2.3 AN OVERVIEW OF NETWORK SECURITY

With the rapid developments of computer networks as summarised in the previous section, security issues become of concern to the community, and ways of trying to tackle such issues becomes of paramount importance. Looking back 20 years ago, network security was mainly at the level of separating networks. In 1994, William Cheswick and Steven Bellovin wrote a book titled 'Firewalls and Internet Security' [22]. This book defines the prevailing mentality that vulnerable services and vulnerabilities in general are the source of security breaches. At that time vulnerabilities were mainly caused by errors in code rather than hacking into a system. To some extend it is still true today.

However, it was in 1993 when a classic paper titled 'Improving Security of Your Site by Hacking Into It' [23] was published, which shined a further spotlight on defending against security breaches. With the development of the Internet in the early 1990's, this led to a dramatic change in the scope of information and computer/network security. This was a consequence of the requirements for opening up networks to promote information and computing resource sharing, which made the networks far more vulnerable to new threats. Hence, network security has been challenged to adapt to new and constantly changing threats while trying to keep up with performance. This has created the need of a connection-based security – affecting network layers 3 and 4 respectively.

One of the most widely used models for networks is the International Standards Organization's (ISO) Open Systems Interconnection (OSI) 7-layer reference model [24]. However, it is worth pointing out that the OSI 7-layers are not the same as the Internet stack. The Internet Stack mainly provides a number of trace sources in its protocol implementation. The 7-layer model is the main guide to designing network protocols. Each of the layers is named and numbered from bottom to top as shown in Figure 3. Each layer is designed to fulfil specific functions in communications within a network setting. The application layer consists of a number of protocols that are commonly needed, for example, File Transfer Protocol (FTP) and Simple Mail Transfer Protocol (SMTP). The presentation layer defines the common formats for the representation of data. The session layer manages sessions such as login to a remote computer. The transport layer is designed for a main functionality to let computers carry on a conversation and is the heart of the whole protocol hierarchy. Two types of transport service, i.e. the connection-oriented Transmission Control Protocol (TCP) and the connectionless User Datagram Protocol (UDP. The network layer is in charge of address assignment and data delivery across a physical network by using the Internet Protocol (IP). The data link layer formats data in frames and delivers frames through a network interface.

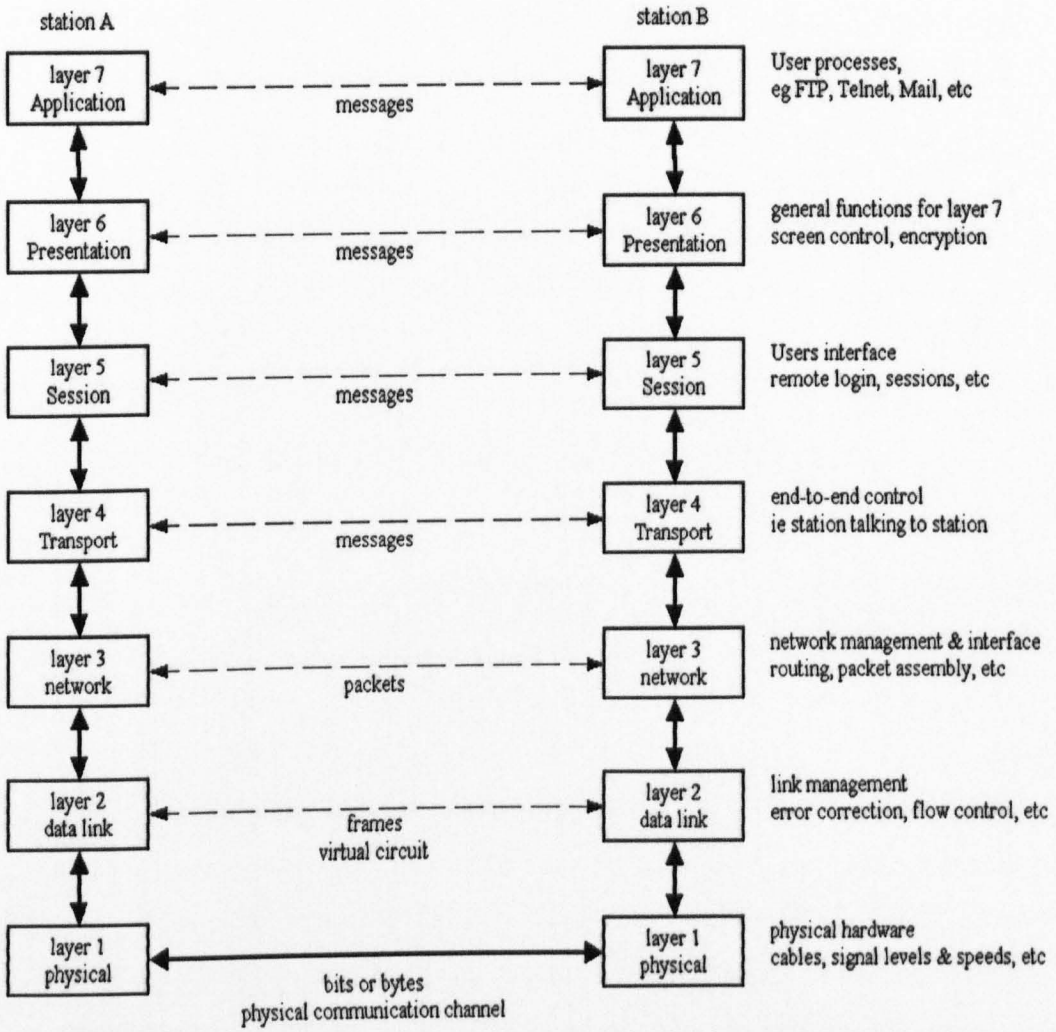


Figure 3 ISO OSI 7-layer Reference Model

After the initial stages have been developed and the basic firewall phenomenon has been examined, the trend in network security has consistently moved up towards layer 7 in the network stack. What started out as computer security has evolved through network security, enterprise security and information security in the years that follow to get where it is today. Whereby, every aspect of technology or our life is analysed to see how to break networked systems or make systems more secure in the ubiquitous world we are currently living in.

The flexible layered architecture allows multiple networks and computers to be connected in a seamless way, irrespective of the requirements demanded by

various applications. Software implemented from the layered design has layered organization. The software for each layer depends only on the services of the software provided by lower layers. The software at layer $n$ at the destination receives exactly the same protocol message sent by layer $n$ at the sender. This implies that the protocols designed can be tested independently and replaced within a protocol stack. The software at each layer communicates with the corresponding layer through information stored in headers. Each layer adds its header to the front of a message from the next higher layer. Headers are nested at the front of the message as the message traverses the network.

In the past years, we have seen an enormous development of market segments, methodologies and security requirements. In a conference in 1978 [25], the main issues of concern for the conference were policy development, disaster recovery planning, data centre physical security and the new technology of access control systems. Well, arguably these areas highlighted above are still some of the key security concerns today. The main difference is that in the early stage, researches were concerned with protected systems (single networks), but in today's world from the early 1980's everything has changed. With the introduction of portable and affordable computers in 1981 and when Weiser coined his notion of ubiquitous and pervasive environments, these developments make network security much more challenging and interesting. Hence, we are no longer dealing with enclosed networks. While some people are trying to solve loopholes, others are busy on trying to create and find new defects of computer systems.

In the late 1990's, a general overview of traditional (old) generation security models, the trend in security, areas of specialisations, principles and advances was examined and analysed [26]. It highlighted and compared the military security models and computer security models that were used in older days as well as the similarity in the changes that they had both gone through. In the past, everything was centralised, but now everything is becoming more decentralised and distributed, which makes computer security more vulnerable.

Computer security used to be largely perimeter-oriented: block outsiders and give insiders access. Security management was centralised. However, now it is becoming more difficult to differentiate between an insider and outsider, e.g. a Trojan horse program can be downloaded remotely into a local area network. The rapid advance of computing technology is making computer security difficult to keep up with its pace in general.

Hence the following factors have been considered as a must for all security system designers and implementers to put in mind:

- Policy: it is of paramount importance to have/develop a good policy framework. The three properties of security need to be carefully considered while drawing up a policy: confidentiality, integrity and availability (CIA). However, it is always difficult to have CIA at the same time. Therefore a balance between CIA needs to be addressed for a particular environment in consideration.
- Privilege: this deals with the issue of assigning privilege controls to either individuals or a group of computer users.
- Correctness: the needs to mediate access privileges correctly.
- Audit: making use of log files to plan for better security analysis.

Other identified areas, where computer security as a profession has been diversified and specialised within the computing field, include trusted systems, policy, operating systems, database management systems, distributed systems, cryptography, protocols, system correctness, intrusion detection systems, malicious code protection and mobile code.

Hence, it is apparent by looking at the fact that at that time, even now, most of the methods for protecting computer systems were based on building string perimeter defence as was widely used when computer systems were around the mainframe, central administration and distinct organisations era. However, in the current age

these models are no longer applicable due to some trends developed. Some of the trends include: a significant shift from centralised to single user work stations, desks becoming virtualised with virtual workstations (people working from home), companies becoming multinational, and the notion of ubiquitous computing coined by Weiser becoming reality. Therefore, the security paradigm of perimeter protection is becoming less appropriate. The emerging mode of secure computing builds on computer security fundamentals that are established over the last three decades and makes use of self-protecting systems, strong protection mechanisms, dynamic security policies, and rigorous assurance techniques. Nevertheless, the fundamental security goals of confidentiality, integrity and availability still apply.

Figure 4 illustrates an overview of information security presented in the ISO 17799 [27], showing a general relationship between threats, threats agent, assets and vulnerabilities as well as possible safeguards. The Figure has classified threats into three categories: people, accident and nature. Asset has a value and sensitivity level indication which together defines the potential impact of the asset. These are then associated with security risks, and used to define possible safeguards based on security requirements.

Figure 4 Information Security Overview
[27]

## 2.4 COMPUTER SECURITY

Computer security as an area of research comes out as a subfield of computer science, which can be seen as trying to control or minimise risks related to computer systems. This can be in terms of the data stored in computers and physical or remote access to computer resources. In the early development of computer security, a typical traditional approach to security was by physically restricting access to a computer, and then specifying and enforcing a security policy on the computer system to restrict the actions an entity (user or program) could perform. It is still debatable with regards to what a secure action really means. Hence, the notion of security can have different meanings depending on the environment or situation involved, e.g., a university may have a very different notion of security from a military or bank. Consequently, security here is a property that is only one of its kinds to each situation and so must be explicitly defined by a security policy.

In computer security, there is no such state as 100% secure. However, a secure system should still permit authorised users to carry out legitimate and useful tasks. It can be argued that one might be able to secure a computer beyond misuse only by using extreme measures such as those noted by Eugene H. Spafford: *The only truly secure system is one that is powered off, cast in a block of concrete and sealed in a lead-lined room with armed guards - and even then I have my doubts* [28]. However, as we are dealing with real systems that are functioning, not turned off, the system mentioned by Eugene H. Spafford would not be regarded as a useful secure system, but it helps to point out that to have a completely secure system is a continuous challenge or a dream. For that reason, there is always a trade-off between the security and utility of computer systems.

When talking about system security, we usually mean a system that has at least some of the characteristics and challenges of a secure system, which include reliability/availability, safety, integrity, confidentiality, accountability, robustness, etc. Hence, a short description of some related terms is provided below:

- Data accessibility – the contents are accessible to legitimate users.
- Data integrity – the contents are not modified by unauthorized entities. This ensures that any received data has not been altered or deleted in transit. Hence, we should keep in mind that an adversary could initiate attacks when necessary measures are not taken to prevent such attacks.
- Data confidentiality – the contents are not revealed to unauthorized entities. To secure data from eavesdroppers, it is essential to guarantee the confidentiality of private data. To achieve data confidentiality, encryption functions are normally used, which are a standard method. However, to protect the confidentiality of data, encryption itself is not sufficient, as an eavesdropper can perform traffic analysis on the overheard, in order to gain access to sensitive information about the data. Likewise, to avoid the misuse of information, the confidentiality of private data also needs to be enforced via access control policies from the owner of the data.
- Accountability - responsible for tracking who has accessed the data.
- Authorization - responsible for who is allowed to access the data.

As a general concept, computers used in the work environment are interconnected via the Internet. However, the future can lead to the fact that billions of miniature intelligent devices will inhabit the world, their means of connection will be via the wireless medium, and the number of such devices will be continuously increasing more than the human population. These devices will also be embedded into many physical objects, which form what Gartner calls the "Supernet" [29]. A so-called Hyper Cycle Graph is used to illustrate how new and promising technologies will go through various phases of development. Figure 5 known as the Gartner's Cures is taken from a report looking at ICT related technologies [30]. It shows the typical progression of a technology from over enthusiasm, through a period of disillusionment, to an eventual understanding of the technology's relevance and role.

Figure 5 Hyper Cycle

## 2.5 CONCEPT OF UBIQUITOUS COMPUTING

Future access to the Internet will not always be through the traditional means of desktop computers, laptops, mobile phones, ipads, e-book readers etc. Common equipment, like watches, clothes, household furniture or even household appliances, will connect to online information. In a recent development a PaperPhone prototype has been seen to do all the things that a normal Smartphone is capable of doing [31]. The devices look, feel and operate like a small sheet of interactive paper. A user performs relevant usual functions of a Smartphone via bending the PaperPhone into a cell phone, flipping the edge to turn pages, or writing on it with a pen. The authors of the PaperPhone also predicts that everything is going to look and feel in a similar way like the PaperPhone within five years' time, and that this technology will realize the paperless office dream. However, one of the limitations of PaperPhone is that it still needs to be connected

to a laptop for data and action interpretation. The PaperPhone only replaces the interface part of a normal Smartphone functionality. There is another prototype called the Snaplet, which takes on different functions depending on how it is worn. It is a watch when it is convex, a PDA when it is flat, and a phone when it is concave. This can even happen without the users being aware of which appliances are actually connected. Instead of individuals actually making a connection to a network, their tools will automatically connect to it without personal intervention. This concept as invented by Marc Weiser is called "ubiquitous computing". However, the security issues will be more crucial and importance to address and plan beforehand. It is also not about the provision of security, but how to configure it right, especially when these devices are used within emergency ad-hoc environments.

Ubiquitous computing in a way can be referred to as the post-desktop model of Human Computer Interaction (HCI). Users can escape the normal stereotypical interaction of computers via the use of keyboards, mice, etc. Instead, they can interact with systems via everyday objects and activities, e.g. remote controls, gestures and thoughts. Hence, the aspiration of ubiquitous computing is to move computers away from the central focus of users, so that they are used subconsciously to enhance existing tools or communications.

The term ubiquitous computing was brought out by Mark Weiser, who is widely considered as the father of ubiquitous computing, at Xerox PARC (Palo Alto Research Centre) in 1991. His article "The Computer for the 21$^{st}$ century" [1] coined this notion and described a version of technologies that weave themselves into the fabric of everyday life until they are indistinguishable from it. He further explained that the most powerful and successful technologies are those that naturally blend into our world until they are effectively invisible.

Hence, these technologies can become human's second nature due to their usefulness and wide availability. This will result in a situation where people will

stop thinking of themselves as using a technology. Instead, they just consider themselves capable of doing whatever the technology enables and more.

We can elucidate that this approach of understating technology is based on the fundamental principles and observation that the ratio of computers to persons is constantly increasing (see figure 2). Currently, most people might have three or more computers wherever they are either in the office, home or on the move.

As an example, during the first era of computers, this ratio was one mainframe to hundred or even thousand persons. It quickly reached a stage of one-to-one – the personal computer era. Now it is advancing to the situation of one person having more than one computing unit, e.g. a mobile phone, PDA, watch, or iPod – the ubiquitous computing era. As this ratio increases, so is the expectation or intention of users on how to best utilise such facilities in order to cost-effectively interact with the applications built into them. This has prompted some experts in saying that one of the most precious resources in a modern computer system is no longer its processor, memory, and network; but rather a resource that is not subject to Moore's law: user attention [32]. A good example is the communication technologies for mobile phones. If I say "I spoke to my brother in Nigeria this morning", we understand implicitly that they used the telephone networks to do so or Skype or something similar. We would never hear someone saying like "This morning I used the telephone networks to speak ..."

Looking at it from another perspective, pervasive computing, which is a synonym to ubiquitous computing, is defined as the evolution of mobile computing [33]. Mobile computing on the other hand builds on the foundations of distributed systems, mobile networking and energy-aware systems [34]. However, pervasive computing extends these with four additional research challenges: smart spaces, invisibility, localized scalability and uneven conditioning [33]. The main issue surrounding the use of a smart-space more efficiently lies in the fact of trying to bring the physical world together within a conceptual world. Weiser refers to the issue of invisibility as "seamless integration of fabric of our everyday life".

Localized scalability can be looked at from the point of view of finding methods for avoiding intensive interaction with the user, by employing methodologies that help to decrease the similarity of users with services based on the distance between the users.

Just like other technologies, computing is becoming more and more ubiquitous. Five more trends have been identified for the technical feasibility of this change [35, 36]. The first trend is given by Moore's law [35], which talks about the fact that the number of transistors on a single chip doubles every 18 months - hence, chip prices will keep on falling. The second trend is about the emergence of new materials. Thirdly, communication technologies will be further advanced and used more widely. Progress on sensor network technologies and mobile ad-hoc networks dominates the fourth trend. The last trend focuses on the new concept of modelling infrastructures for smart spaces and smart objects in everyday life.

Even though the term ubiquitous computing was first coined by Weiser in the early 1990s, it is still not clear whether the first vision by Weiser has been achieved or not. Perhaps it is still in its early stage. This can be supported by the fact that most of existing research work and publications within the fields of ubiquitous/pervasive computing cite Weiser's initial paper as their main motivation of research and development. The existing work also thinks that the realization of Weiser's vision is still looking at the future of ubiquitous computing despite the fact that this vision was made about two decades ago. It can also be envisaged that the age of ubiquitous computing is here (at the moment), as we currently have computing without computers and utilise information processing in our everyday life. Mostly research in computer science is motivated or defined by a concept of technological problems, hence driven by building or elaborating on a body of past work. For example, user-centred computing is typically driven by end users' needs. The research on ubiquitous computing on the other hand is an area for a wider range of very different kinds of technological areas and interests. They are brought together by a common vision mainly driven by its possibilities for

future developments, needs and concerns for computational challenges and worlds, rather than the problems of the past.

If Weiser's vision of proximate future and our concept of his proximate future are the same, then a question to be asked is how can we prove that this vision has been materialised or is still being materialised? Well, it is possible that the ubiquitous computing vision can never come to pass, i.e. the proximate future analogy is a future indefinitely postponed. Consequently, ubiquitous computing is never about here and now. Alternatively, the vision has already come to pass, i.e. ubiquitous computing is already here. But it has taken a different form of mobile computation and pervasive mobile phones from that envisioned by Weiser. Devices with wireless data communication and powerful computational capabilities are ubiquitous in our society. Hence, "the future that ubiquitous computing has been attempting to build is not our own, but 1989s future - yesterday's tomorrows' [37].

A general methodology to address the problem of application-oriented research within the area of ubiquitous computing has been discussed [38]. Three main goals that need to be focused on have been identified as: theory, technology and scenario elements. Madhavapeddy [39] has identified that the field of ubiquitous computing research consists of people from different areas and knowledge zones, which include software, hardware and social engineers. It has also been pointed out that due to this diversity, a number of research areas have been identified such as context-awareness, sensor networks, low-power computing, activity inference and location sensing infrastructures. One of the main problems identified is the fact that the field is lacking a unifying determination that will drive this scattered research into real-world deployments. A decent market does exist for the media industry that could hugely benefit from the expertise of the ubiquitous computing research community. The problems created by the transition of media (mainly TV) from analogue to digital environments point out a golden opportunity for ubiquitous computing to get involved in creating an effective platform for the future of ubiquitous media for consumers. Hence, it was suggested that the ubiquitous computing research community should step forward and examine ways

of improving the handling of digital media for the next generation of home appliances. For example, RFID (Radio Frequency Identification) tags can be used to allow objects in a home, hospital or emergency environment to be detected automatically for interaction and efficiency purposes [40].

Laurent Ciarletta presented a position paper [41] that sees pervasive computing as a complex and user-centric research and development discipline. The main emphasis of the paper is on an "augmented emulation" as a core toolkit for the development and assessment of future work. Ciarletta defines ubiquitous computing as a convergence of four main computer science areas: Networking, Embedded Computing, Personal Computing and Computer-Human Interaction (with AI providing the needed context-awareness and automatic customization). The paper mainly suggests that the emulation serves as the key to the underlying engine and fuel for ubiquitous computing research. Ciarletta also points out that due to security concerns, ubiquitous computing is mainly confined to entertainment industry and security is the main factor that prevents it from entering a critical field of medicine. Another factor for such hindrance might be that it cannot be convincingly demonstrated to serious users. The paper proposes to follow the two parallel paths below to achieve demonstrable and evolved application oriented research and development:

- To use large-scale (both vertical and horizontal) emulations based on scenarios to test, demonstrate and emulate creativity for applications.
- In other to improve interaction between different communities and foster collaboration, there is a need to develop a simple model to place every work and concept where they belong and define good practice to integrate them within the emulation toolkit.

Hence, Ciarletta proposes the development of a large-scale community test bed and a framework known as PCBone (Pervasive Computing Bone), where everyone could participate and interact with other researchers in a real-time fashion.

Additionally, Salber [42] attempts to define the space of ubiquitous computing applications using dimensions of user mobility and interaction transparency, and identifies that increased interaction transparency can only be achieved via a breakthrough in the area of Human Computer Interaction (HCI) research. The paper also discusses two functional themes that are of importance across the area of ubiquitous computing systems – context awareness and automated capturing as well as integration and access. A clear agenda for HCI research in relation to ubiquitous computing is defined for each of the above two themes. A purpose of the paper is to provide a definition of what is ubiquitous computing. This has been achieved through examining two properties of ubiquitous computing first suggested by Weiser – mobility and transparency.

Whichever view the research community accepts or way they define ubiquitous computing, and to what extends this vision has been realised, the field of ubiquitous computing still remains an interesting research area with more challenges to be addressed. Compared to when it was initially coined, ubiquitous computing still presents research topics that will enable a promising market. Looking at the future of technological development, it can be concluded that perhaps ubiquitous computing can never fully achieve its potentials, and there will still be many exciting ways and applications that need to be developed to help improve the quality of our lives. Hence, the vision made two decades ago has sky and human imagination as its limit.

In a nutshell, ubiquitous computing has three main aims: products are to be everywhere (by being portable), to be small, and to be context-aware (of their environments and users). These aspirations are aimed at giving users complete autonomy of movement, ability to define policies as well as freedom of interaction. Making computing ubiquitous will make computing more attractive to those who find the current way of interacting with computers and networks distant, foreign and not very inviting. Although the idea of everywhere and inside everything may sound rather intimidating at first, if used properly, it could help us with the

problem of 'information overload' in very special ways. However, cleaver methods and means of balancing the availability of this information and security need to be developed.

## 2.6 SUMMARY

We have introduced the history of computer networks in this chapter, looking at the historical development from the first network project carried out in the late 1960s to the current stage. Computer networks are still growing with no sign of stopping. Currently, we have hundreds of millions of computers, devices and ordinary day-to-day equipment (e.g. mobile phones) joined together to form the biggest cyber-society: Internet. The notion of ubiquitous computing was introduced as a prospective view about the future usage of computers, although it can also be argued that today we are in this stage of ubiquitous computing. Smaller and cheaper computer chips are enabling us to embed computing ability into any appliances from a piece of paper to our bodies. People's daily activities are now closely connected with computers and beneficially become ever more convenient. The future seems more on such integrations of humans and computers. However, the great features of ubiquitous computing inevitably expose its inherent vulnerabilities, particularly with regard to security and privacy. The chapter also reviews the issues in relation to the development of ubiquitous computing from its inception by Weiser, where the three main phases: the mainframe, PC (Personal Computing) and ubiquitous computing eras are presented. A ubiquitous network must be properly secured so that it can be relied upon.

In the next chapter, we will present the work related to network security by focusing more on MANets. As a defensive countermeasure, IM, context-awareness and user-centricity amongst other mechanisms will be introduced.

# CHAPTER THREE

## NETWORK SECURITY AND IDENTITY MANAGEMENT

In computer security, identity can be simply referred to as distinguishing one user from another or a device from another device. IM is an act of trying to manage those various identities in a more structured manner. The issue of IM has not always been what it is today. In the early stages of computer development it was more about an issue of a physical control to premises of computer resources. It was only in the mid-60s when time sharing systems like IBM TSS-360 and DEC TOPS-10/20 [25] were introduced, which made accesses or identity management become an issue. While in the 70s UNIX provided users with the ability or concept of giving away their access rights permanently to programs, this made it possible for other users to execute applications and access data which could have been inaccessible with those users' normal access rights. However, this provided a great new functionality but at the same time it introduced new security problems and more complex scenarios or problems. Hence, the new authentication and access management of user identities started to emerge. As a result, a single sign-on became important, and role-based access control (RBAC) was introduced. In this chapter, we briefly introduce these issues within the area of computer security, and examine in more detail the issues of IM and user centricity as well as the role of context-awareness in IM.

### 3.1 IDENTITY MANAGEMENT (IM)

Mobile users may make use of partial identities. Identity is defined as something that can be used to identify a particular person, device or entity. With regard to mobile devices they have fixed identifiers, which essentially provide mobile identities. Such identities take into account locations and user identities that enable users to enforce security and privacy. Partial identities on the other hand is defined as a set of personal attributes of a user, where the user can have several partial identities, e.g. his/her work address, home telephone number, etc. [43].

MANets at the first glance may not seem to be directly related to the issue of IM. IM normally gives the impression of a traditional client server structure, where users can establish a handshake with a server for authentication and other purposes. These technologies cause an enormous impact with implications for security such as packet forwarding and routing, network management etc., which are functions, carried out by all available nodes within the network.

However, we are mainly concerned with the issues of IM, context-awareness, user-centricity, privacy, and user anonymity in this research project. The information related to these issues can be used to track users' whereabouts, monitor their behaviour, collect information about them as well as incriminate individuals based on the location of the devices used in a crime as evidence against them. These can be achieved by building profiles of individuals from the partial identities used by them, which can be utilised to harm people's privacy. The Hype Cycle Graph of Identity and Authentication related technologies and trends as of September 2003 are illustrated in Figure 6. The Figure highlights some of the issues in terms of technology visibility and maturity from inception to the stage that it has been generally accepted.

Figure 6 Identity and Authentication [29]

The research of IM schemes is of crucial importance because of the underlying cause of identity thefts. Although current IM frameworks have provided techniques, methods and policies to securely handle identity information, there are still several vulnerabilities that need to be addressed [44]. Hence, it is an extremely difficult problem to solve or address. Most people's personal data is in literally thousands of systems, potentially accessed by tens or hundreds of thousands of people. The two major threats faced by users are identity theft and identity disclosure. Conversely, it is imperative to acknowledge that above and beyond the risks to individuals, government and business organizations have a newly emerging and substantial liability specifically for the theft of employee identities. These demonstrate the need of ways and frameworks to help in mitigating this threat. A few strategic maps have been proposed, and an example of such maps designed by the BSC designer is shown in Figure 7, while Figure 8 provides a graphical representation of the variety of ways on how personal information is stolen. As reported in the Computer Weekly Magazine [45], a top German Police Officer told the ISSE 2010 conference that fighting cybercrime will be the greatest challenge in years to come, identifying that computer fraud makes up 46% of the total crime, Internet crimes 23% and a 64% increase of phishing. "We expect a 70% growth in 2010 in the number of phishing cases, where access credentials and digital identities are stolen for criminal purposes," said Jürgen Maurer, a Vice-president of the German Federal Criminal Police Office.

Figure 7 Strategy Map Adapted from [46]

Figure 8 How Identity Information Is
Stolen Adapted from [46]

The field of mobile IM in MANets is still in its infancy. However, there are some research papers and other studies that have been published, looking at various aspects of the area as to be briefly analysed in the following paragraphs. This observation also holds true for many MANet security problems as highlighted in Figure 9. Cao et al [47] have provided a survey paper of various IM models and paradigms, where three forms of paradigms have been presented based on the core design principles: network centric, service centric and user centric paradigms. Also when considering different implementations of such paradigms, current IM protocols, standards and systems have been classified into three models – isolated, centralised and federated models. This is done based on functionality, identity storage methods, cross-domain support, user control and privacy protection.

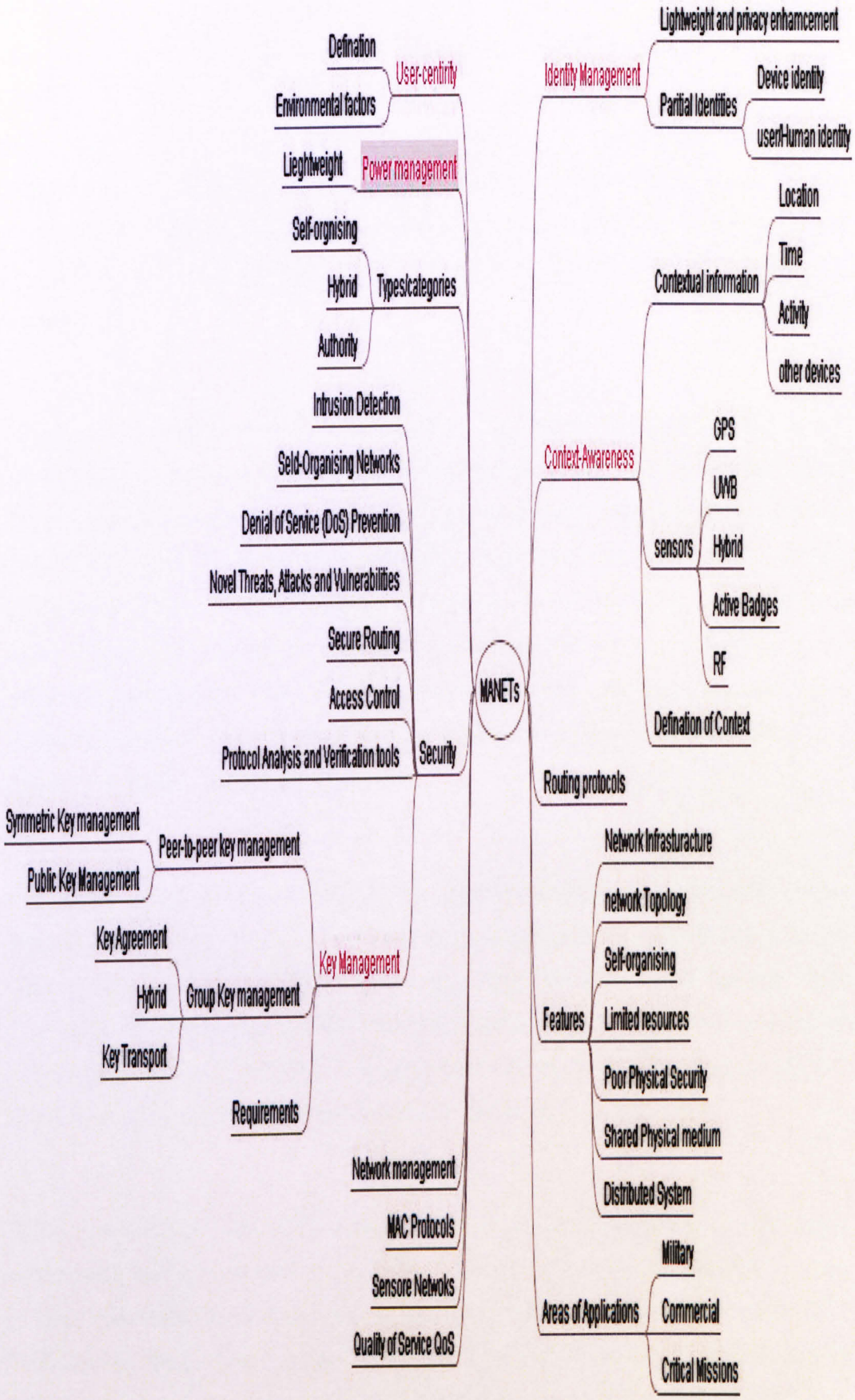


Figure 9 MANet Security Overview

Mobile devices including netbooks, tablets computers, mobile phones and PDAs are the main devices used in ad-hoc networks. It is worth pointing out that the manufacturers of such devices have done little in terms of informing users if and when their information is used by third parties, or when other third party applications are downloaded and how they use users' personal information. This same issue has been pointed out in the work of Enck et al [48]. Their research was focused on the vulnerability of third party applications deployed on Android devices.

The research findings include half of the 30 applications studied shared location information and unique identifiers with advertisers. They also reveal that 15 of these applications sent out location information without informing users that the data was shared. Some of the applications gathered and despatched location information even when they were not running normal operations for the users, and some of them also sent out the information updates every 30 seconds. Seven of the studied applications shared unique identifiers known as IMEI (International Mobile Equipment Identity) numbers, and others also shared users' personal phone numbers or serial card numbers.

Although an IMEI number is only used to identify a device and does not relate to a specific individual, it is still very useful information that when compromised might raise some security concerns. However, other numbers such as the ESN (Electronic Serial Numbers) and MEID (Mobile Equipment Identifiers) can link an individual to a phone. Usually, an International Mobile Subscriber Identity (IMSI) number stored on a SIM card can identify the subscriber on a network.

While some of the applications ask permission to gather information from users before installation, none of them informs the users of how the data would be used or who will share it. Hence, users have no control of such information after installing the applications – these imply that users blindly trust the applications. In some cases, applications have a legitimate reason of accessing users' sensitive and private data. However, it is of paramount importance that users have full control of

such data and its usage. Thus, there is a need to assure that the users' data will be used properly and they will be able to revoke the data usage.

It is not just Android devices that pass on user information to third parties, but also a far more controlled environment like Apple's IOS is another guilty party, as reported by the study performed by Eric Smith [49]. There is a historical background for the study. In 1999, Intel announced a Pentium III processor that contains a unique serial number per processor. The main problem with this is that it could be used to track users' online behaviour, and some governments even went as far as asking for a ban on Pentium III processors. Intel removed this serial number shortly afterwards. The Unique Device Identifier (UDID), which is found in iPhone, iPod Touch, and iPad, is something similar. Apple promotes the use of UDIDs as a way for application developers to link information to specific devices, e.g. storing high scores in a game on a central server. While Apple states that the UDID may not be linked to personally identifiable information, there is no mechanism in place to prevent this from happening, nor is there a mechanism to prevent the UDID from being shared with third parties (such as advertisement firms).

The study discussed above was carried out to "determine if the privacy fears surrounding the Pentium 3 have manifested themselves on the iPhone platform" [49]. Hence, the author studied 57 random popular applications from the App Store, and came out with two interesting conclusions:

- "We found that 68% of these applications were transmitting UDIDs to servers under the application vendor's control each time the application is launched. Furthermore, 18% of the applications tested encrypted their communications such that it was not clear what type of data was being shared". The study notes, "A scant 14% of the tested applications appear to be clean. We also confirmed that some applications are able to link the UDID to a real-world identity."

- "For example," the study continues, "Amazon's application communicates the logged-in user's real name in plain text, along with the UDID, permitting both Amazon.com and network eavesdroppers to easily match a phone's UDID with the name of the phone's owner. The CBS News application transmits both the UDID and the iPhone device's user-assigned name, which frequently contains the owner's real name."

The study [49] states that all these pose a real threat to iOS users. "Privacy and security advocates, personal iPhone owners, and corporate iPhone administrators should be concerned that it would be feasible - and technically, quite simple - for their browsing patterns, app usage, and physical locations collected and sold to unintended customers such as advertisers, spouses, divorce lawyers, debt collectors, or industrial spies". The study argues, "Since Apple has not provided a tool for end-users to delete application cookies or to block the visibility of the UDID to applications, iPhone owners are helpless in preventing their phones from leaking this information."

The above two examples have painted a gloomy picture of security worries in mobile applications and devices. Surely, as a matter of principle, devices should not discharge personal information or any information that can be linked to a user for everyone to see without users' consent. On the other hand, allowing very fine-grained control over these matters will only serve to confuse most users. This confusion could have two outcomes. One the one hand, users can see a complicated privacy dialog and automatically cancel out of fear. On the other hand, considering just how many applications use personal data, it could also lead to users becoming insensitive to such dialogs [50]. The insufficiency of transparency while exchanging people's identity and other information makes it hard or even impossible for users to participate in the protection process of their identities and personal information. In some applications or services that provide such facilities, most of the protection is done by the service providers rather than the users.

This issue is not limited to applications developed for mobile devices. As reported recently, Facebook had to take an action of banning developers that they caught for selling user names and contact lists. In the report, it is stated that developers trade user details to data brokers who use the information to target advertising more precisely. However, Facebook did confirm that the sale of user identities did not give access to other personal information. Hence, no private data has been sold or compromised [51].

For IM in MANets we are not only concerned with fixed identifiers, but also with other personal attributes of a user, as we are more interested in identifying and providing security to the user of the device rather than the device itself. *Mohammad et al* [52] categorised user identities as: personal identity (PID), corporate identity (CID), and social identity (SID). PIDs can be used to identify users in their very personal and commercial service interactions. CIDs and SIDs can be used in professional and social interpersonal interactions respectively. Additionally, a user's interests, preferences or tastes can be part of his/her identities as well, which may be dealt with by the user's SID. Some of these identities are very sensitive in nature, and therefore stronger authentication requirements have to be satisfied for their protection.

Privacy refers to the claim of individuals, groups or institutions to determine for themselves, when, how and to what extent information about them is communicated to others. IM can be looked at in different contexts with regard to privacy. One example is that it can refer to an integrated system of business processes, policies and technologies, which enables organizations to facilitate and control user access to critical online applications and resources while protecting confidential personal and business information from unauthorized users. In the case of mobile IM, location data may also include partial identities of the subject concerned.

Consequently, it can be seen that the issue of user privacy profiles is of crucial importance. Moreover, IM has only a chance to succeed if it is clear from the

beginning that users remain in control of their IM systems. The interoperability of such systems will not be accepted by the users unless it is, by default, controlled by the users themselves.

This feature makes MANets much more exposed and susceptible to security attacks, as the infrastructure is not fully defined and fixed. Thus there is a pressing need for cost-effective IM solutions in such environments. We can see that mobile IM is still in its infancy, where location information and users' personal preferences for the configuration of mobile devices and user interfaces present a complex research area and significant challenge to the research community.

As seen above, establishing identity management within a single entity has proved to be difficult and challenging. This takes another dimension when we consider establishing and implementing identity management within MANets and Systems-of-Systems (SoS) in ubiquitous environments. This also involves mutual contracts that have been setup amongst various systems and systems boundary. It also involves agreements such that an identity in one organisation is recognised by another identity within the Community of Interest (CoI) in SoS. This necessitates establishing a mapping among the different identities related to a user, which is called a single virtual identity domain [53]. Some of the challenges involved include: technology, security, regulatory requirements, privacy governance and legal issues.

Balasubramaniam et al [54] have also pointed out that establishing trust between identity providers, synchronization of such identities, agreeing on data ownership issues, minimising risks involved, compliance with regulations, preventing privacy violation and policy enforcements constitute some of the other challenges. However, as can be seen in chapters five and six, a successful implementation of identity management within MANets and SoS provides a number of benefits that outweigh the challenges above. Some of such benefits include: quicker development of systems–of-systems, minimisation of identity duplicates, diminishing of privacy and security violations, reducing data and profile misuse,

the ability of making use of partial identities, etc. However, in order to be able to succeed in doing so, we need to make the systems more user-centric in nature, so as to allow the systems to engage with users and be more adaptive to their needs.

### 3.2 User Centricity in MANets

The main aim of user centricity is to allow users to be in control of their applications as well as devices with the ability to present the information that is suitable for the users at any given time based on their conditions/policies. Hence, ubiquitous computing within MANet environments needs to provide the ability to empower users for human-computer interaction as well as embodiment of a sense of control from the users. ""Feeling in control" is a key concept in the context of the oncoming information society. People should be able to get the information they need (and deserve to get) in a convenient way. They should not feel intimidated or dominated by either the amount of information coming towards them, or by the technology (e.g. computers or computing devices) needed to get it" [55].

User centricity issues have been addressed by *Eap et al.[56]* by proposing an architecture based on a service-oriented framework called Personal IM that allows users to be in control of the management of their identities. However, the requirements of MANets have not been addressed by *Eap et al. Camenisch et al.* pointed out that user centricity is a significant concept in federated IM as it provides stronger user control and privacy [57].

*Bartolomeo et al.* have also considered a shift from a technical-centric approach of current IM solutions to a user-centric one. They have proposed a user profile and designed a distributed approach to manage user profile information and examine the possibilities for the choice of a unique user identifier [58]. The issue of user centricity has also been looked at from the point of view of its usage in Enterprise Directory Services to provide complete protection from a user's perspective. It has been suggested that combining public key infrastructures, user-centric IM and

Enterprise Directory Services would allow users to have control of their personal information stored within a directory as well as who is allowed to access the information [59]. Thus, a user may employ PKI to encrypt attributes, and then share decrypted details with selected entities. User-centric IM has also been examined in [53, 60, 61].

DAIDALOS (Designing Advanced network Interface for the Delivery and Administration of Location independent Optimised personal Services) is another project of interest to our proposed framework as it addresses some of the elements of context-awareness within pervasive environments. The project has over 40 partners, which ran between 2003 -2008 [62]. The pervasive computing element in DAIDALOS is addressed for supporting ordinary users within IM context [63].

### 3.3 CONTEXT-AWARENESS IN MANETS

The first question to be asked is: what is context? By the definition of the Oxford Dictionary, context is a circumstance in which something happens or in which something needs to be considered. However, many researchers in the research industry are not satisfied with such a general definition of context. As a result they have tried to come up with a more accurate one. Schilit et al. [64] claimed that three important aspects of context are: where you are, who you are with, and what resources are nearby. Context is also defined as any information that can be used to characterise the situation of entities (e.g. a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves [65]. Chen et al. [66] redefine context as a set of environmental states and settings that either determine an application's behaviour or in which an application event occurs and is interesting to the user.

One of the most essential aims of context-aware applications is to deliver contextual resources efficiently and effectively [67]. In today's real world, context-awareness is a key factor to the success of any ubiquitous application, which will enable conceptual data to be understood and communicated along with other entities in the system. *Verkasalo* proposed and developed a specialized algorithm

that can be used in conjunction with hand-held devices to acquire contextual information and classify them into home, office and on the move categories [68]. However, it is worth pointing out that by storing the information centrally, the problem of a single point of failure has not been addressed; hence the network can be affected by compromising one of the nodes that stores the data.

From the above definitions, contexts are typically referred to locations, identities and states of people, groups, and computational and physical objects.

Context is of paramount importance in ubiquitous computing devices, having to interact with users automatically with or without their direct control. Hence, context-aware applications should be insightful in respect to the users. They should react as the users expect them to. This requires the applications to collect and analyse contextual information. A possible taxonomy of this context information can be expressed by the following questions:

- Identity: Who is the user?
- Time: When did an action happen, or when should it happen?
- Activity: What did the user do, or what does he wants to do?
- Location: Where is the user?

Context-aware applications may use all these information to react automatically to input. However, the taxonomy needs to be developed in such a way that the notion of "feeling in control" is applicable. This will allow users to define or change their policies, status etc. on the fly, which will in turn reflect how the application is used.

Another question also needs to be addressed as to the reason of why context information is important in ubiquitous computing. Context reduces the input cost. Unambiguous input from users is also very expensive and time consuming. It

interrupts the users' thoughts and slows down the speed of the interaction, hence affects the efficiency of the application. By making the application able to sense the environment and interpret explicit actions, mobile devices could provide a rich and implicit context. This can further enhance the communication between humans and computing devices and make it much more efficient. Contextual information may provide an exciting user experience without much effort from the users. Users benefit through context sharing, if we assume that users and their friends have similar preferences, which means something that attracts one group member's attention has a higher probability of being preferred by the other group members. By sharing the context, the system could provide a better service.

There are some challenges involved when moving the interaction beyond the desktop to small portable devices, as this requires stirring the interface from humans vs. computers to humans vs. context-aware environments. A user may have multiple devices such as phones, tablet computers, etc. It is also possible that a device may be shared by different people. Therefore resolving the possibly conflicting input of different cooperating devices becomes more crucial than before in the design process. This makes it of paramount importance to have knowledge in the world. One of the goals of HCI design is to create a convenient user experience by better predicting the user's behaviour and understanding the delicacy of everyday activities.

It is also worth pointing out that ubiquitous activities are not so task-centric while the majority of usability techniques are. It is not at all clear how to apply task-centric techniques to informal everyday computing situations [69]. To address these requirements of ubiquitous computing, the notion of context in the constantly changing environment needs to be understood. Context can be classified into three categories instead of the narrow perspective that just focuses on locations. *Computing context* refers to the hardware configuration used, such as the processors available for a task, the devices accessible for user input and display, and the bandwidth. *User context* represents all the human factors, such as a user's profile, calendars, policies, involvements and profiles. *Physical context* on the

other hand refers to the information provided by a real world environment, which includes variables like locations, times, lighting, noise levels, etc. The three categories are equally crucial as the information can be used to determine the appropriate and customized interaction between the users and applications [70].

Context-awareness is also of crucial importance in IM because users within MANet environments would be able to access and view services or available participants based on contextual information. For example, a user who is within a city centre and looking for friends to socialise with can make his social identities available within a MANet and search for friends within such an environment. Hence, services and security settings can be tailored based on the contextual information provided. Context-aware services are often viewed as a generalisation of location-aware services, and subsequently many context-aware systems necessarily inherit the data management problems associated with any location-aware subsystems [71]. Context-awareness within the field of pervasive computing allows systems to adapt their operations based on the current contextual information with or without explicit user intervention and thus has the capability of increasing usability and effectiveness by taking contextual information into account. Putting this into the context of MANets, such networks might react specifically to the current location, time and other contextual attributes as well as adapting behaviour according to the changing circumstances as context data may change rapidly. With regard to user centricity and privacy, contextual information can play a crucial role in terms of allowing the policy maker to set privacy rules which depend on dynamic context data. For example, a policy can be set where access to specific information is either granted or restricted based on the location of a requesting user or device. Hence, context-awareness is one of the key elements for developing adaptive applications in ubiquitous environments and MANets in particular. The use of contextual information helps to provide personalised services to the users. The future will demand personalised search and contextual information results delivered to devices in accordance with users' mood or context [72-74].

*Dingldine et al.* [75] proposed anonymous communication for context-awareness issues for the traditional Internet. Some basic principles of cultivating identities on the web and the importance of personal identities to oneself as well as others in order for them to recognise one's contributions have also been examined [76]. Some location-centric isolation of misbehaving nodes in sensors networks has been proposed [77, 78]. This is of a data centric nature and suitable for use in energy constrained networks. Some versions of centralised trust based security systems, which deal with the specific needs and challenges of MANets by combining decentralised security management and context-aware computing capable of establishing an appropriate trust level for various situations, have been proposed by *Moloney et al.* [79].

Google has unveiled *Latitude* [80], a Google map for mobile and Internet applications, which allows users to report their approximate locations or different ones to friends. Other similar tools include FireEgale developed by Yahoo that helps users take their locations to the Web while giving them the ability to easily control how and where their locations are shared [81], and IYOUIT which allows users to share their information among friends where such information is automatically available to friends [82]. The tool (IYOUIT) allows users to opt-in so as to protect their privacy, and in addition users can also falsify their locations.

*Hadjiantonis et al.* [83] have presented a hybrid approach for MANet management that has considered some aspects of context-awareness and the capability of effectively managing a MANet. The issue of quality of context in pervasive context-aware systems for dealing with the complexity of context-specific operations such as acquisition, aggregation, reasoning and distribution has been addressed by *Sheikh et al.* [8], through the definition of five quality-of-context indicators for context-aware middleware. Recent publications have also looked at the issue of context awareness within the domain of publish-subscribe in MANets. The publish-subscribe paradigm has been extended with the ability to manage and exploit context information by using formal models for context-aware publish-subscribe implementations for MANets [84].

Additionally, *Verkasalo* proposed an algorithm for hand-held devices to acquire contextual information classified into home, office and on-the-move categories [68]. The algorithm classifies contextual data based on the usage of the devices involved. Users can also use their mobile devices for business purposes even when they are at home or on the move. *Chen et al.* proposed a paper-based leaning support environment where mobile phones, traditional textbooks and web-based forums are integrated to promote students' acquisition of knowledge. Students receive contextual messages from an online learning community based on their learning statuses [85].

*Chen et al.* [86] proposed a framework for supporting context aware environments in MANets. They make use of a virtual overlay network and two approaches (pull and push) to improve the efficiency of data delivery in MANets. The surrounding context of mobile nodes is used to determine which scheme, push-based or pull-based, is employed. In the framework nodes are divided into mobile context managers which are organized into an overlay segment-tree virtual network, and context providers/service requestors which send/receive contextual information via the segment-tree virtual network [87, 88]. Then, a push-based approach is used to handle real-time information and combine a pull-based approach for supporting context-aware environments in MANets.

Another application for Mobile Location-Aware Handheld Event is an event planner that has been proposed in [89], which provides a tourist guide service based on GPS location acquisition. Users may also use this system to send or receive emails. The system also allows the users to set up event reminders. Moreover, the system takes the privacy issue into account. A user could set a visibility based on persons or groups, which could help to control who can see whom. AccesSights is a multimodal user interface to support different user groups [90]. Its main objective is to support both blind users and sighted users at the same time with the same tourist content but for each user group in their preferred modality. Not only providing suggestions during the tour, the system also warns by

vocal output when some obstacles are in the way, which especially benefits blind visitors.

As seen above, context-awareness is one of the key elements for developing adaptive applications in ubiquitous environments and MANets in particular. Some of the research questions that need to be addressed in terms of context-awareness include the followings amongst others:

- What information should be sufficient to describe a mobile user's identity so as to present the current mobile situation and context (e.g. locations, personal information, general preferences set by users, temporal constrains etc.)?
- How should the data used to present a mobile user's identity be collected, i.e., which parties (e.g. network operators, law enforcement agencies, profile providers, and/or service providers) should be involved in addition to the mobile user?
- What technical standards need be imposed so as to obtain access to these different components of mobile identities?
- Would it be necessary to introduce group identities such as work, friends and private for easy privacy and policy management?
- What policies should be used in terms of negotiating the exchange of information on mobile identities?

### 3.4 Context-Aware Monitoring in System-of-Systems (SoS)

Context or situation awareness in SoS is of crucial importance, especially when we are dealing with crisis management and emergency situations. As a result, some research efforts have been directed to this area and have been considered by various projects. The e-SENSE project focuses on the notion of context in order to understand how to capture the appropriate context data to enable real applications [91]. The main focus of the project is on psychosocial aspects of context in order to

enhance context-awareness by considering the activities of users with mobile lifestyles.

The e-SENSE scenarios are based on Mood Based Services provided to users in a range of application areas to enable the variation of psychological, sociological and physical aspects of context. Another similar project, EMERGE [92], aims at investigating situation recognition approaches that enable dependable emergency assistance services based on unobtrusive sensors seamlessly integrated into daily living environments. By monitoring the behaviour patterns of assisted individuals, the system automatically reasons based on deviations in behaviour and assess the functional health status of assisted persons. If suspicious situations or upcoming trends are detected, the system provides stepwise assistance by checking with and notifying external helpers, *e.g.* a socio-medical service centre, which might then undertake further preventive measures or rescue missions.

Situation awareness within the domain of emergency medical dispatch (EMD) and the way systems can support it appropriately have been examined by Blandford *et al.* [93]. They conducted a study of situation awareness in a large ambulance service, and the study has revealed some of the issues encountered during the development and exploitation of situation awareness particularly among the more senior EMD operators (called allocators). The notion of a 'mental picture' as an outcome of situation awareness and the issue of how an awareness of the situation are developed and maintained dynamically, as well as the difficulties they face in doing so are described. One important finding of the ambulance control system identified by the study is the relatively routine behaviour, which occasionally intermingles with incidents that demand much higher levels of attention than usually expected, but that the routine work must still be completed as a matter of urgency and importance. Operators exhibit contrasting levels of situation awareness for these different kinds of incidents [93].

Another aspect of addressing situation awareness in emergency services has been researched by Craddock *et al.* [94] by utilizing the concept of Web mash-up. Their

Secure Situation Awareness (SSA) concept demonstrator integrates several technologies together into a single situation aware system by using both public and private information, such as GPS locations, maps (using Google Maps) and location information about emergency service equipment (*e.g.* fire engines from a fire service department). A EU FP7 project 'Innovative and Novel First Responders Applications' (Infra) [95] has also provided us with the basic layers of interoperability and communication needed for a successful implementation of identity management within SoS. The Infra project's main objectives focus on three main levels: providing first responders with reliable communication, interoperability of navigation systems based on three location sensors, and standardization of the framework of communications and applications. Our work aims to make use of these fundamental principles and add a layer that will provide users within SoS (or first responders as referred to in Infra) with the ability to manage not only their identity information but also any other information that is of crucial importance in terms of sharing information and identities with other users within SoS, by focusing mainly on the use of mobile devices.

A recent market survey for alternative positioning technologies in consumer devices, including mobile phones, portable computing devices, and cameras, reveals that it is set to explode over the next five years with revenues reaching more than $2.5 billion by 2015 as shown in Figure 10 [96]. The demand for low cost, ubiquitous location data has increased significantly over the decade. Companies such as Google, Microsoft, Apple, Nokia, Facebook and others are battling to enable and control consumer locations. This will bring new opportunities and threats (as hackers will try to break into the new developments and exploit devices) to the market but ultimately will drive a 400% [96] increase in alternative location technology penetration across a range of portable devices, location technologies and location providers.

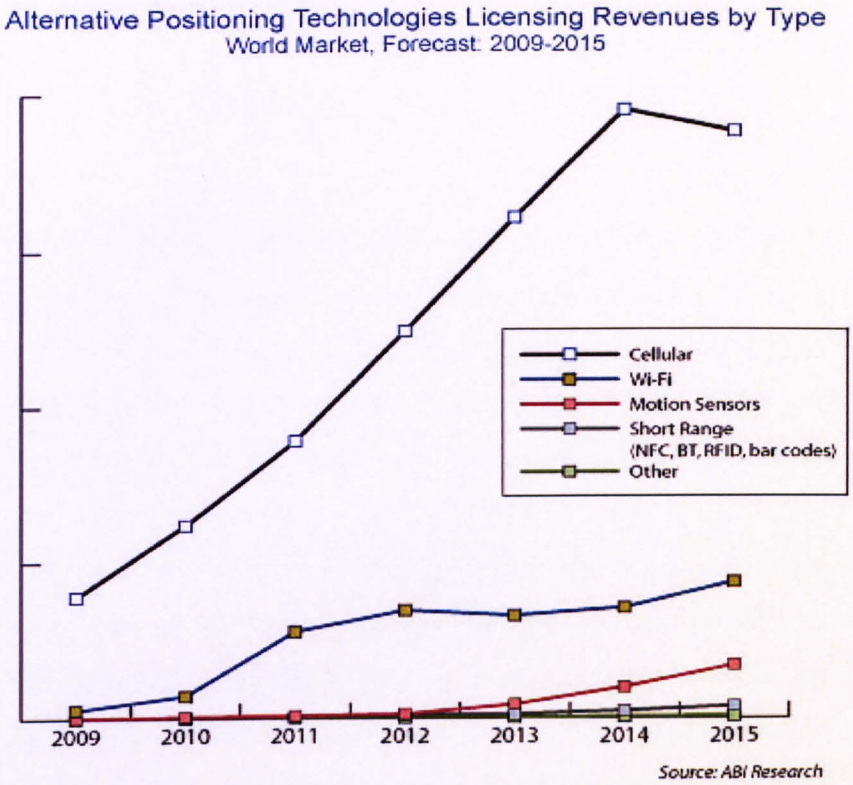


Figure 10 Positioning Technologies

### 3.5 SOCIAL IMPLICATIONS

In today's digital communication revolution, it is all about connecting people ubiquitously and bringing communities together globally. Hence, 'social networks' is the keyword for such a current trend. Different applications are developed to fulfil the needs of such growing demands. We have seen applications including Facebook and MySpace. However, in the past, researchers studying social networking or systems interaction have not addressed the utilisation of contextual information to enhance user experience. The adaptation of such applications mainly focused on establishing the most appropriate model with the potential to formulate and use contextual information, but did little or no effort in considering the security implications of such actions.

Even those applications with the consideration of security implication have not provided means of putting such policies under the control of users or handling the matters arising after the exchange of either partial information from users or

files/data shared with participants in an ad-hoc manner. These issues have been discussed previously in section 3.2, where Facebook, Android applications and iPhone applications allow developers to sell users' private data.

Additionally, item-based filtering algorithms for online recommender and advertising systems have been proposed by various researchers, but these algorithms are mainly focusing on the use of either k-nearest neighbours [97] or top-N item-to-item similarities recommender models [98]. Other well-known ranking algorithms include the Google pageRank algorithm [99], which is also extended for different purposes such as the SimRank [100] that computes the structural context similarities. The initial pageRank algorithm is aimed at ranking and displaying top relevant pages from an individual keyword search. However in the late 2009 things changed. A post appeared on Google's cooperate blog with a headline "Personalised search for everyone". Hence our assumption that when we search a term on Google we all get the same results becomes false. As Eli Pariser [101] pointed out in the book titled "The Filter Bubble: What the Internet is Hiding from You" that the result you get is what Google's algorithm thinks is best for you based on your location, previous search terms and internet viewings. This is done without the user knowing about it. However, another survey carried out by Click listeners [102] suggested that the claims made by Eli are inconclusive, as there has been little or no variation on the results obtained in the survey. Other social networking and web-based services have followed suit on trying to provide customised services for users intrusively.

Other recent work in this area mainly focuses on database-oriented [103], agent-oriented [104] and service-oriented approaches [105]. *Helmhout et al.* [106] have also proposed an instant knowledge platform in P2P environments that enables the privacy-conscious sharing of mobile social context information either automatically or manually at an enterprise-wide network of contacts [107]. As part of the Core 4 Research area: instant Knowledge of Mobile Virtual Centre of Excellence [108], the project aims to allow calendar entries on personal devices to be instantly harnessed to deliver real value to telecom operators. The project also

plans to develop a distributed algorithm to support machine learning, context-based, recommender engine for use within a user's personal network.

### 3.6 SUMMARY

This chapter has established and introduced the multiplicity of possible attacks involved in computer systems, with more focus on identity theft. The chapter has highlighted the major trends of technological development in terms of inception, maturity and general acceptance by users. We have also provided an overview of security issues in MANets, some mechanisms and impacts of contextual information and user centricity, and the ways in which they can be used to help improve security in MANet environments.

In the next chapter we will summarise the main limitations of the existing work discussed in this chapter and chapter two. We will also provide the motivations for our research project based on these identified limitations.

# CHAPTER FOUR

# LIMITATIONS OF CURRENT IDENETITY MANAGEMENT SCHEMES

Traditional identity management schemes were developed for wired networks. To our knowledge, there is no appropriate identity management framework yet, which has been particularly proposed to meet the special requirements of MANets normally involving devices with lower battery capacity and processing power. However, with the continuous development of MANets, especially the progress made on wireless ad-hoc networks and distributed systems, we believe that some existing solutions could be extended or some of their methodologies can be used for ubiquitous computing. In this chapter, we will summarise the limitations identified from the critical review of the existing work in the previous chapters two and three respectively. The chapter also introduces the main aims and objectives of the project and then the requirements of UCIM.

## 4.1 User Centricity

User centricity issues have been addressed in *Eap et al.* [56] , *Camenisch et al.* [57] and *Verkasalo* [68]. Other similar tools include Google *Latitude* [80], FireEgale [81], and IYOUIT [82]. *Chen et al* proposed a paper-based leaning support environment where mobile phones, traditional textbooks and web-based forums are integrated to promote students' acquisition of knowledge. Students receive contextual messages from an online learning community based on their learning statuses [85]. However, all the proposed work has the following limitations:

- Storing the information centrally can lead to the problem of a single point of failure and this has not been addressed. Hence the network can be affected by compromising one of the nodes that stores the data.
- Google *Latitude* [80] for mobile and Internet applications allows users to report their approximate locations or different ones to friends. However, users cannot specify who amongst their friends is able to see such information or to be able to stop them from misusing the information
- *Verkasalo* classifies contextual data based on the locations (i.e. Home,

office, etc) of the users as the usage type of the devices is involved. However, users can use their mobile devices for business purposes even when they are at home or on the move.

- *Chen et al* proposed a paper-based leaning support. The main issue with this is all students will be able to view the details. This would prevent some students from using the tool, as they might just want the lectures/tutors to view the details or just certain students. It might also be the case that it will distract other students from their work. For this reason, it is of paramount importance to be able to define polices to allow users to control such details.

### 4.2 Dynamic Policy

Policies are currently expressed as an integral part of the compose scripts within a provided solution. However, this can be clearly separated from the security policies in a system, so that the policies drive the corresponding analysis, rather than *vice versa.*

Separating the two would also allow us to construct and express policies in a way that would be easier for users to work with. The development of composed scripts is neither user-friendly nor intuitive. Hence, there is a need of a clear separation between policies and analysis, while allowing users to be able to define and modify policies to meet their situations. Other policy-related questions include whether context, trust, reputation and risks should have an effect on driving dynamic policies, as well as issues relating to policy interactions.

Some of the related work on policy creation discussed in chapter three includes: role based access control (RBAC) proposed by *David Ferraiolo and Rick Kuhn* [109], and the *Ferraiolo-Kuhn* model integrated with the framework developed by *Sandhu et al.* [110] to create a unified model for RBAC [111]. However, all such proposed work has the following limitations:

- The approach is mainly based on a central RBAC system, where context data is assumed to be globally accessible [112], and the static and

dynamic separation of duties makes up the most common constraints in RBAC.

- The summarized related work in section 3.6 based its solutions on fixed constraints or elements to compare against those not allowing the interaction of the systems with their users.
- Although many publications as seen above have examined the issue of policy specification, data filtering, etc, they have been done almost universally from a theoretical standpoint. Some relevant work with a more practical focus has grown out of the interest in web based retail applications to recommend products based on the content of users' items in their baskets or that they have viewed.

### 4.3 CONTEXT-AWARENESS

Some of the limitations of related work with regards to the utilisation of contextual information and context-awareness issues in MANets includes amongst others:

- The use of an overlay network in *Chen et al* [86] framework via the segment-tree virtual network [87, 88] creates another extra load on the processing power needed for dividing nodes into context providers, context servers, etc. Such nodes with designated functionality creates another security issue of what happens if a context server node has been compromised, or is out of the range. This issue makes the framework not suitable for the requirements of a MANet ubiquitous environment, as it is trying to mimic a central server process.
- There is a possibility of a single point of failure in the *Chen et al [86]* proposed framework as a results of dividing nodes into mobile context managers, context providers, etc. Also if the node(s) performing a specific task is compromised, there will be some security effect within the network itself.

### 4.4 AIMS AND OBJECTIVES

The earlier discussions in chapters 2 and 3 as well as the sections 4.1 - 4.3 of this chapter indicate that the evolution towards ubiquitous computing demands a new

generation of resource-efficient identity management frameworks. Hence, the main focus of this project is on the interoperability among different IM techniques and the development of an IM framework for fulfilling such interoperability in MANets. As outlined earlier in Section 1.2, the framework should be efficient, user-centred, context-aware and lightweight so as to meet the specific needs of MANets. It should also be able to manage multiple identities for different MANets situations, to perform negotiations with other peers about necessary identity information to be used for identification in relation to given security policy settings and situation awareness. The framework should be able to provide users with friendly control over their own identity information and security in terms of mobility across different MANets. Moreover, it should be able to separate users from the complexity of the technical implementation and operation issues of IM in MANets while allowing users to focus on policy aspects of IM. This ability is essential, as users typically do not want to learn detailed security techniques.

The aim of the project is to design such a framework, which is able to minimize the use of system resources such as energy consumptions and communication overheads, make use of relevant contextual information and put the users of the system in full control of their identity information and resources. It should have an appropriate system architecture that is flexible and scalable.

More specifically, the following main objectives are set out for this project:

- To provide a background to ubiquitous computing and demonstrate the unfitness of existing identity management frameworks when applying them to ubiquitous computing environments, particularly MANets.
- To specify a set of requirements for the proposed IM framework through a detailed analysis of the needs, characteristics and application scenarios of IM in MANets.
- To propose an original set of mechanisms, strategies and protocols that together achieve resource-efficiency in identity management.
- To propose a set of mechanisms, methods and strategies that together will

constitute and establish the IM framework, with respect to the requirements specified.

- To prototype the framework in order to provide a proof-of-concept for the proposed work and perform an assessment in relation to the proposed requirements, where possible.
- To apply a simulation technique and its related tool such as MATTS, NS2, GTNets or Matlab to produce a simulation of the framework.
- To perform an assessment of the simulation based on case studies to demonstrate the effectiveness of the proposed framework in terms of the expected characteristics of privacy, user centricity, context-awareness, lightweight and user friendliness.

## 4.5 Requirements

In this section, an analysis of the requirements for our proposed framework is presented. The analysis has been undertaken from three perspectives: anonymous communication, identity management, and usability requirements. The requirement analysis is expected to cater for the needs of end users and the key functionalities of the framework such as operation, mobility, security and personalization. The requirements can help answer two questions:

1. What partial identities should be used in certain situations?
2. Shall requested data be delivered in a specific situation to a particular requester, and what data is communicated if the delivery is permitted?

### 4.5.1 Requirements for Anonymous Communication

Anonymous communication is needed as a tool to protect a user's privacy against certain adversaries [43]. It is worth pointing out that most of the current anonymous communication mechanisms in use today are developed mainly for wired networks, whereby ad-hoc networks have other questions or issues to be answered differently. This includes the following:

1. Can the existing mechanisms provide enough protection for ad-hoc users, meet the low energy requirements of devices involved, and offer good performance?
2. Are the mechanisms dynamic enough to meet the required mobility of MANets?
3. Can anonymity be possible for both large and small ad-hoc networks?
4. Is it possible to provide total anonymity for ad-hoc networks without the use of a fixed infrastructure?

The questions above lead to part of the solutions that we have provided in UCIM. These have been demonstrated in the chapters six and seven of the thesis, where we have provided detailed solutions and evaluations of UCIM.

Hence, the following requirements for an anonymous communication mechanism might enable us to address the above issues in a more constructive manner:

1. Scalability: Enables the mechanism to be dynamic enough to operate on different network topologies
2. Security and reliability: Provide security against well-known protocol attacks, while maintaining the quality of communication
3. Performance: Takes into account the issues of mobile devices' limited resources such as low battery and processing capabilities
4. Robustness to topology changes: Addresses the nature of dynamic topological changes in MANets to ensure the sustainability of security and performance
5. Independence of a fixed infrastructure: Makes the mechanism independent of any fixed infrastructure such as the Internet PKI
6. Privacy and trust: Ensure the authenticity, confidentiality and legitimate unlinkability of information transmitted.

### 4.5.2 Requirements for Identity Management

Managing identity information plays a very curial role within our proposed framework. The following set of requirements is essential for guaranteeing users that their identities are well protected:

1. Functionality: This includes handling and representing identities; having pseudonyms with specific properties and ability to recover real identities; enabling history management (i.e. storing and analysing communicated data or data flows); helping to identify which partial identity is used for what transactional context, when, where and how; allowing users to have control on their identities by choosing their required profile settings and preferences; and managing multiple identities of a user.
2. Interoperability: One major characteristic of MANets is the variety of devices, e.g. PDAs, smart phones and laptops, which need to communicate with each other or can be found within such a networking environment. Hence, the proposed framework should be able to handle and communicate with any of these devices effectively with little effort required from their users. The success of an IM system very much depends upon such ability to interoperate across a network of businesses, partners, and services regardless of the platforms, programming languages, or applications with which they are interacting. The framework should also be able to easily integrate into similar tools.
3. User-centricity: It means the system should only reveal identity information about a user with his/her consent. Security is a main concern of this system. It should protect the user against deception, and verify the identities of any parties who ask for the user information to ensure that it goes to the right place. In the user-centric approach, the user will decide and control the extent of his/her identity information to be transmitted. The system should disclose the least information needed for the user to gain requested services. By following these practices, the least possible damage can be ensured in the event of a breach. These are some of the requirements

employed to design a user-centric identity management system in The Laws of Identity [113].

### 4.5.3 User Requirements

1. The user to be able to adaptively control information usage and disclosure
2. Lightweight to be usable in energy and memory limited resource devices
3. Customisable in response to available contextual information
4. Location authentication - location authentication is of paramount importance with regard to security. Note that a device used in the network can be identified as being used at certain location, but this does not imply that the device owner is using it, unless the owner authenticates him/herself as the user.
5. Location determination: This helps to identify users at certain locations, and to allow them to set the profiles that fit their commitments and possibly the environment.
6. Security and privacy: To enhance security, users should be able to choose end-to-end data encryption. Unauthorised users should not be allowed to access, view, or modify identity information. With the growing awareness of privacy and the wish to protect it, users would be looking for more control over their privacy, in particular, what information is known about them and by whom. With an effective IM system, a user should be able to exert some control as to how much identity/data they want to release (which may include an approval for sending some particular identity attributes) as well as being able to retrieve data concerning the location of their identity data and who is able to access it. Users should also be able to stay anonymous while accessing some network services such as the network time protocol (NTP).

## 4.6 Summary

In summary the key weaknesses of MANets that need to be addressed include among others the followings:

- Mobile nodes in MANets can be used to track people and also monitor their behaviours without any mechanism that allows users to protect their privacy and anonymity.
- Most of the proposed frameworks analysed earlier are aimed at wired networks and not in compliance with the requirements of MANet environments.
- The contextual information considered is mainly based on locations, and there is a need to look beyond this to other contextual information such as times and commitments.
- The current IM has an insufficient privacy and security protection baseline.
- There is insufficient user control in IM, i.e., it is not user-centric, as users are required to trust providers.
- No universal or standardised approach has been adopted, because the current standards are only specific to some applications.

Therefore, it is clear that due to the characteristics of MANets, none of these proposed existing approaches can be directly used to meet all the requirements of MANets [114]. Even the solutions that are provided for peer-to-peer networks (p2p) such as Tarzan [115] and MorphMix [116] do not meet the requirements and characteristics of MANets. In response to the above challenge, our aim is to conduct research into the issue of IM frameworks for MANets, which should be user-centric, context-aware, lightweight and user-friendly. Any new framework developed should fully meet the needs of MANet environments.

The next chapter will capitalise on the identified limitations presented in this chapter and present the aims and objectives of our project for the development of our proposed UCIM framework. The chapter will also provide the overall design of the framework showing what modules or components are needed to create the framework. The chapter further looks into the details of each module by focussing

mainly on how they meet the specified requirements of the framework and how they overcome the limitations summarised in this chapter. Some desirable features such as portability, heterogeneity, lightweight, dynamic policy creation and context filtering are also explained in detail with regard to the proposed framework.

# CHAPTER FIVE

## UCIM DESIGN FOR MANets

During the study, we have identified several major weaknesses within (see Chapter 4) existing frameworks and solutions. For example, they are mainly aimed at wired networks, which do not meet the requirements of ubiquitous ad-hoc environments, neglect the important role of users, and are not lightweight. We have also specified our requirements [117] for IM in MANets and justified the need for new IM frameworks in section 5.2. The chapter provides detailed design of UCIM including each of the modules, protocols and other techniques used during the design with clear justifications.

### 5.1 PROPOSED FRAMEWORK

We assume that mobile nodes that joined a network have gone through a proper authentication process, and that they are aware of their locations through GPS, UWB, and Bluetooth or other means. Hence, these nodes are able to infer information about their neighbours. We refer to a node as a proxy node if it is the closest to the node concerned in terms of the distance between them. If two or more nodes are within the same distance, then the available energy levels of these nodes will be used to establish which node will serve as a proxy node. A proxy node is required in this case to help provide a more secure and energy efficient way of communication between the devices so as to provide better services.

Another security issue is about the possibility of users, after successfully authenticating themselves to the network, trying to change their profile information for malicious reasons. To tackle this problem, our framework will only allow a user to change parts of its profile (i.e. its office address, room number, extension number, etc) before joining the network but not afterwards. On the other hand, the user's profile information such as its name, date of birth, role, etc will be static, and the user will be prevented from changing such information. Users will only be allowed to switch between their available profiles based on their current

commitments. However, all the information about such profile types will be fixed while connected to the network.

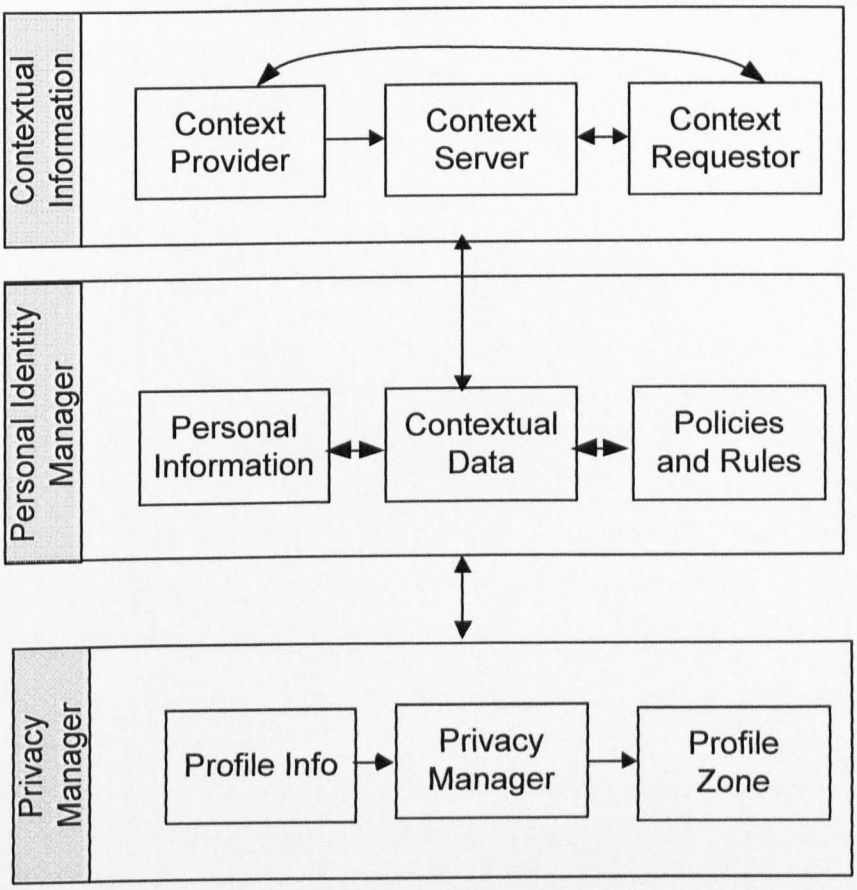


Figure 11 UCIM Framework

To address some of the weaknesses identified in chapters three and four respectively, we have proposed a new framework for User-centred and Context-aware Identity Management (UCIM) in MANet environments as shown in Figure 11. More specifically, we are considering Mobile Ad-hoc Networks and Crisis Management environments. The framework is designed to be context-aware, user-centred and lightweight. In trying to achieve the requirements of a lightweight framework, we have designed a novel hybrid metric that will calculate the energy level of each node in comparison to the required receiving and transmission energy and get a balance for deciding the means of transmitting and communicating between devices either directly or via the use of proxy nodes. We have also designed and used XML schemas to allow the information stored to use minimal

storage space and speed up processing and transmission of data. An algorithm has also been proposed that reduces information granularity without loss of integrity before transmission. The framework also allows users to be in full control of their identity information as well as designing policies that will be used in requesting information and communication between other devices. Users are also allowed to create and modify policies via the use of an interface and the underlying policy is created after selecting the required properties within the interface. These policies are also stored as XML files.

The modules shown in Figure 11 will be explained in the following sub-sections.

### 5.1.1 Contextual Information

The Context Provider (CP) is responsible for acquiring contextual information from various contextual sensors or providers. It is also responsible for processing contextual information into meaningful information that will be easily understood by non-technical users for its presentation within the user interface.

The Context Server (CS) stores the processed contextual information. After the contextual information has been acquired and processed, the information can be sent via the use of the pull approach to the context server. The main role of the context server is to store the information of a user and respond to the query of other users about the contextual information of the user/devices. It is also used to query other devices and store relevant information of the current devices for its own usage. From our literature review, it is evident that location information is widely studied and used as contextual information. We have studied social identity theory [118, 119] which is used to help identify other contextual information that is significant to users (see Figure 12).

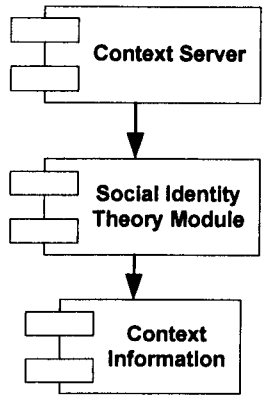


Figure 12 Relevant contextual information

The Context Requestor (CR) module is responsible for invoking queries to both CP and CS. It invokes queries to CP via the use of the push method, which provides a real time response to the query only within the device itself. While the query via CS is in the form of getting information about other devices within the MANet environment, the pull method is used.

### 5.1.2 Personal Identity Manager

The Personal Identity Manager consists of the user's personal information, processed information from the contextual information layer, and the set of policies and rules required for the application. As each user plays numerous roles in life, some of these identities are very receptive in nature.

The Personal Information module contains user details stored as an XML file. This structure is preferred to a normal conventional database because we are dealing with devices with limited resources.

The Contextual Data contains the processed data from the Contextual Information layer, where all relevant contextual information is identified by applying the social identity theory (Figure 12) and ready for usage by devices in a way that is comprehensible to users.

The Policy and Rules module deals with relevant security issues to protect the user's information from unauthorized access or disclosure. The user will be able to tick some boxes within a graphical user interface for the specification of the policies and rules. Such rules will also be depending on the contextual information that must be evaluated if personal data is requested. The rules can further deal with issues of access control and data abstraction.

### 5.1.3 PRIVACY MANAGER

The Privacy Manager module consists of the profile information, privacy manager decision module and the profile zoning. In the current solutions policies are expressed as an integral part of the scripts. Other policy-related questions include whether context, trust, reputation and risk should have an effect on driving dynamic policies, as well as issues relating to policy interactions.

The Privacy Manager module makes use of the Profile Information and the Contextual Data to decide what user information can be released or made available within the environment. This will classify the user's personal information into two groups: allowed or not allowed to be accessed by other users within the environment. This information is then passed on to the Profile Zone module. The interaction between devices within the MANet environment requires the user to be able to select part of his/her identity (partial identity) to be visible to other users based on the contextual information. We define this information as a set of attributes as to be detailed in section 5.4.1 PARTIAL IDENTITY.

## 5.2 USER CENTRICITY

Putting users at the forefront of systems within ubiquitous environments is an indispensable aspect for the adaptation and accomplishment of any development in such environments. Users within a ubiquitous environment always have the desire to be able to protect personal information as well as secure access to their resources. This can be achieved in two ways: determining who other users are and what they are allowed to do. This also refers to authentication and authorisation. In order to provide authorisation facilities for users, it is of paramount importance

to allow users to define policies using a simple interface, without the need of knowing the techniques underlying such policies. Policies are currently expressed as an integral part of the compose scripts within the provided solutions. Composed scripts are rules that are hard-coded into the main scribe as policies, giving users less ability to make changes to the policies. However, policies and composed scripts can be clearly separated from the system, so that the policy drives the analysis, rather than *vice versa*. Hence, in UCIM we have proposed and developed a system whereby users are able to define polices to control access to their resources. This is possible as users will be able to define who their users are, create their authorisation and access policies (using a dynamic policy creation interface) and establish agreement with other users on how such resources will be utilised.

### 5.2.1 SECURITY POLICIES

Separating the two would also allow us to construct and express policies in a way that would be easier for users to work with. Development of compose scripts is neither user-friendly nor intuitive. Hence, there is a need of a clear separation between policies and analysis, while allowing users to be able to define and modify policies to meet their situations. Other policy-related questions include whether context, trust, reputation and risk should have an effect on driving dynamic policies, as well as issues relating to policy interactions. Hence, we have provided an interface (a software component) as part of the Privacy Manager module that enables users to create and modify policies from a set of defined node properties that consist of user statues, locations, profile information, etc. Currently we have developed a tool that allows users to define such properties and store them as XML files to be used latter. To generalise this process, XML .NET is used to create a corresponding dialog without the need to rewrite code as new properties are introduced. Other issues include:

- Would it be necessary to introduce group identities such as work, friends, private, etc. for easy privacy and policy management?
- What policies should be used in terms of negotiation of exchange of mobile identity information?

- How to represent such policies in the interface and the underlying policy file?

A user may then have an option to decide under what circumstances the other nodes are considered as safe, or the opposite. For access control, users can decide the sensitivity levels of components based on their own application scenarios. In this case, they may have a list of access control policies to choose from, such as the Biba or Bell-LaPadula model. In both cases the policies could be recorded as a list of rules.

In UCIM we have represented such polices in XML files by using a binary tree structure, and the use of disjunctive and conjunctive representation of logical operators within policy creation and analysis. These also involve the exploration on how Lex and Yac [120] can influence such representations. We have explored the use of the JBoss Drools business rule management system [121] for policy rule specification and logic.

### 5.2.2 Context Based Access Control (CBAC)

In UCIM we apply the concept of RBAC but put users in control of their access policies and rules. This is achieved by integrating contextual information (mainly users' statuses/commitments) in the access decision process instead of using traditional RBAC. Hence, users' contextual information and roles have a significant and active influence on the access decision making module. Consequently, we refer to our improved approach as Contextual Based Access Control (CBAC), which takes context information, mainly the users' condition, as the deciding factor of either allowing users to access the required information or denying such access. We have further improved CBAC by designing and developing a dynamic policy creation interface that eliminates the problem of creating nested policy rules as well as users getting confused when they are required to make use of nested brackets while creating policies.

The main difference of our proposed CBAC from existing ones is the fact that the rules have totally eliminated the need of a role when accessing the information and have eliminated the problem of conjunctions and disjunctions. Other related works extend RBAC to include a new constraint of context. Hence, users still need to be assigned roles either by an administrator or request roles dynamically. The CBAC has a different set of rules and policies and it is designed and implemented with full control of system users.

In the case of conflicts between rules and policies, a Deny Override Policy will be applied, where if any of the rules results in a 'deny' response, the requesting user will not be allowed to perform the transaction and an unavailable message will be sent to the user. Each profile type is identified by its unique name, i.e. HomeProfile, OfficeProfile, or HealthProfile. Users are also allowed to define a new profile type. However, this depends on the situation that this framework is used. For example, when used for a social event then this can be permitted, but when we are dealing with crisis management situations of crucial importance then users are limited on what they can change or portray.

The structure of the user profile is a 3-tuple consisting of:

$$<DeviceId,\ Profiletype,\ pIDProfile>$$

which represents the device identification number, the profile type of the users of the device and the profile identification number currently in use by the user.

The profile information consists of attributes that users have selected to present their current profiles, e.g., if a user is at work, his/her profile attributes might comprise his/her office room number, extension number, calendar commitments, etc, as shown below:

$$Profileinfo = \{officeDetails,\ commitments,\ availability\}$$

To access such information we design a CBAC method for utilising the user's current status as contextual constraints in allowing other nodes to see and access relevant parts of the user's profile as depicted in Figure 13. A requestor first checks its table for the availability of the user ID that the requester is trying to request the profile from and the profile type of the user. If the user is available within the network, then using the distance metric, the requesting user will decide if it wants to make a direct or indirect request. A direct request will allow such request to be sent directly to a relevant user ID. An indirect request will require the use of a proxy node. Then the proxy node will deliver the message to the relevant user. When a proxy node needs to make an indirect request, it will first determine if the nodes in-between are considered secure to route the information to. This part is considered as part of network optimisation, by trying to find the most cost effective and secure way of coordination between nodes, so as not to disallow a function that might be of crucial importance, e.g. transferring patient data for an urgent medical usage. After receiving the request the node will check the required policy from the privacy manager. If the policy matches, profile details are dispatched either directly or via a proxy node to the requesting node, and the process is completed. Otherwise, some information will be fed back to the requesting node either directly or indirectly with a proposed modified policy or a deny reply. If it is a proposed modified policy the process starts again, and else the process ends.

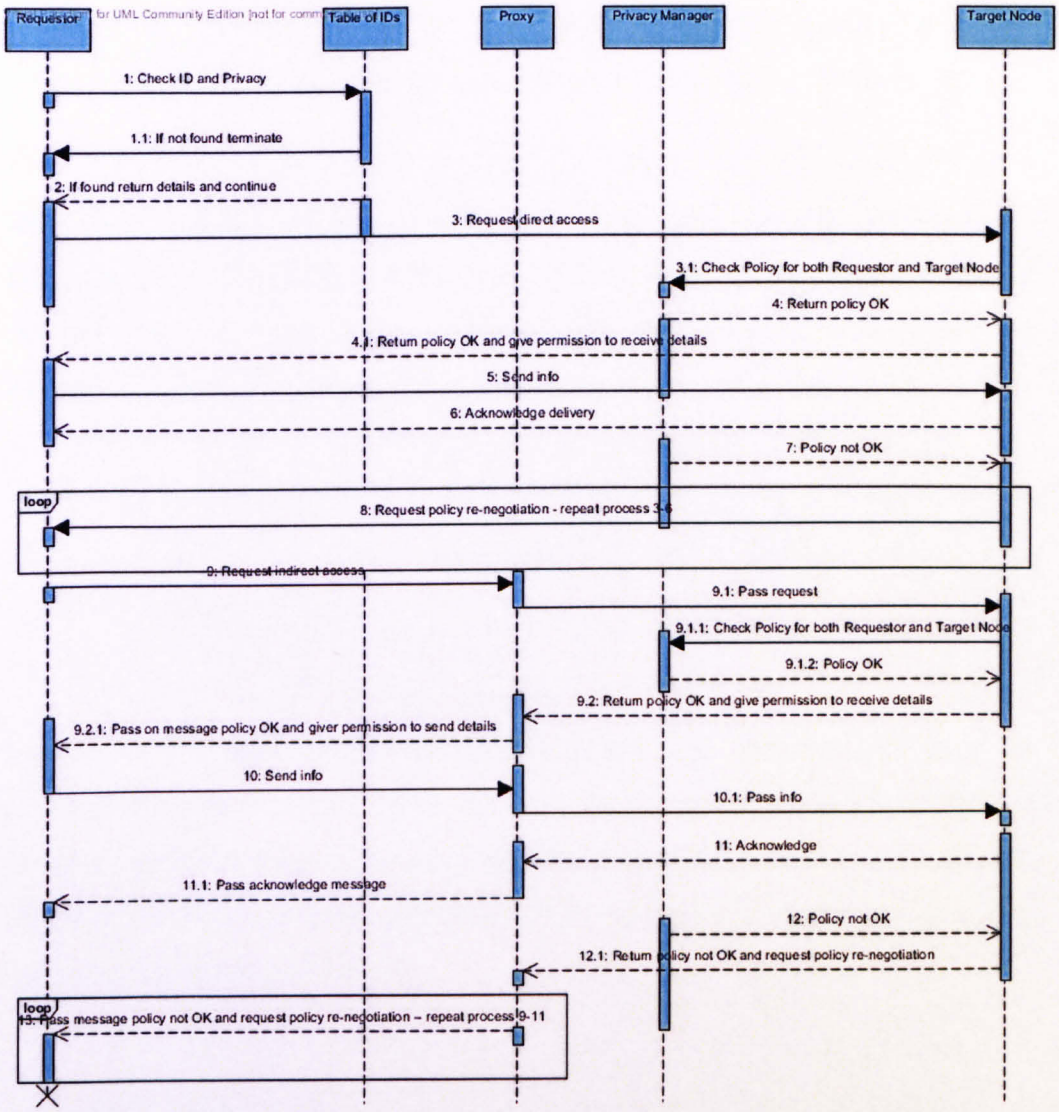


Figure 13 Privacy Policy Protocol Sequence Diagram

## 5.3 LIGHTWEIGHT

### 5.3.1 HYBRID EUCLIDEAN METRIC

When dealing with small ubiquitous devices it is of paramount importance to try to balance system resources such as CPU usage, network overhead, storage space, and energy consumption. When users within Community of Interest (CoI) need to share information, it is essential to find a cost-effective and secure means of exchanging such information between users. For example, if the distance between the users concerned is excessive, the transmission of shared data via a proxy node

will enable a used device to save energy. Hence, this hybrid metric is aimed to solve or minimise resources usage and improve efficiency.

In UCIM we have proposed a hybrid metric to measure these factors together for the dynamic determination of cost-effective identity management deployment. Another added feature is that a node's trustworthiness should also be considered in this hybrid metric to enhance the system's security policy. Specifically, it shows how a node can be utilized to save the computing-related energy; how a user profile can be managed in a distributed pattern to reduce the communication-related cost; and how a hybrid metric is used to balance both of them in order to extend the network lifetime.

Below, we describe a restrictive hybrid metric that measures the balance of system/node resources such as energy level, CPU usage and the distance between nodes to determine how to access or request contextual profiles from other nodes. The trustworthiness of nodes will be addressed as future work to meet the required node security policy and rules.

The metric will be based on two sets of energy metric calculations that are: the required communication related (both transmitting and receiving) energy metric and the node energy or battery level as explained below:

- The communication-related energy refers to the energy used by the radio transceiver of a node to communicate with others. This includes transmitted/received event records such as contextual information and request/transmission of user profiles. An approximation of energy consumption when transmitting or receiving $r$ bits between two nodes $n_1$ and $n_2$ with a distance of $d(n_1, n_2)$ is given in [122] as:

$$E_{tx} = (\alpha_{1,1} + \alpha_{1,2}\, d(n_1, n_2)^n)r \quad (1)$$

$$E_{rx} = \alpha_2 r \quad (2)$$

where $E_{tx}$ denotes the transmitting energy and $E_{rx}$ denotes the receiving energy. $\alpha_{1,1}$, $\alpha_{1,2}$ and $\alpha_2$ are constants, and their typical values are $\alpha_{1,1}$ = *45nJ/bit*, $\alpha_2$ = *135nJ/bit*, $\alpha_{1,2}$ = *10pJ/bit/m*$^2$ (for $n$ = *2*) or *0.001pJ/bit/m*$^4$ (for $n$ = *4*). $n$ is the reduction factor. Based on the results of the typical values obtained for $\alpha_{1,1}$, $\alpha_{1,2}$ and $\alpha_2$ by using $n$ = 2 and $n$ = 4 respectively, we choose $n$ = 2 in this proposed framework as it gives $\alpha_{1,2}$ a much greater value than *0pJ/bit/m*$^4$.

The computing-related energy refers to the energy used to implement the UCIM modules. It is used for the purpose of monitoring network statuses and user activities, executing algorithms, maintaining and updating user profiles. We assume this part of the consumption is proportional to the number of event records. That means the more user activities are observed, the more computing-related energy will be consumed. For each record, its processing is assumed to have a fixed charge $E_c$ = 5 *mJ* so as to have a uniform unit of measurement in performance level for our work.

- The energy of a node or its battery refers to the expected remaining lifetime of the node concerned in terms of its remaining battery level. For simplicity, we will model this metric with an initial fixed amount of battery level. This level is reduced whenever an activity that consumes energy occurs, i.e. processing, transmitting or receiving. Hence, this level is automatically updated, and when the level equals to a pre-set minimum, the full level of energy is therefore used as the replacement value.

$$E'_{n_1} = E_{n_1} - (l \times E_c + m \times E_{tx} + (1 - m) \times E_{rx}) \qquad (3)$$

In the above equation, $E_{n_1}$ represents the current battery level of node $n_1$ before the occurrence of the next transmitting or receiving activity $A$ including its related profile record processing, and $E'_{n_1}$ denotes the

updated battery level after the occurrence of $A$. The initial value of $E_{n_1}$ is set to the full battery level of the device used by $n_1$. When $E_{n_1}$ reaches a threshold of minimal energy, the battery should be fully recharged and $E_{n_1}$ is re-set to the full capacity.

Additionally, $l$ ($\geq 0$) in equation (3) is the number of profile records processed by $A$, and the energy consumed for processing each record is $E_c$ as stated earlier, i.e., the total energy consumption of record processing by $A$ is $l \times E_c$. $m$ ($\in \{0, 1\}$) signifies the mode of $A$ with $m = 1$ for transmitting or $m = 0$ for receiving. $E_{tx}$ and $E_{rx}$ are calculated using equations (1) and (2) for the energy consumptions of transmitting and receiving by $A$, respectively.

- Consequently, to determine whether the information should be requested directly by node $n_1$ or via an intermediate node (proxy node) $n_p$, a restrictive third metric $ED_h$ is used. This enables the node with the maximal energy as represented in equation (4) to be selected:

$$ED_h = Max(E_d, E_p) \quad (4)$$

Here, $E_d$ is the positive value of equation (5), which represents the energy level left after a direct transaction from $n_1$ to another $n_2$. $E_p$ is the positive value of equation (6), representing the energy level left after using proxy node $n_p$ for an indirect transaction from $n_1$ to $n_2$ via $n_p$.

$E_{dr}$ in equation 5 refers to the required energy for the direct transaction from $n_1$ to $n_2$, which is calculated using equation (1) defined earlier. $E_{pn}$ in equation 6 is the energy for the indirect transaction from $n_1$ to $n_p$ that then forwards the information to $n_2$, and $E_{pn}$ is also calculated using equation (1).

$$E_d = (E_{n1} - E_{dr}) \quad (5)$$

$$E_p = (E_{n1} - E_{pn}) \quad (6)$$

### 5.3.2 CONTEXT ONTOLOGY

As soon as contextual information is detected, the model needs to represent such information into an XML schema, which is integrated into an XML-based metadata. We decided to make use of XML models as a result of a survey by *Strang et al* [123] which demonstrates that XML models are suitable and meet the requirements of our framework. The decision was also based on the fact that one of the advantages of using XML schemas, compared to other ways of contextual representation analysed by *Strang et al*, is that XML provides a means of encapsulation and reuse of models (inheritance).

Figure 14 presents the defined XML schema for contextual information. We first start by creating a general schema that contains the base elements of all schemas, and then from there other schemas are created inheriting from the general schema as well as other schemas. The line(s) between the boxes that represent each of the individual schemas represents the inheritance. In Figure 14 both OfficeProfile and HelathProfiles inherit from the RootContext schema, ProfileContext schema and Status schema respectively. We adopted the notion of defining the schemas shown in Figure 14 to reduce the level of duplication in terms of defining users' profiles. For example, to define a new profile "SocialProfile", we can inherit the elements from RootContext and just add new properties for the new profile type. This not only eliminates repetition but also saves storage space, level of user interaction etc., hence makes the system more portable, lightweight and efficient.

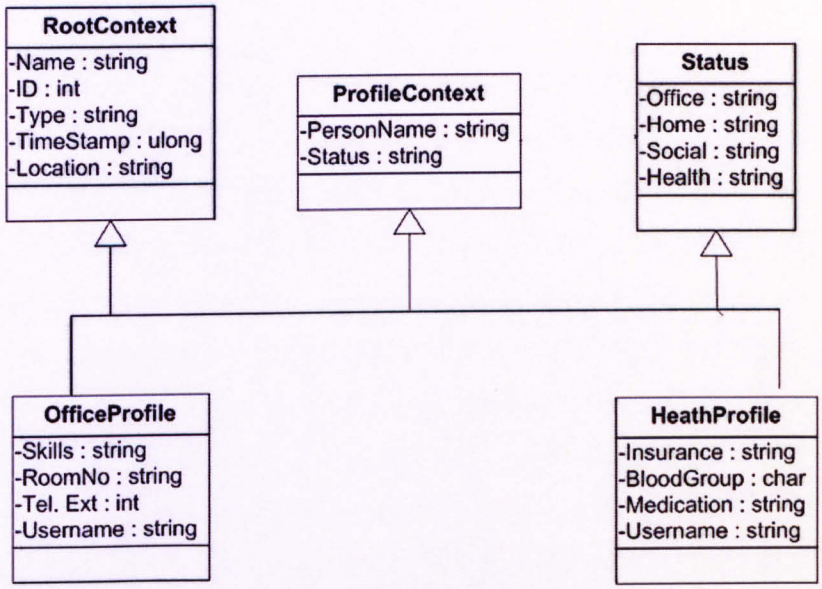


Figure 14 XML Schemas

```
//set of profile types
profileType = {Office, Social, Home, HealtCare}
ActionType = {triggerCommunication, makeAppointment, bookTable}
//relevant profile details
pIDProfile_office
          = { Skills, commitments, phoneNumber, room, calendar }
pIDProfile_Home = {address, availability, phoneNumber }
pIDProfile_Social = {Hobbies, phoneNumber}
pIDProfile_HealtCare = {insurance, bloodGroup, medications, history}
```

The above just show an example of elements that can be included in profileType, ActionType etc. Other things can also be included as and when needed.

### 5.3.3 Information Granularity

In a context aware system, it is possible that the sharing of information between participating parties is of paramount importance for fulfilling a given common task to achieve an objective between users of Community of Interest (CoI). The level of information shared or sent at any given time should be as minimal as possible for many reasons such as: limiting the bandwidth during transmission, particularly when the devices involved are small, reducing the amount of information that unauthorised users might intrude, and preventing users from building a whole profile of a user.

For these reasons, to reduce the amount of information sent and received within the network, we are proposing an algorithm for the flow of events depicted in Figure 15 to be implemented in the Privacy Manager module for adjusting the granularity of information before sending. The process is initiated after receiving a query where a relevant policy will be checked. If the request is successful then this is passed on to the algorithm used to reduce the amount of information before sending. The 'Adjust Required Information (Granularity)' process is triggered to allow users to be able to reduce the level of information that is to be sent to the node involved. The user will be given the option and an interface showing the current information available to other nodes. The user will then decide if he/she wants to reduce such information by providing a tick to the appropriate boxes that represent the user's various information.

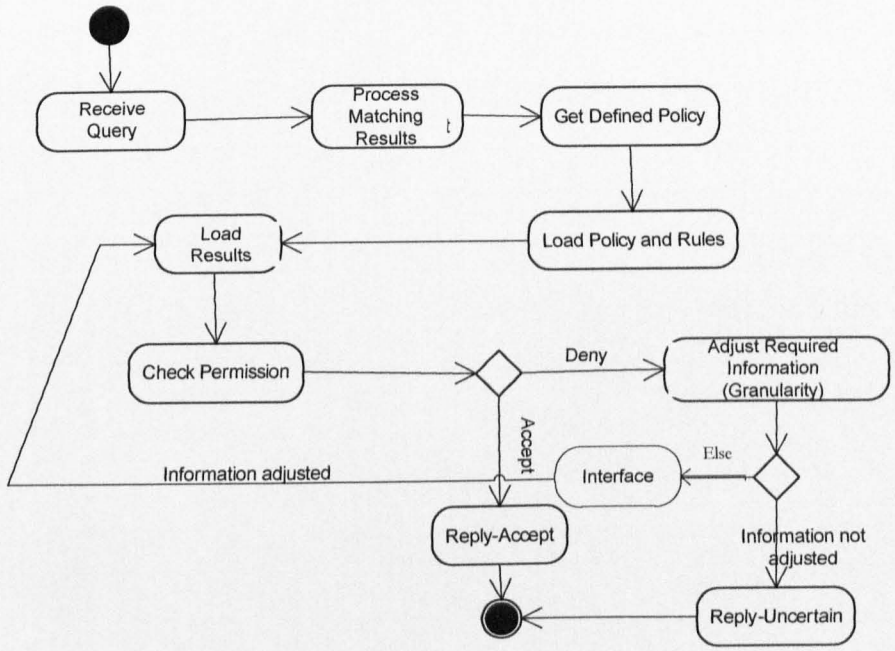


Figure 15 Information Granularity

### 5.6 DYNAMIC POLICY SPECIFICATION

Currently policies are defined or specified as an integral part of the system either hard coded as a programming tool into the system or designed in a way that users

will be able to interact with the policy environment, without the need of having a superficially technical knowledge or understanding. Hence, there is a need for policy-based systems to have policy rules defined using declarative policy languages that are distinct from actual system programming languages. A key advantage of using declarative languages to express policies is that the defined policies are more suitable for humans to view and edit. In addition, by separating the logic (i.e. policy rules) from the control (i.e. program implementations) of the systems, policy-based security and privacy protection systems are typically more flexible and adaptable than other non-policy-based systems. Also by creating a suitable user-friendly interface, the use of policies and user control will be much more friendly and suitable for humans, especially when using small portable devices in dynamic ubiquitous environments.

It is also important to emphasise users' desire privacy protection because they are concerned about the possible misuse of their information. However, this does not imply that they desire all of their information to be hidden from context-aware systems or that they want to know all the technical details required in setting up the required policies for their data usage. By completely prohibiting context-aware systems to share information, it will get in the way of the systems' ability to provide relevant services and information. The same principles apply, when editing or adding new policies to the systems becomes very complicated. This will deter potential users to adapt the usage of the systems. Hence, the reason above forms the basis of our motivation to provide a balanced system that is dynamic and adaptive.

Our concept of dynamic policy specification is motivated by the work of *Janicke et al.* [124], a case study on NHS's policy specification [125], and the Cassandra access control policy [126]. By policy we mean a set of rules that can be used to determine if a given query will be materialized or not. The policy consists of a head and body. Initially to deduce the head of the rule, all the body predicates must be deducible in such a way that the constraint is also satisfied. The sequence of predicates on the right hand side of the arrow is the body of the rule, while the

head of the rule is on the left hand side of the arrow. Mathematically, a policy $p$ can be expressed as $p\ \mathrm{Re}q \leftarrow Dec$ , where $\mathrm{Re}q$ is the set of access requests and $Dec$ is the set of decisions made from the requests. Decisions can either be *grant* permission or *deny* access permission. Queries can be in two forms, perform an action or request credentials or information, which are explained further below:

- Perform an action: the query corresponds to *permit* predicates/functions/procedures
- Request: the query corresponds to $can\mathrm{Re}ad$ predicates and then the information itself.

*permit* and *canRead* functions are defined as follows:

$$canReg(p, constraint)$$

$$permit\ (p, item)$$

In the case of *canReg*, this implies that the user with profile $p$ is allowed to make a request for information based on the constraints *constraint* specified. Function *permit* means that the user with profile $p$ is allowed or permitted to view or perform the action represented with the predicate *item*.

Each function represents a rule, where each rule consists of either conjunctions or disjunctions only. All the rules on the body of the policy need to be faired in order to deduce the head of the rule. Our policy specification makes use of dynamic literals as shown in the equations below using propositional logic so as to meet the demanding nature of ad-hoc environments rather than just hard coding the rules in the rule engine.

$$Dynamic\ Literals\ \langle L \rangle ::= A \pm \{P(\overline{X})\}$$

$$Dynamic\ Rules\ \langle R \rangle ::= A \leftarrow L$$

Notation: the notation of a line above a variable, e.g. $\overline{X}$, represents a (possibly empty) sequence of distinct variables $X_i ... X_n$. $P \in Pred, A \in Atom$. This definition is based on the first order function-free $\sum = (Const, Pred)$ with possibly infinite many constants *Const* and a set of finite predicates *Pred*. This generates a set of atoms that represents predicate names applied to equations of appropriate rules, denoted as $Atom(A)$. A policy $P$ is a finite set of rules $R$.

Using these connectives of propositional logic will allow us to define more complex predicates of our policy creation. These can be built from simpler ones in such a way that the truth or falsity of the compound in some binding depends only on the truth or falsity of the parts in the same binding.

We use predicates for a variety of purposes. Dynamic predicate names represent actions (or access requests), and dynamic rules define the conditions and state updates associated with actions, so dynamic rules are also called *action definitions*. Starting from the basic atomic objects defined here, we can utilize Z composite types, mainly Sets, Cartesian products and Schemas ([127] P.25), to define composite types as well as predicates. For example, the following rule:

$$DellInfo(u_2, f) \leftarrow Rang(u_1, u_2)$$

represents a simple example of deleting a file $f$ based on the function for the range between two devices, $u_1$ and $u_2$, where device $u_2$ already stores $f$. This rule can be further expanded to check if a user $u$ has a file $f$ using a function $has(u, f)$ with its result being true or false:

$$DellInfo(u_2, f) \leftarrow Rang(u_1, u_2), has(u_2, f)$$

An owner of profile information or any information can also specify who is allowed to read and modify the information, e.g.

$$can\,Read(u, f) \leftarrow \neg Sealed(u, f)$$

This implies that the user $u$ can read the file/information $f$ given that the information is not sealed by for the user.

In general, the function to seal information takes three parameters: the owner of information $x$, the person or group $y$ of individuals' devices that the information $f$ is sealed from. This function is denoted as *Seal*($x$, $y$, $f$). The + in front of the sealed function specifies the addition of the associated information to the existing list of sealed information – logical "and".

Additionally, another function *canSeal*($x$, $y$, $f$) is used to determine if a user $x$ can conceal the information represented in $f$ from another user $y$. This means that user $y$ will not be able to view the information $f$.

$$Seal(x,y,f) \leftarrow canSeal(x,y,f), +Sealed(y,f)$$

In more general terms, we can define a rule to seal information as:

*Seal*(*who, whom, what*)

where each of the three parameters

*who, whom* and *what*

can be a $n-tuple$ parameter, e.g.

$$who = (owner, area, group, user)$$

We define a projection function to take the required element from a $n-tuple$ to represent any of the parameters of the *Seal* policy. That is

$X_i^n$, *Where n is the number of elements in the tuple and i is the* $i^{th}$ *element in the tuple*

So in the above 4-tuple of $who, X_1^4 = owner$, where the *start and end* parameters define the validity period of the sealed item, either in terms of time or distance.

Generally speaking, we anticipate that any further dynamic rules can be derived from five special functions which are:

By making use of this process, we aim to make the policy creation more expressive. Also by using procedures in the body of the dynamic rules, we aim to eliminate the problem of policy expressiveness in terms of conjunctions and disjunctions. Therefore, each procedure can be further expressed or simplified by just having conjunctions or disjunctions (using 'and' or 'or').

Our proposed methodology puts a user's dynamically defined policies at the forefront and only displays Context Information (CI) that is relevant as well as allowing other CI that meets such policies to view the presence of the user. Consequently, these methods can not only accomplish the requirements of context-aware services provision with privacy in ubiquitous environments, but also deliver only suitable contextual information or services based on the user's policy settings, or even attempt to provide a means of user control policy creation and make it more expressive for users.

### 5.4.1 Partial Identity

With regard to mobile devices, they have fixed identifiers which are built into the devices for their identifications. Such identities take into account locations data or coordinates and user identities (can either be defined by users or fixed from network providers) which enable users to enforce the protection of security and privacy. In the identity management in MANets, we are not only concerned with fixed identifiers, but also with other personal attributes of a user, as we are more interested in identifying and providing security to the user of a device rather than the device itself. Partial identities on the other hand are defined as a set of personal attributes of a user, e.g. his/her work address, home telephone number etc. They can also be categorized as: personal identity (PID), corporate identity (CID), and social identity (SID) [52]. Hence we can firstly define a basic identity set ParitialID as $\{PID, CID, SID\}$.

Hence, an identity of a person can comprise of many partial identities that help to represent a specific context or role of a person. As stated by Pfitzmann [128], partial identities consist of subsets of attributes of a complete identity. Looking at it from a technical point of view, these attributes are data. Hence, a *pseudonym* can be defined to be an identifier for a partial identity. Whereas an "identity" can be assumed to uniquely characterizes an individual, a partial identity may not do, thereby enabling different degrees of anonymity [128].

The issue of users controlling their identity information and privacy plays a crucial role in the motivation of this research work. To address the issue, we adopted a strategy of making use of identities and partial identities to form part of a multi-factor authentication principle. As highlighted in Figure 16, we have divided users' identity and presented an identity of an entity "Ababa", where the identity of "Ababa" is the combination of a number of partial identities such as work, health care and social. This will enable Ababa to have control over which partial identity he should make available for other users based on his commitments and requirements. Individual partial identities and a device identity can be used for a multi-factor authentication of the device when joining a network.

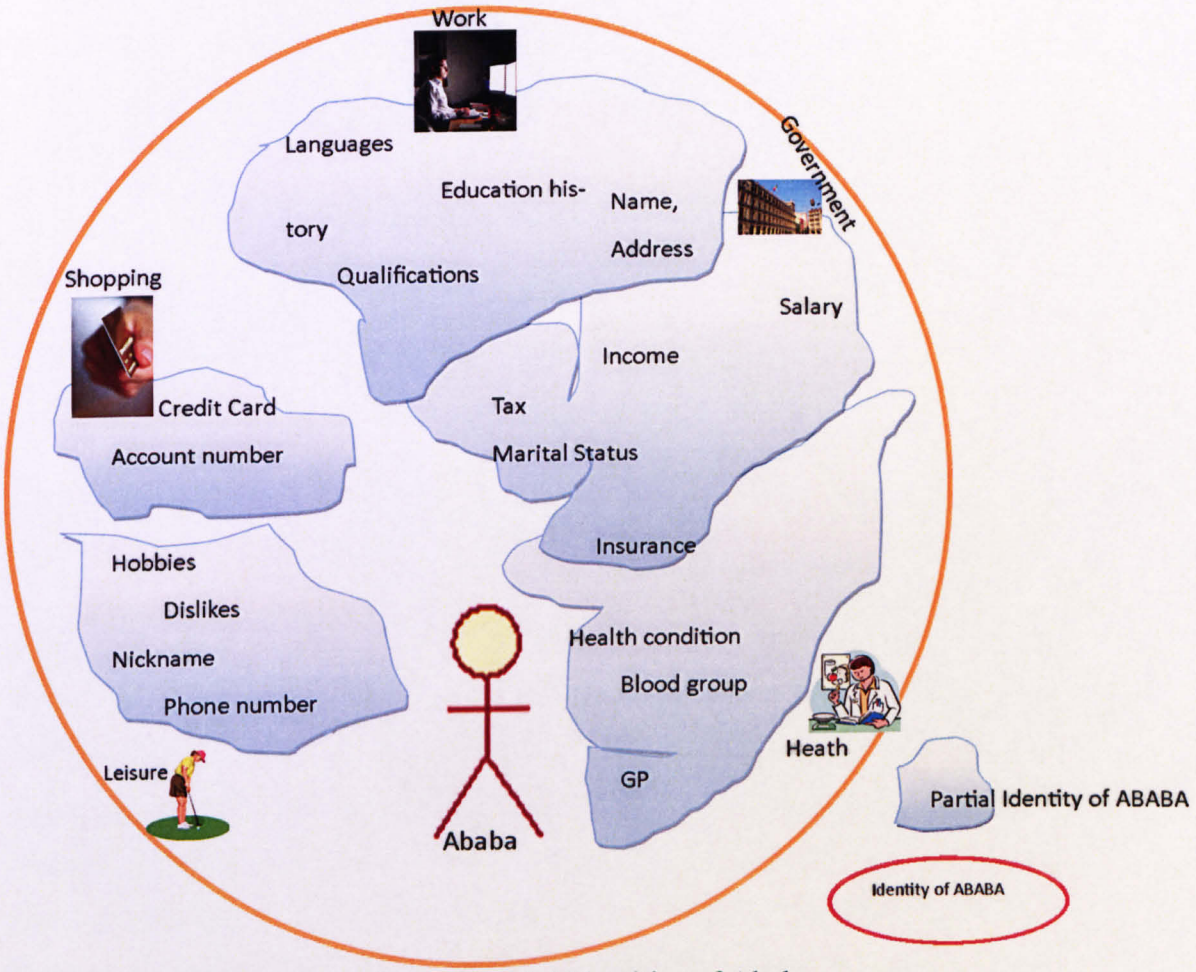


Figure 16 Identity and Partial Identities of Ababa

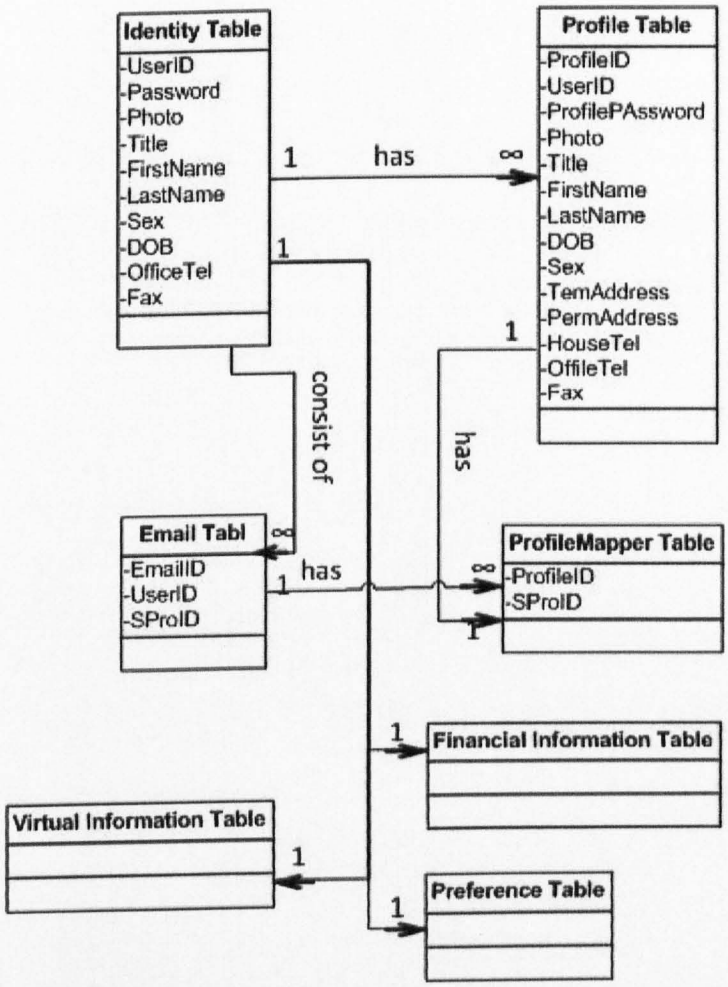


Figure 17 Identity Attributes

Although there is a trade-off involved in terms of the processing of a second identity for each device, the benefits outweigh the drawbacks as users of the MANet environment will be assured that they only hand over the information that is required for certain communication. This will help to eliminate the problem of identity theft and prevent the problem of impersonation attacks that are becoming a common problem in MANet environments [129]. Figure 17 further expands the attributes in Figure 16 as structured identity attributes organised into tables with relationships between them. Each of the tables defines a structure in a way that information/attributes disclosure will be kept to minimal and not propagated to the world. Each identity or partial identity can be mapped to multiple profiles; each profile table has a one to many relationships with the ProfileMapper table, while each identity table consists of several email tables. The ProfileMapper table can have and manage more than one profile type. This also allows each user identity in

the identity table to appear in more than one profile table, i.e. social profile, home profile, work profile etc.

Hence, user profiles are structured to contain various partial identities, and the structure is a 3-tuple consisting of:

$$(IDeviceId, profileType, pIDprofle)$$

The profile information consists of attributes that the user has selected to present its current profiles, e.g., if the user is at work, his/her profile attributes might comprise his/her office room number, extension number, calendar commitments, etc.

Here we define Profiletype as an element of a set of partial profiles that represents the identity of an individual:

$$profileType \in \{Home, Office, Public, Police\}$$

or $profileType$ is not limited to the elements of the set above, but users are able to define new profile types to suit their needs and to meet the conditions and needs of where their applications are deployed. For example, the element *Police* in the set above can be further divided into other sub-profiles, e.g.

$$\{Inspector, Constable, Sergent\}$$

To put this back into our earlier defined dynamic policy specification, we have $profileType \in P(\overline{X})$ . Hence the profile type or setting of each individual is factored into the elements that constitute the policy of either the device or data within the device. We have also made use of only a dynamic access policy without the inclusion of the profile type when needed as well as using *profileType* as the only element or constrains in a policy without the inclusion of other rules such as access policy – *canRead, view etc.*

## 5.5 CONTEXTRANK

Sharing information by dynamically and automatically discovering contextual information within ubiquitous environments can greatly enhance the usability and adaptableness of context-aware applications/systems. However, there is a trade-off between how much and how little contextual information should be shared. To be more specific, how much of such information should be displayed to the users within Community of Interest (CoI)? Many factors need to be addressed to determine how much contextual information should be available to each user within CoI. For example, the information may include the role which a user is playing within CoI, its device type (in terms of the screen size, processing power, security settings, etc), what information will be relevant to help the user achieve its objectives within CoI, what other information the user can share with others, in what ways such information will be displayed, etc.

In order to solve and answer the questions and concerns above, it is of overriding significance to comprehend the role of the users and give them the full influence on defining what possible contextual information will be able to help them within CoI and ubiquitous MANet environments. When this is achieved, users within CoI will make better use of contextual information and acclimatization of systems within ubiquitous environments. With this in mind, in UCIM we have presented an algorithm –ContextRank -that will help users define policies that will only display contextual information needed at any given time to help them fulfil their goal within CoI. The algorithm will also be able to suggest and predict possible contextual information for the users based on their defined policies.

The goal of the proposed ContextRank algorithm is to suggest new contextual information or to predict the utility of certain contextual information for a particular user based on the user's profile settings or defined policies.

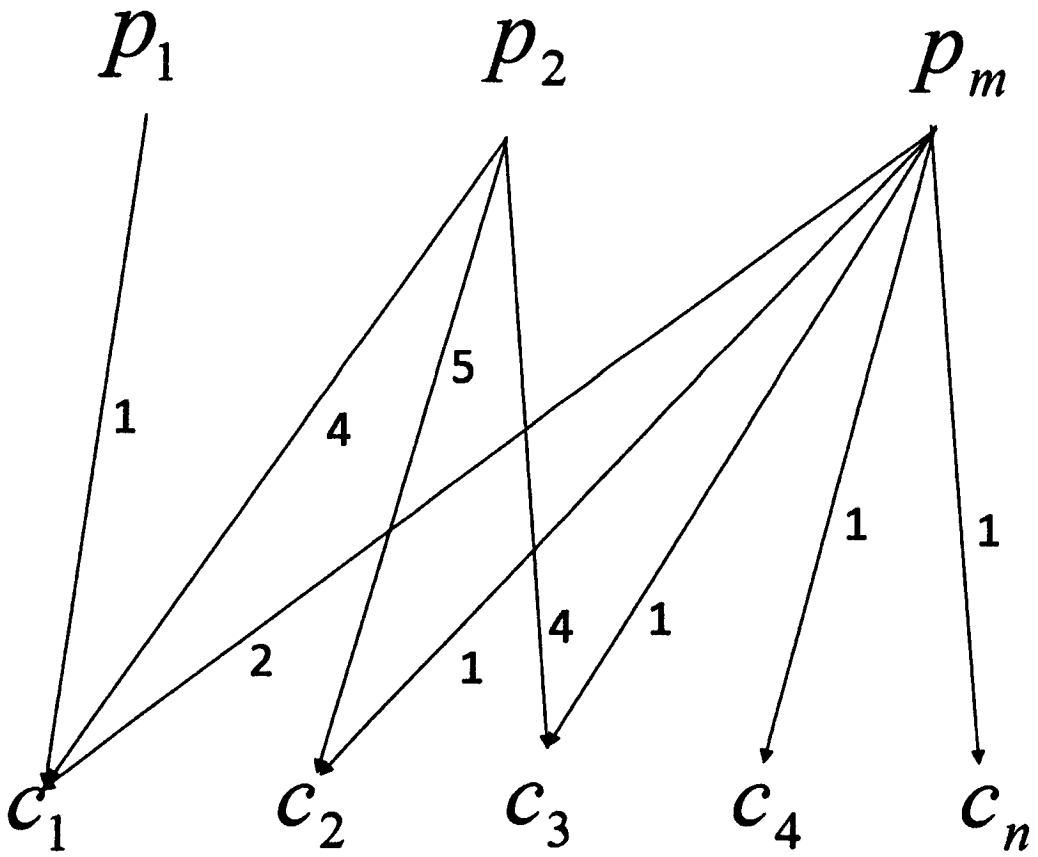


Figure 18 Profile Type and Context Information

In a typical CoI scenario, we suppose:

- There are a set of $m$ profile types, $P = \{ p_1, p_2, ... p_m \}$, and a set of $n$ contextual information items, $C = \{ c_1, c_2, ... c_n \}$.
- There is a set of $l$ users, $U = \{ u_1, u_2, ... u_l \}$, where each user $u_i$ can have more than one profile type $p_i$ .
- Each profile type $p_i$ has a set of contexts, $C_{p_i}$, which a given profile utilises or is associated with. Such usage or linkage is added up to a ranking score, generally within a numerical scale, by mining links with other contextual information and profiles. Note that $C_{p_i} \subseteq C$ (i.e. the set of the contextual information of profile type $p_i$ is a subset of the overall set of contextual information, $C$, and it is possible for $C_{p_i}$ to be a *null-set*.

Figure 18 represents an example of a situation where there are three users with three different profile types represented as $p_1$, $p_2$ and $p_m$ respectively, and the availability of five contextual information items, $c_1$, $c_2$, ..., $c_n$. The links with arrows represent the connections between the profile types and contextual information items.

The numbers of profile information and contextual information items can be different. Hence, the values shown in Figure 18 are not the same. For example, there can be 3 profile types and a set of 8 contextual information items. The number next to each line in Figure 18 represents how many times each profile type uses each of the contextual information items. For example, $p_1$ uses $c_1$ once, and $p_2$ uses $c_1$ 4 times, $c_2$ 5 times and $c_3$ 4 times.

The diagram in Figure 18 can also be represented by an adjacency matrix $A=\{a_{p,c}\}$ where each entry $a_{p,c}$ is 1 if an edge exists between user profile $p$ and context information $c$, and $a_{p,c}=0$ otherwise.

We define the degree of a context information item $c_i$ to be the number of edges linked to $c_i$, and denote it as $d_{c_i}$. Additionally, we use $L_{c_i}^{p_j}$ to signify the number of usages from a profile type $p_j$ to $c_i$.

For example, the degree $d_{c_i}$ of each context information item depicted in Figure 18 is listed in Table 1, e.g. $d_{c_2}$ = 2. Note that degree $d_{c_i}$ only counts the number of edges linked to context information item $c_i$. The number associated with an edge from profile type $p_j$ to $c_i$ is represented by $L_{c_i}^{p_j}$.

Table 1 Edge Weight

| | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_n$ |
|---|---|---|---|---|---|
| $d_{c_i}$ | 3 | 2 | 2 | 1 | 1 |

Moreover, the value of $L_{c_i}^{p_j}$ for each pair of profile type $p_j$ and context information $c_i$ in Figure 18 is given in the table below, e.g. $L_{c_3}^{p_2} = 4$:

Table 2 Relationship between $p$ and $c$

| | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_n$ |
|---|---|---|---|---|---|
| $p_1$ | 1 | 0 | 0 | 0 | 0 |
| $p_2$ | 4 | 5 | 4 | 0 | 0 |
| $p_m$ | 2 | 1 | 1 | 1 | 1 |

There are several conditions to be aware of when it comes to displaying contextual information in a given order within the system:

- New contextual information $c$ becomes available within the system - recommendation of the importance of such information should be based on similarities with the existing information and profile types.
- A new link is established from a profile $p$ and a contextual information item $c$, where both $p$ and $c$ already exist. This will elicit the increase in the importance of $c$ within the system to reflect its usage within the algorithm.
- New profile $p$ is introduced to the environment. The framework should be able to compare new $p$ with existing ones and suggest possible contextual information that will be useful to $p$.

- New $p$ and $c$ - the framework should use similarities between the information and profile in recommendation for usage.

From the above description, it is clear that our proposed ContextRank concept is inspired by the PageRank algorithm [99]. It will display the top contextual information relevant to a given profile type and its security settings.

We will define a method to predict the likelihood for associating a profile $p$ to a contextual information item $c$. The method will be able to predict the likelihood because we assumed that if a similar profile $p$ has used a similar context $c$, then there is the possibility that a newly introduced context similar to $c$ or profile similar to $p$ will associate itself with $p$ or $c$. This is also based on the social identity theory discussed in section 5.3.1 and presented in Figure 12. This method is defined to aid in recommending contextual information $c$ to the user of profile $p$.

The method requires that for a given new profile $p$, select the most used contextual information for $p$ from the set of all existing context items in $\{c_1, c_2, c_3, \ldots, c_n\}$. To do this, assume that there is a set of $k$ profiles, $P = \{p_{s1}, \ldots, p_{sk}\}$, each of which is similar to $p$. For every context item $c_j$ $(1 \leq j \leq n)$, to determine whether $c_j$ should be recommended to $p$, the following equation is used to calculate the average usage of $c_j$ by the similar profiles in $P$:

$$F_{p,c_j} = (\sum_{l=1}^{k} L_{c_j}^{p_{sl}}) / k \qquad (7)$$

Here, $L_{c_j}^{p_{sl}}$ is the number of usages associated with the link between profile $p_{sl}$ and context item $c_j$, If $F_{p,c_j}$ exceeds a pre-set threshold $t_c$ (i.e. $F_{p,c_j} > t_c$), $c_j$ is recommended to $p$.

A similar method can be used to decide whether a new context item $c$ should be recommended to an existing profile. Suppose that $C = \{c_{s1}, \ldots, c_{sw}\}$ denotes the set

of all context items similar to $c$. For each profile $p_i$ ($1 \leq i \leq m$), compute the following average usage of the context items in $C$ by $p_i$:

$$F'_{p_i,c} = (\sum_{l=1}^{w} L_{c_{s_l}}^{p_i}) / w \qquad (8)$$

If $F'_{p_i,c}$ is greater than a pre-set threshold $t_p$ (i.e. $F'_{p_i,c} > t_p$), $c$ is recommended to $p_i$.

For example, if a new profile $p$ is introduced to Figure 18, which is assumed to be only similar to two existing profiles $p_2$ and $p_m$ in the figure, i.e. $P = \{ p_2, p_m \}$, then we have the following calculations for context items $c_1$ and $c_4$:

$$F_{p,c_1} = (L_{c_1}^{p_2} + L_{c_1}^{p_m}) / 2 = (4+2)/2 = 3, \text{ and}$$

$$F_{p,c_4} = (L_{c_4}^{p_2} + L_{c_4}^{p_m}) / 2 = (0+1)/2 = 0.5.$$

If threshold $t_c$ is set to be 2, then only $c_1$ is recommended to $p$ due to $F_{p,c_1} > 2$, but $c_4$ is not suggested to $p$ because of $F_{p,c_4} < 2$. Similar calculations for the other three context items in the figure can be done to determine whether they should be recommended to $p$.

The above method is able to calculate the likelihood of associating a given profile type to a given context item because we assumed that if a profile has used a context item many times, then it is reasonable to suggest that such an information item is useful to the profile. Hence, if another profile similar to the one in consideration is introduced in the environment, we can recommend the same item to the new profile so as to help to improve the security of the system and reduce computation resources required to identify possible contextual information to use. The new profile can reject and not use the suggested contextual information with little or no impact to the security of the whole system.

## 5.6 SUMMARY

In this chapter, we have presented the proposed UCIM together with its modules and methods. Some of the significant features of UCIM that makes it unique and novel include: the use of contextual information, user-centricity, ability to define policies dynamically in an ad-hoc manner and the ability to filter contextual information using the defined policies. The next chapter will concentrate on providing the implementation results of UCIM based on lab tests, deployment of some of the modules in a real test bed and use of real devices with simulated scenarios that aim to represent real life events.

# CHAPTER SIX

## UCIM IN MANETS IMPLEMENTATION

Most of the frameworks for identity management strive to be general purpose and able to provide minimal protection for identity and most importantly lack the facility of allowing users to be able to manage their identities themselves. But all control is down to the application itself and they are mainly targeting fixed networks. This is clearly a very hard challenge, and as we presented earlier, it causes problems in ubiquitous computing. In this chapter, we introduce our proposed UCIM framework and simulation/Lab testing, an adaptive and resource-efficient identity management system with a novel policy specification, context-awareness and flexible user-centric design. By working together with user defined policies, it can reliably and effectively allow users to create partial identities, manage such identities and be able to minimise and prevent data misuse and data mishandling. This can further be enhanced by creating an expiry date for partial identities and monitoring whom such information is passed on to and how they can utilize it. UCIM is suitable for heterogeneous environments such as ubiquitous networks with devices known to have less processing power and capabilities. It has the following features: a reliable policy creation tool, a resource-efficient identity management scheme, a user-centric approach, and context-aware and flexible system architecture.

In this chapter we have presented different examples of implementations and simulation results. This is mainly focused on and explained via the use of motivational scenarios. Section 6.1 is concentrated on a scenario based on crisis management (section 6.1.1) in a controlled Community of Interest (CoI), where a level of pre-agreed policy is used. This also requires the use of an element of a centralised point of control as "the Gold Command" control room. Moreover, the role and impact of identity management in SoS is presented in section 6.1.3. Section 6.2 provides more details of the functionality of MATTS and its simulating environment. Section 6.3 mainly provides the implementations of our proposed ContextRank methodology. Sections 6.5 and 6.6 provide more details on

our dynamic policy modules and how our solutions are more user-centric and allow a dynamic, secure and user control policy definition. In section 6.5, we have demonstrated the dynamism and portability of the UCIM modules; this is done by integrating one of the UCIMs modules into another P2P framework for a secure home gateway.

Before going into the details of our implementation, we first present an overview of what our implementation has covered. We make reference to individual modules within the framework and how those modules are linked together in our implementation. To do so, we re-present our proposed framework in Figure 19 to highlight the modules that we have fully implemented and those that we have partially implemented or made assumptions of using other available sources of information for the modules.

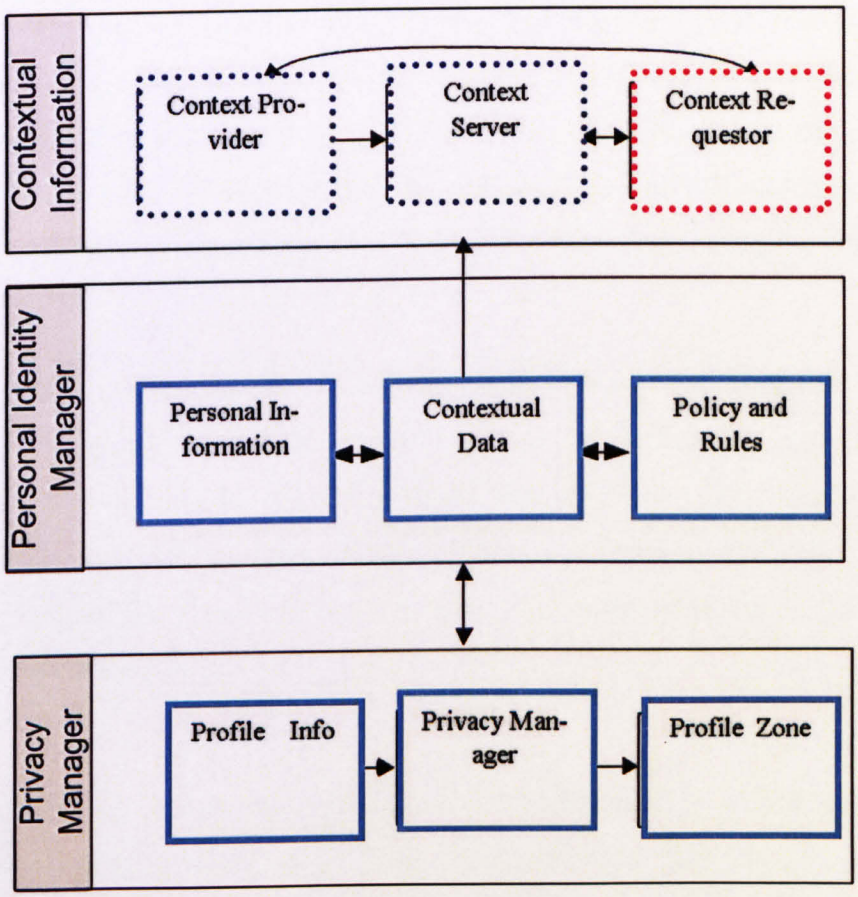


Figure 19 UCIM Implemented Modules

As shown in Figure 19, the modules with solid blue lines have been fully implemented, and those with doted blue lines have been partially implemented, while the one with doted red lines has not been directly implemented but the implementation in the simulator used provides the functionality of this module.

The context provider module has not been fully implemented. We first assumed that other available context sensing tools are available and the framework will be able to plug those tools. In our implementation we have only used contextual information within the simulated environment but have not acquired any other outside contextual information. The context information that we have used is the list of nodes/devices available within the simulating environments and the properties of such devices. The implementation and test results shown in sections 6.2 and 6.3 provide the proof of concept for this module.

For the context server module, in our current implementation we have stored the available context (list of devices/nodes) within the environment as a list. This is then stored within the server environment of MATTS (Mobile Agent Topology Test System, which will be introduced in sections 6.1.1 and 6.2). Other nodes can request such information when needed, especially when first joining the network.

The context requestor module has not been implemented as a standalone module but has been incorporated within the MATTS server environment. This allows a joining or any device to use the pull method to request information from the server. An example is shown on Figure 26 where if the user wants to send a file to other devices, then the list of all available nodes are first loaded in the form and the user selects from the available list.

The personal information module has been implemented by allowing users to define their profile information and that of the devices. This has been done by providing various interfaces such as the node property interface, of which full details are presented in section 6.4.3 and Figures 45, 46 and 47 respectively.

The contextual data module is implemented within MATTS, where the list of available nodes/devices are collected and stored; this can be used to send it to the devices when needed. Figures 27 and 28 provide further details.

The policy and rules module has been implemented within our proposed framework by providing interfaces that will enable users to perform such a function. Sections 6.4.3 on the policy interface and section 6.4.4 for policy logic provide further information and the proof of concept for this module. The results are also shown in Figures 48, 49, 50, 51, 52 and 53.

The profile information and privacy manager modules are implemented as part of the MATTS's server and contextRank simulation environments as presented in sections 6.4 and 6.3 respectively.

To demonstrate how the various modules within the framework are linked; we make use of scenarios such as the Boundary check problem in section 6.1.1, Identity Management in SoS in sections 6.1.2 and 6.1.3, and the home gateway scenario in section 6.5 as well as other scenarios within chapter 6.

To facilitate the understanding of our implementation, we first present an overall motivating solution to System-of-Systems (SoS) composition security problems as scenarios. We have also demonstrated two such problems with SoS that we have implemented. The rest of the implementation follows on from this motivating scenario and where appropriate we also present a simpler scenario for some of the modules of UCIM in order for the reader to be able to follow through the reasons behind such implementations.

## 6.1 SCENARIOS

In this section we will present some motivating scenarios used in the implementation of our framework. This is done so as to help the reader to follow through our implementation and to make things clearer.

### 6.1.1 SoS COMPOSITION SCENARIO (CRISIS MANAGEMENT)

As an illustrative example, consider the following fictional scenario – during a public event, multiple agencies and organisations who form a community of interest (CoI) must work together to ensure that the event proceeds smoothly, especially if an emergency situation develops. In our example, the set of emergency services and organisations shown in Figure 20 have formed a composed system in order to easily share data for the purpose of managing an event. Due to their differing areas of responsibility, the different parties will have different security standards. However, for the purposes of managing a public event, data collected by each party may be useful to the others for identifying malicious activities and to help maintain public order. As a result, each party within CoI will need to protect their personal information, to be able to manage their identities and may obtain temporary access to certain areas of the other parties' database systems. This type of system presents an ideal example for use in demonstrating various scenarios such as identity management issues, data transfer between organisations, data mishandling issues, profile building and boundary checking principles.

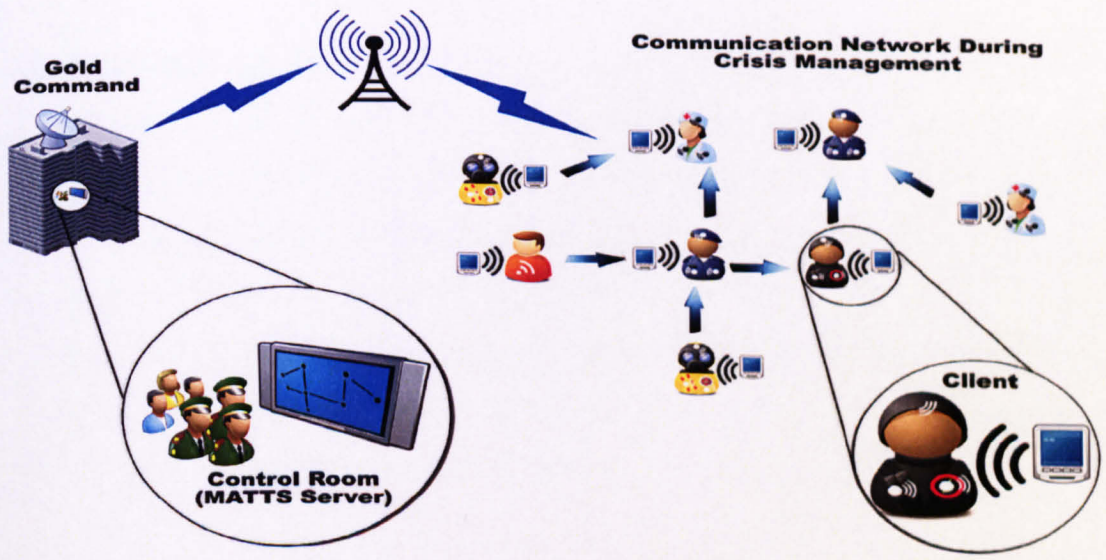


Figure 20 SoS Composition Scenario

The roles assigned to each system or component and their security properties are as shown in Table 3. Each component within the system needs to cooperate with each other so as to be able to resolve the scenario involved in the crisis management situation. Each system might need to share its resources or information that is

deemed relevant and necessary to help resolve and prevent any further disastrous event within the composed system. Specifically, the Sensitivity Level represents the importance of a component and its data in the composed system. Sensitivity levels range between 0-9, where 0 is the highest and 9 is the lowest. The Encryption Strength is represented in an "Algorithm-Key Length" format. A component may support more than one encryption algorithm with various key lengths. The algorithms and key lengths listed in Table 1 are those used by the components to establish external connections. The Encryption Strength is broadly ordered from weak to strong as: *WEP-114*, *RC2-128*, *TDES-168*, *AES-256* [130]. The encryption strengths depicted in this demo can be expanded to include other cryptosystems such as public-key ones. But for the demonstration reason only, we selected some of the most widely used and known cryptosystems. The 'Staff Skills' property considers the technical knowledge and IT skills of each organisation's staff. It might be measured based on training courses that have been attended or certificates obtained. We simply categorise it into levels of *Low*, *Mid*, and *High*. Finally, the use of firewalls and IDS are also considered. Defined properties are saved as an XML file.

To demonstrate the above scenario, we need to employ a simulation tool MATTS standing for the Mobile Agent Topology Test System. This tool is created as part of a project undertaken within the School of Computing & Mathematical Sciences at Liverpool John Moores University. Full details of the project and MATTS can be found in MATTS's manual [131], and a summary of MATTS is provided below. MATTS has been designed principally to allow the testing of secure component composition analysis techniques. Full details and functionality of MATTS relevant to this framework are provided in section 6.2.

The code that is used for sending the property described earlier to MATTS and others nodes is represented in Figure 21, while the actual code that sends the XML file is presented in Figure 22. Before initiating any communication either with MATTS or other mobile clients, the intending devices need to first establish connection. Currently it is done via socket connection with its code represented in

Figure 23. After successful connection, each user within CoI is able to define and communicate their required policies so as to minimise the misuse of identity information and enhance identity management within the composed system.

Code 1:Sending Properties of Node/Device

```
private void SendPropertySet()
    {
      // Send the property set file to the receiver
      FileStream readStream =
new FileStream(Configuration.DEVICE_PROPERTYSET_FILE_LONG, FileMode.Open, FileAccess.Read);
      SendTxt("SEND_PROPERTYSET " + nodeID + " " + readStream.Length);

      // Now actually send the file
      SendFile sf = new SendFile(readStream, sock, false);
      sf.ShowDialog();
      sf = null;
    }

    /// <summary>
    /// Send the node's properties to the server
    /// </summary>
    private void SendProperties()
    {
      // Send the property file to the receiver
      FileStream readStream = new FileStream(Configuration.DEVICE_PROFILE_FILE_LONG, FileMode.Open, FileAccess.Read);
      SendTxt("SEND_PROPERTIES " + nodeID + " " + readStream.Length);

      // Now actually send the file
      SendFile sf = new SendFile(readStream, sock, false);
      sf.ShowDialog();
      sf = null;
    }
```

Figure 21 Sending Properties of Node/Device

The code presented in Figure 21 contains a section of the code that locates the property files of the node that is trying to connect to another node or MATTS. It

TABLE 3. ROLES INVOLVED IN THE DEMONSTRATION AND THEIR SECURITY PROPERTIES.

| *Role* | *Sensitivity Level* | *Encryption Strength* | *Staff Skills* | *Firewall* | *IDS* |
|---|---|---|---|---|---|
| Police | 0 | AES-256 | High | Yes | Yes |
| Ambulance Service | 3 | RC2-128 | Mid | Yes | No |
| Mobile Network Operator | 3 | TDES-168 | Mid | Yes | Yes |
| Department of Transport | 3 | TDES-168 | Low | Yes | No |
| Banks | 0 | AES-256 | High | Yes | Yes |
| Hospitals | 4 | AES-256 | Mid | Yes | No |
| Fire and Rescue Service | 5 | WEP-114 | High | No | No |
| Event Organiser | 5 | RC2-128 | Low | No | No |

takes the file and sends it to the MATTS service. The MATTS server then interprets the file after receiving it, allows the node to join the network, populates the node with its property, and reconfigure the whole network based on the property of the new node. Here, it is worth pointing out that in the UCIM design

and implementation, we have not addressed the issue of key management that is beyond the scope of this project.

Let us assume an attacker is trying to break into a Police computer network and steal valuable data such as information concerning criminal proceedings, or the user identity information so as to impersonate them. However, the attempt fails because the defence mechanism adopted by the Police is too strong for the attacker to penetrate. Therefore, one of his alternatives is to attack via other systems that are connected to the Police system, for example, the Event Organiser's system. Knowing this, the attacker shifts his attention from the Police to the Event Organiser's workstations. Suppose that due to lower security requirements, the Event Organiser is running a server suffering from a flaw that can be exploited by the attacker. The attacker can use this flaw to gain access to the Event Organiser's system, and it is then straightforward for him to gain access to the Police data through the connection between the Event Organiser's system and the Police system. However, in normal circumstances the Event Organiser is not allowed to access the Police data, but if the attacker is able to explore some weakness within the Event Organiser's system, there is a possibility of gaining access to some valuable and/or sensitive information that comes from the police systems.

If the user of the Event Organiser's system has specified a stronger policy in such a way that his/her information cannot be passed from other third parties, and if the policy specified is rightly enforced, then the attacker will not be able to obtain such information. Hence, the attack will not be successful. On the other hand, if the specified policy has been inappropriately enforced, the attacker will be able to gain access to such information. At the same time, if a boundary check to be introduced in the next sub-section was performed when external devices or organisations attempted to join the composed system, such weaknesses could be exposed, and appropriate measures taken to address them. We will demonstrate how this could be done for the systems described in the scenario.

The code presented in Figure 22 performs the function of sending any XML file between the nodes or the nodes and MATTS server. It first locates a file, and reads the content of the file into a stream reader. It then sends the stream across the network connection. It also creates a new file on the other end and copies the stream into the new file. The new file will be available for the intended recipient to make use of. Any exceptions/errors that occur within this process are thrown as required. The code in Figure 23 is used to create a new socket that allows the nodes/devices to be connected to the service. It first creates a new socket, checks its IP address and port, and calls the connect () function within the code. When connection is successful, it disables and enables relevant user interface functions on the device.

Code 2:Sending XML Files

```
//send the xml file to Matts
    private void SendXMLToMatts()
    {
      data = "Initialization;";//beginning of the data to send, indicating the
                    //purpose is to send the device profile/initialise
      try
      {
        //fileStrem  read content of a file for sedning
        FileStream readstream = new FileStream(Configuration.DEVICE_PROFILE_FILE_LONG, FileMode.Open, FileAc-
cess.Read);

        XMLFileData = new byte[readstream.Length];
        readstream.Read(XMLFileData, 0, XMLFileData.Length);
        //attach the name of the profile
        data += Configuration.DEVICE_PROFILE_FILE_LONG + ";";

        sw.Write(data);//sending "Initialization;Profile name;"
        sw.Write(XMLFileData.Length);//send the length of the profile
        sw.Write(XMLFileData);//sending the actual content of the profile file


      }
      catch (Exception ex)
      {
        statusBar.Invoke(new SetStatusBarTextDelegate(this.SetStatusBarText), ex.Message.ToString());
      }

    }
```

Figure 22 Sending XML Files

Code 3:Establish Connection

```
public void ConnectToServer() //change private to public
    {
      //initialisation of communications
      statusBar.Text = "";
      ipep = Configuration.IP_ADDRESS;

      server = new Socket(AddressFamily.InterNetwork,
                SocketType.Stream, ProtocolType.Tcp);

      //start connecting. if success, enable/disable buttons
      try
      {
        server.Connect(ipep);
        Send.Enabled = true;
        btnBrowse.Enabled = true;
        //Getip.Enabled = false;
      }
      catch (SocketException)
      {
        statusBar.Text = "Unable to connect to server";
        Send.Enabled = false;
        btnBrowse.Enabled = false;
        return;
      }


      ns = new NetworkStream(server);
      sr = new BinaryReader(ns);
      sw = new BinaryWriter(ns);
      try
      {
        //Send the "Hello" Message to indicate presence of device

      }
      catch (Exception)
     {}
    }
```

Figure 23 Establish Connection

### 6.1.1A The Boundary Checking Problem

A boundary checking scenario is used to see how a user will be able to protect its identity and other information when other users/nodes are involved outside the boundary of the initial system configuration. As we have also integrated our work into the domain of secure system components composition, the boundary checking is perhaps the most effective and sensible procedure in secure components composition as well as user identity and information protection within a system. In general, by exploiting a weak external interface provided by one component, an attacker can get access to critical information originally well protected by other components. To prevent it, a boundary check should cover many security-related aspects along the communication channels between a composed system and the external world. As an example, the security properties that need to be checked might include the server version and its last update date as well as an automated scan of server-side vulnerabilities. It may then be possible to make a sound judgment about the system based on the information checked [132].

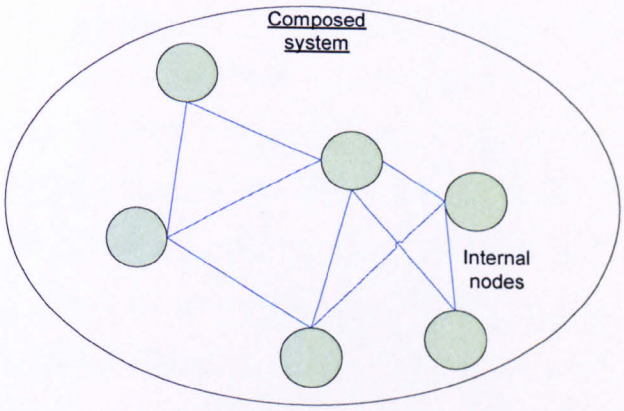


Figure 24 A composed system of trusted internal nodes, with no external connectivity.

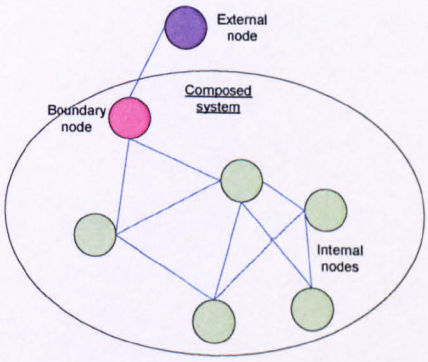


Figure 25 A composed systems with a connection to the outside.

To aid with the explanations in the scenario, two examples are presented in Figure 24 and Figure 25. In Figure 24, a composed system is shown consisting of internal nodes only. As all the nodes are trusted by each other because they form a CoI among themselves, and there is no connectivity to the outside, the security properties of the individual nodes are assumed to be acceptable to all of the organisations involved. In Figure 25, a connection is desired between one of the internal nodes and an un-trusted external node whose security properties are unknown. The internal node is now known as a boundary node and its security capabilities are important for protecting the composed system from the external node. Therefore, the boundary node security properties must be at a level that complies with the prescribed security policy for the composed system in order for the connection to be allowed.

This topology is also represented in the MATTS modelling environment as shown in Figure 27. In this figure, the modelling environment shows the various nodes of the composed system (shown as labelled circles - *P* is the Police, *E* is the Event organiser, *M* is a Mobile network operator, *B* is a Bank, *T* is the Department of Transport, *R* is the Rescue team and fire service, *A* is the Ambulance service and + is a Hospital). The circle in the top right corner of the display is a visual indicator of the results of boundary check system analysis. The system analysis indicator turns green to indicate an acceptable network configuration, and turns red if a boundary node / nodes do not meet the specified security policies. In such a situation, communication between the boundary nodes and the external nodes would be denied and the location of the offending boundary node / nodes causing the security problem would be highlighted.

Having established details of the new device, MATTS adds the new node with appropriate connections to the network model and uses this to perform a re-evaluation of the network properties. If the new network model is considered secure, then the new device is allowed to communicate with the boundary node and the result is used to assign appropriate rights to the connected device. Otherwise, the connection is not permitted. A re-evaluation is triggered each time the connection structure changes, *i.e.*, a device wishes to connect to a boundary node or withdraw from the network.

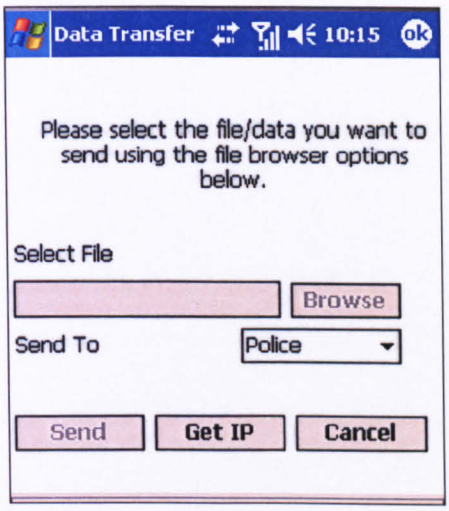


Figure 26 Sending Files between Organisations once connected to the Network using a PDA.

If such a connection is allowed, the user is then given the opportunity to interact with other organisations' devices (either real or modelled) that are present on the network, e.g. by sending a file to another organisation using a PDA as shown in Figure 26.

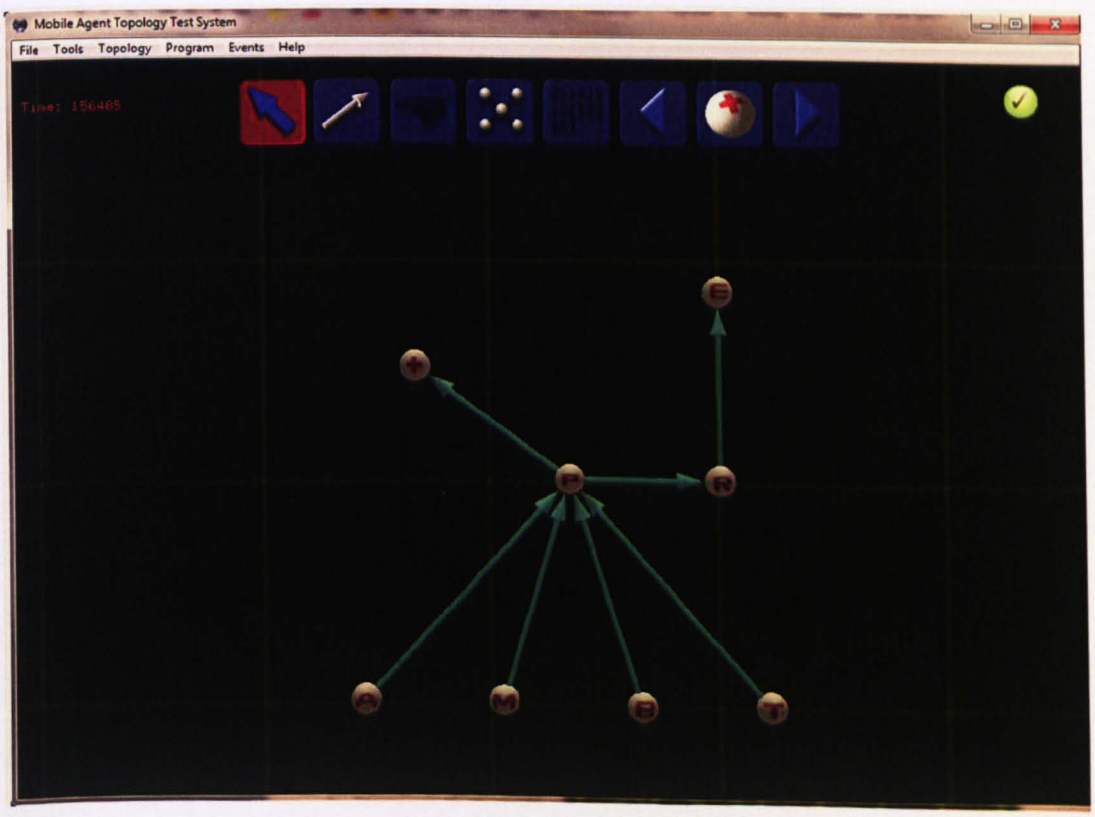


Figure 27 Initial Network

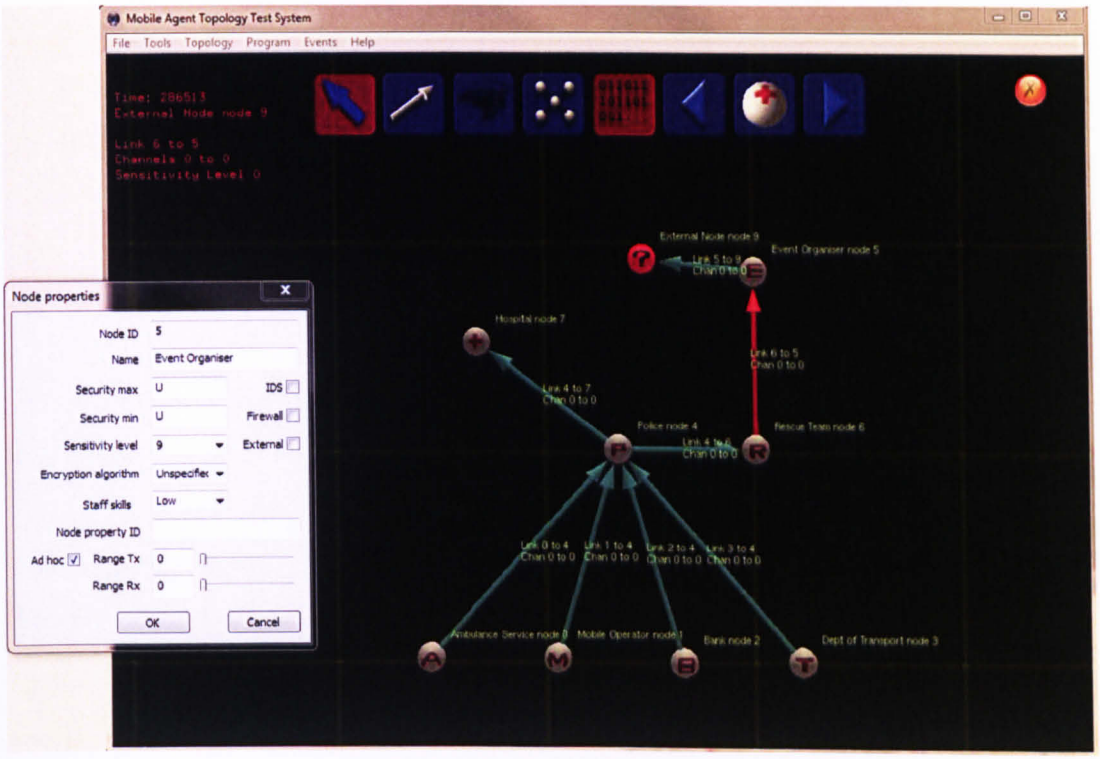


Figure 28 Analysed Network

Returning to the scenario and the application of this policy, if a PDA were to connect to the Police, Mobile Network Operator, Bank or Hospital, the system analysis indicator highlighted on the network topology screen would turn green to indicate an acceptable configuration, assuming that the security properties of these organisations' systems and PDA meet the given security policy. If the PDA connects to any other organisation that does not meet the specified security policies, *e.g.* the Event Organiser, the indicator would turn red and communication between the nodes would be denied as shown in Figure 28. In addition, a visual indicator is presented in order to highlight the location of the offending boundary node or nodes causing the security problem.

Tests of the system using this configuration show that it is able to effectively identify potentially unsafe boundary connections based on the policy, and that it is able to achieve run-time re-evaluation as the network topology changes. Through the identification of unsafe boundary connections, it is easy to establish which nodes in the network constitute an increased risk due to their external network interfaces. Greater security can then be applied to these nodes as required. By being able to properly detect issues resulting from boundary check related

problems, we are able to detect attackers both within and outside the composed system from access to user identity information as well as other information. At the same time, we are able to give the users full control of their personal information by allowing them to define the relevant required policies. This process provides an efficient user centred identity management platform within CoI.

Proper detection and protection of user identity information within the boundary of CoI in SoS is one element that addresses the problem of identity theft and other issues. However, there should be a time when users within CoI need to use their own devices. Hence, proper identity provision & preservation and authorisation & authentication are also of crucial importance. These form the focuses of the next section, where we will extend the above scenario to make use of a test bed for a verification of our work.

### 6.1.2 IDENTITY MANAGEMENT IN SOS

Establishing an IM policy within a single entity is undemanding compared to situations of CoI. Therefore, establishing an IM strategy can be a challenging task for CoI because of its heterogeneity in technology, standards, and identity management implementations. This takes a new dimension when we are dealing with SoS because of the need to have interoperable IM across multiple systems. However, implementing it in a SoS coalition's crisis management environment presents inimitable challenges. Coalition activities or events in regular day-to-day work and involvement always prove to be difficult when it comes to agreeing on how to use personal identity information.

This is more so when at the same time trying to allow users from different organisations to make use of their own devices. Consequently, it is more problematic when it comes to natural or man-made crises that can result in widespread damage to the information infrastructure with large-scale financial, environmental and/or human losses. This can affect the emergency services' (police, fire and medical) ability to communicate with restrictions on the sharing of

information, making it difficult for them to provide emergency services to the population. The availability of important information can play a vital role in humanitarian assistance. Emergency ad-hoc communication networks could be established in order to operate and provide emergency services.

Identity management solutions typically consist of various functionalities. The details of such functionalities are summarised in other publications, e.g. [133]. In this section, we are mainly concerned with two main functionalities: identity provision & preservation and authorisation & authentication. So these attributes are of crucial importance in SoS for managing services such as access control of resources based on specified policies and criteria such as roles, device types, and sensitivity of information within SoS. Life-cycle management of user profiles and other information within CoI in SoS is also essential in terms of who should access such information and share it with whom, what level of details is required, etc.

However, as can be seen below, a successful implementation of identity management within a SoS Crisis Management situation or SoS environment provides a number of benefits. Some of such benefits include: quicker development of SoS, minimisation of identity duplicates, diminishing of privacy and security violations, reducing data and profile misuse, the ability to make use of partial identities, etc. However, to be able to succeed in doing so, we need to make the systems more user-centric in nature.

Building from the scenario in the section 6.1.1 above and Figure 20, we assume four individuals from various agencies and command structures are involved to help resolve the situation within the incident. In the case of this scenario we have a Gold command officer from the Police, a Silver command member from the ambulance service, a Bronze command member from a location where the incident has happened, and a volunteer who is there to help.

The Gold commanders or officers are in overall control of their organisation resource at the incident. They will not be on site, but will be based in a control room. These consist of officers from various agencies involved in the scenario. They will be responsible for formulating the strategy for dealing with the incident. The Silver commanders or officers on the other hand are those that are on ground within the surrounding of the disaster/scene but are not directly involved and are in charge of all their resources. They report back to the Gold commanders. They decide how to best utilise their resources so as to achieve the strategic aims of the Gold Commanders as well as determining the tactics to be used. At the scene of the event, each Silver commander works in proximity and harmony with other Silver commanders from other agencies. They will not be directly involved in dealing with the incident. The Bronze commanders from each agency involved in the incident are those officers that are directly involved in dealing with the incident. They directly control the organisations/agency resources at the incident and will be working with their staff to deal with the incident.

We assume that each of the individuals involved is equipped with their own computing devices (e.g. laptops or smart mobile phones) containing valuable information that is of crucial importance in helping with the sequence of events within the incident. Hence, it is of paramount importance that each user is able to make use of their own devices; their information should be kept as secure as possible; other members within CoI should be able to request relevant information from other members; and individual identities within CoI need to be verified and authenticated. The fifth node within this scenario is the MATTS server that is used as a central focus for authenticating data and forwarding information as required. Details of the scenario and its implementation results based on a test bed are presented in the next sub-section.

#### 6.1.3A OUTDOOR EXPERIMENT USING MESH NETWORK TESTBED

In this experiment the MATTS application has been improved to allow testing of UCIM, by adding modules and applications together with other functionalities. Specifically the extension includes an integration of several tools:

- A secure sandboxed virtual machine, helping in keeping different processes and applications in separate partitions, so as to isolate them from the system and enhance security of the system.
- A mobile code framework, allowing communication between agents running on different network nodes.
- A composition analysis tool, and a formal analysis tool, for analysing the systems for any security breaches or access requests. The tools perform the main analysis job regarding any interaction between the nodes within the system.

Using these tools, a user can model pre-crisis scenarios and a crisis management response emergency network using different nodes, assign security and non-security properties to these nodes, and perform a vulnerability analysis of the modelled network.

To summarise the scenario described so far, we have the following:

- The partners have their own devices, e.g. laptops/netbooks, PDAs, or smart mobile phones.
- The devices have policies and property files as shown in (Figure 29) pre-loaded. The policy file illustrated in Figure 29 demonstrates that each policy contains a device name, a device type, and the organisation of the device owner/user, the role of the user, the sensitivity level of the information to be shared, and the user's authorization level.
- As they try to join the network using the AWDS protocol to be discussed further later, their policy and property files will be sent to the network to define which of the other partners will be able to view their presence, share information with them, etc.

The profile or information from each CoI member is stored in a XML file created using the property interface application, which includes:

- Device type (e.g. laptop, netbook, PDA, or mobile phone)
- Organisation (e.g. police, military, hospital, emergency, or volunteer)
- Role (e.g. gold, silver, or bronze)
- Sensitivity level (0-9)
- Authorisation level (0-9)

```
<?xml version="1.0"?>
<PropertySet>
  <Property id="Name" type="string" />
  <Property id="Type" type="option" />
  <Property id="Id" type="int" rangeMin="0" />
  <Property id="Access Control" type="bool" />
  <Property id="Staff Skills" type="option">
    <Option enum="1">Low</Option>
    <Option enum="2">Medium</Option>
    <Option enum="3">High</Option>
  </Property>
  <Property id="Physical Security" type="int" />
  <Property id="Period of Inactivity" type="float" />
</PropertySet>
```

Figure 29 XML Property File

Each user within CoI will only be allowed to join the network if its policy is in compliance with the network system's policy. Each request of identity information will also need to be made after the initial authentication. The owner of the information has full control on the policy regarding how, whom and when such information should be used or forwarded to. To provide a proof of concept for this scenario, we have conducted an outdoor experiment using a test bed of 5 laptops/netbooks running the Ubuntu Operating System [134] with the capability of creating a mobile ad-hoc network. This is achieved through the use of the Ad-hoc Wireless Distribution Service (AWDS) [135] installed on the netbooks. AWDS is used due to the benefits it offers over other available mesh networking protocols, such as being open source software, running in user space, and more importantly operating in the layer 2 of the network. This results in an Ethernet-based LAN like experience for connectivity requiring no IP addresses, meaning that Layer 3 is left free to deploy any required protocols.

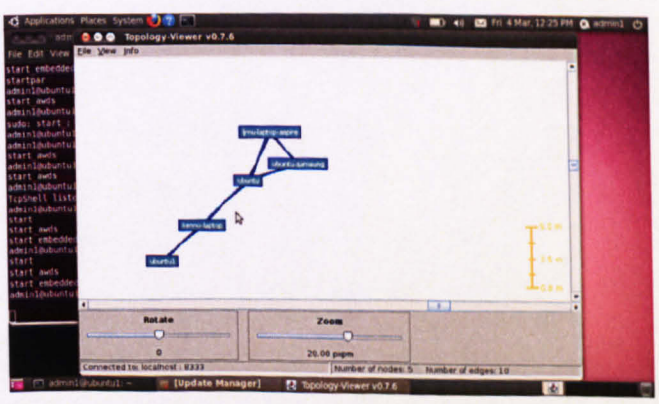


Figure 30 AWDS Topology Viewer screen shot showing node connectivity

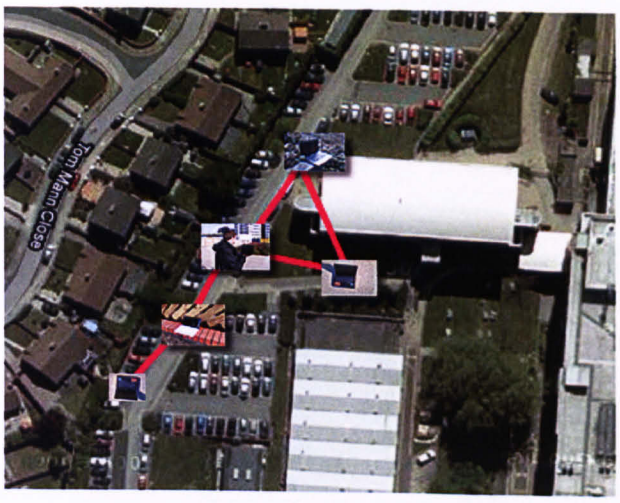


Figure 31 Mobile ad-hoc network in a 100 square metre area using five netbooks

These features allow us to control the connectivity and communication rules within our scenario based on the individual devices' XML property files. AWDS also provides a topology viewer (Figure 30) that allows the user to access a visualisation of the nodes in the network in real time.

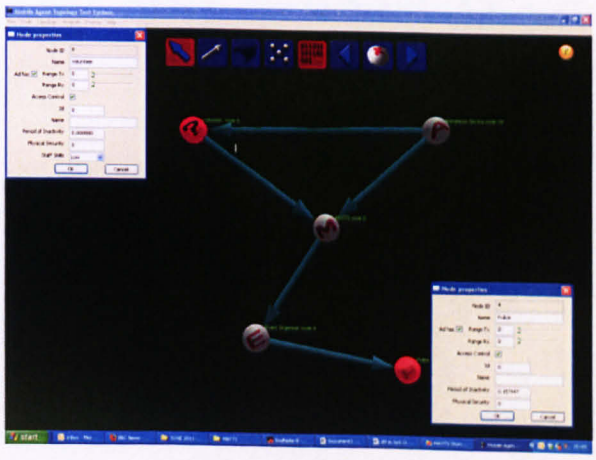

Figure 32 MATTS Interface for IM in SoS

Figure 31 provides a 100 square metre representation of our outdoor experiment using a mobile ad-hoc network on five netbooks representing individuals from the CoI as discussed earlier. In this experiment four of the netbooks represent each of the four individual users within the CoI and the fifth netbook acts as the MATTS server. Each of these netbooks has defined their own policies using the policy interface shown in Figure 45 and Figure 46, and has other identity information stored as XML files. This experiment has demonstrated a case where the volunteer node could access the information of the police node, which would violate the

system security policy. This is because the volunteer node was not secure enough to have access to confidential information such as those stored in the police node. This situation was identified by the MATTS node, so any request from the volunteer node for access to the police node's information was rejected. This is shown in Figure 32 with the affected nodes highlighted in red, with their properties displayed in the box next to the nodes.

The mobile ad-hoc network test bed described above allows the simulation of the scenario with the benefit of executing in a real world. It provides the means of connecting multiple wireless nodes, avoids the need to have a consistent managed infrastructure, or can be used in areas where the infrastructure may be unavailable.

The mobile ad-hoc network capabilities make it an attractive choice for mobile devices because of the benefits it offers, which are self-healing, self-managing, and providing a quite reliable network because every node serves as a relay for the other nodes in the network, if managed properly.

These benefits will be of crucial importance in a crisis situation where the attendees are likely to have devices that offer network connectivity through wireless means such as PDAs, laptops and mobile phones, and the traditional static infrastructure based communications are damaged.

The network connection mentioned previously will allow the sharing and gathering of information and data about the situation that can be used by the command and control structures to inform the decisions made [136].

The interface for this server-side software using IM in SoS is shown in Figure 32. The figure shows nodes from various interconnected organisations. In the upper right corner of the interface the 'X' indicates a security problem has been identified in the network that involves multiple nodes simultaneously. This provides a notification. The dialogue on the left in Figure 32 shows how the user

can modify the properties of a node in order to manage the model. This also provides a means of resolving security issues by making suitable changes to node properties (e.g., by increasing staff skills assigned to the node).

After establishing the required authentication, another issue that is of paramount importance with CoI in SoS is how the information transferred from one member in CoI to other is used. The information could be mishandled and possibly used for profile building by attackers of CoI. These are the focuses of the next sub-section.

### 6.1.3 Data Mishandling and Profile Building

Data mishandling/forwarding and gradual profile building, possibly for some malicious intent, is one of the serious concerns in ubiquitous computing as a result of the growing threat in identity theft. In general, being able to forward certain confidential information from the original intended recipient of the data based on its fulfilment of the policy required or trust of the data owner without the knowledge or approval of the owner might result in serious consequences. Also, gradually storing partial identity information of a user from one social event or interaction to another might result in an array of partial profiles that when put together might reveal more than what the users might want to reveal to anyone. Hence, a process of being able to control the usage of information after its initial exchange is of curial importance. Also by exploiting a weak external interface provided by one user, an attacker can get access to critical information originally well protected by other users, if these information is not fully protected from being forwarded or is allowed to stay with the other user after the expiry of its usage. To prevent it, the owner of the information should be able to define policies to cover many security-related aspects along the communication channels between a networking system and the external world. As an example, the security properties that need to be checked might include that if the distance between the users is violated, the information should be automatically deleted; and the information should only be available for a given duration; and there should be the ability to automatically check against such rules. It may then be possible to make a sound judgment based on this. The pre-agreement security policy within the individuals

in Fig 13 is to share any required information between the parties of CoI if they are within the boundary of the system and scenario environment. Any member of CoI, which violates this agreement by moving out of the system boundary, is to relinquish the right of using the already shared information and not to forward it to any other parties outside the system boundary.

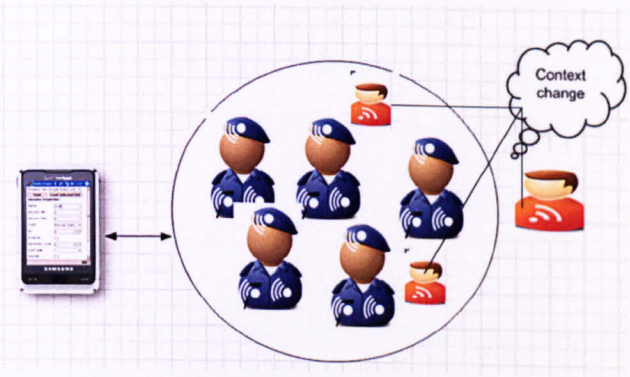


Figure 33 Profile Building

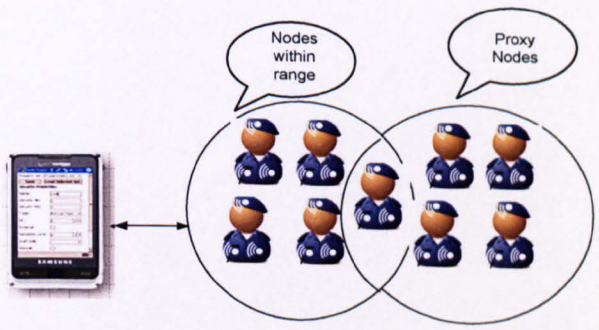


Figure 34 Data Mishandling

We utilize two examples depicted in Figure 33 and Figure 34 to aid with the explanations contained in this section. In Figure 33, a possible example of profile building after a change of user contextual information is presented. It shows a set of users that meet the policy of the devices connected as well as a situation of 3 users who have changed their profile settings or violated some of the device policies. If the partial profile of a user is not automatically removed from the devices of such users after few exchanges of different profiles and partial identities, it is possible to build a whole picture of the other users from those obtained partial identities.

In Figure 34, a connection is established between one of the nodes, which shares connection with another device or set of users, and those users referred to as proxy nodes whose security properties are unknown. The node that intersects the two sets of nodes can potentially forward data from the nodes within one set to other nodes in the other set, if relevant security policies have not been specified for such data. A classic issue that we are trying to address is how an operating platform like Android [48] or Apple [49] OS allows installed third party applications to forward users' private information, device identities etc. to third parties without informing the users.

Continuing from the scenario in the previous section, let us assume that one of the ad-hoc connected trusted users is somehow forwarding some important data to another user outside the network due to lower security requirements or measures taken to disallow any form of data transfer without the authority of the original owner of the data. This can also be a case whereby a trusted user has legally obtained the information before leaving the connected environment, and there is no mechanism to examine when any of the users has violated any of the security requirements, i.e. leaving the network with the sensitive information. All sensitive data and user partial profiles should be revoked from the user as soon as possible. If there has been no pre-defined or dynamic way of defining or enforcing such policies, the weaknesses could be exposed. However, by careful policy definition, appropriate measures can be taken to address them. We will demonstrate how this could be done for the system described in the scenario.

In our demonstration, we use the icons shown in Figure 27 to represent various organisations/users within the scenario. To establish the data and partial identity properties of the individual users, each organisation represented by an icon checks its own security-related properties (this information may be pre-established) and the user is allowed to define new policies as required. Hence, we use the windows mobile HTC shown in Figure 35, the PDA in Figure 36 and Figure 37 for the deployment of our implementation in the Microsoft Visual Studio 2005

environment. Figure 35 shows a state whereby one of the nodes has violated the set policy and a message is shown informing the user that this has been violated and the shared file/information has been deleted from the user's device. Figure 36 on the other hand shows a similar situation, but now the user concerned has fulfilled the required policy and the required file is transferred to the node. Figure 37 is similar to Figure 35, but on a simulating device instead of a real device. Communication between the devices is done via wireless communication. We make use of the x and y co-ordinates of individual users within the devices to determine their locations in relation to other users within the system – for simulation purposes. This can be replaced by actual coordinates of devices in real life situations.

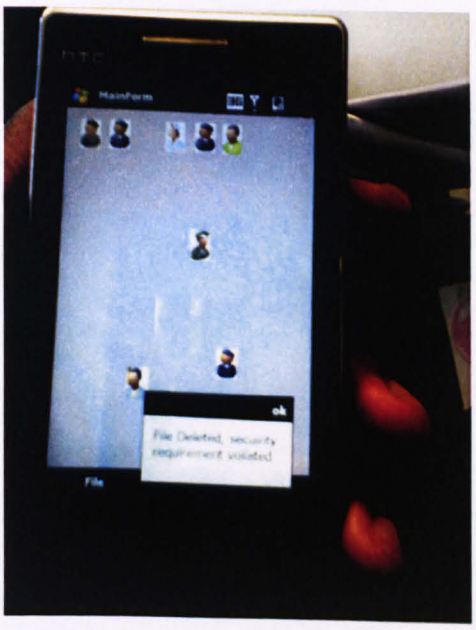


Figure 35 Dynamic Policy Demo Tool - with the file deleted due to a policy violation

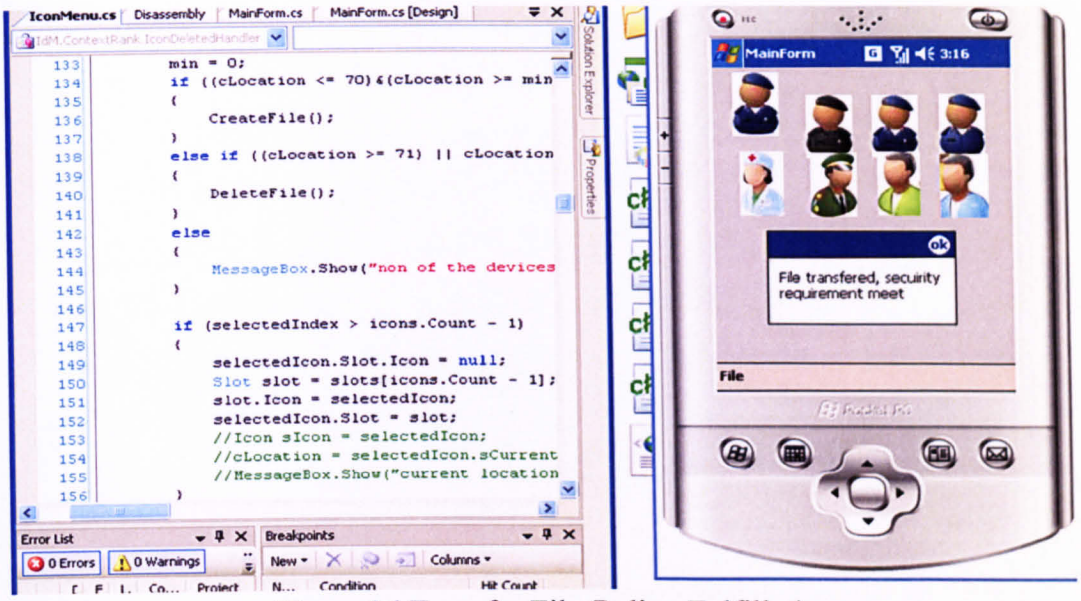

Figure 36 Transfer File-Policy Fulfilled

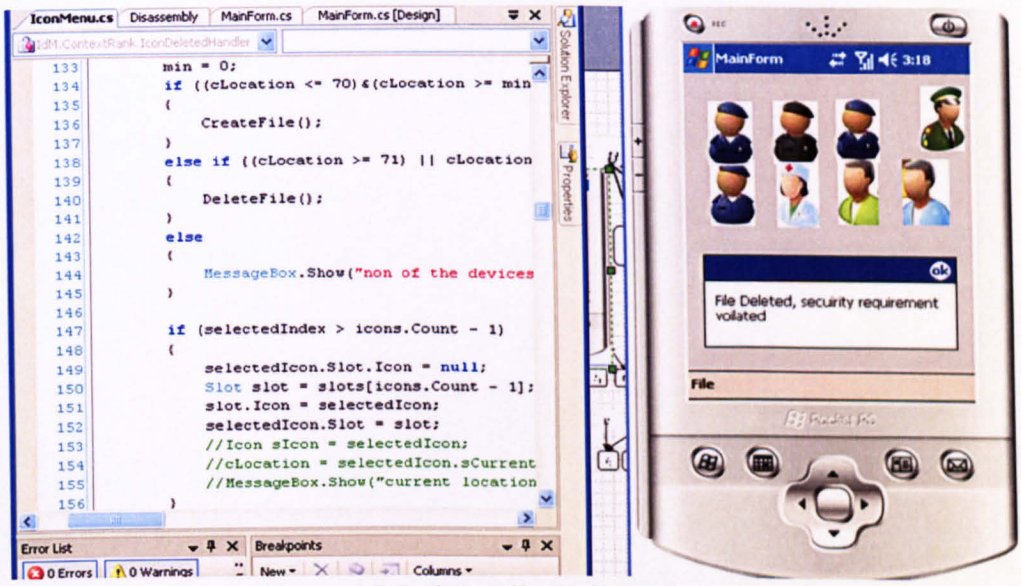

Figure 37 Delete File-Policy Violated

The user of the HTC and PDA is able to specify to which organisations' systems he wishes to send its information, either partial profile or data, after checking the compatibility of its policies.

The method presented in this thesis provides a number of advantages against existing traditional methods. It allows devices to be configured during runtime with little or no interruption to service provision. Users are able to make some

changes to the settings, which will instantly take effect dynamically. It makes the configuration process dynamic, increases the range and security control of the application, and eliminates the needs of an expert to change policies and other settings by allowing users to define such policies via a simple user interface, which will in turn be translated into policies and the device get configured dynamically. As it currently stands, we only anticipate problems when it comes to dealing with more complex policy negotiations, possibly requiring a more in-depth method and algorithm to handle such situations, which forms part of our future direction of study, as well as enabling multi-hop communication and handling other security issues such as an integration of services.

The implementation of the above example using the policies defined is shown in Figure 36 and Figure 37, where profile files are either transferred or deleted based on the location of the owner of the information and its location in relation to other users, contextual information and policy specification. Here, each device is configured dynamically according to the policy specified, and re-configuration is also done dynamically and automatically after any change of contextual information, policy etc.

### 6.1.4 Device Configuration

Before going into the details of device configuration, let's define some of the terms used within this section and thesis:

- Target or target device: this connotes to the device or resource that other devices/resources will like to communicate with, e.g. by requesting profile information, data etc.
- Subject: this denotes either a device or service that is requesting information from the target device.
- Proxy: this signifies a subject that can go through to request information indirectly from a target that can send relevant information to the subject. For example, *target A* is connected to *subject B*, *B* is connected to *subject*

*C*, but *A* has no direct connection to *C*, so *A* is indirectly connected to *C* via *B* and *B* is a proxy of *C*.

It is also worth pointing out that although we have utilized the components within the proposed methods as either devices or persons (individuals), they can also be equipment's, e.g. medical equipment, fire service equipment's, other equipment's/services that might help users of the devices to fulfil their responsibilities in large dynamic crisis management in ubiquitous environments.

We divide the process of dynamic device configuration in three main stages illustrated in Figure 38: configure the interface based on either a pre-defined or user defined policy, obtain contextual information within the range of the device, and configure it dynamically based on the contextual information gathered and negotiate policies and user requests dynamically. The algorithm depicted in Figure 38 shows the process of configuring the device. It first gets the list of all nodes within the range. Compare the target and subject node policies. If there is a match, it creates a list of nodes, sends the new list to the target node, and configures the device. If there is any change in the contextual information of proxy nodes, this will also notify the relevant nodes and trigger the process of reconfiguration of the nodes.

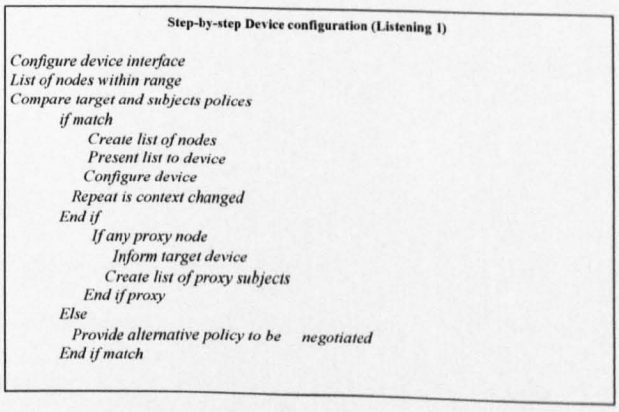


Figure 38 Step-by-step Device Configuration

### 6.1.5 USER INTERFACE (UI) CONFIGURATION

Each device will have some default settings of its device profile, user profile and policy settings. When the device is introduced into the environment, the automatic configuration process is initiated. Figure 39 presents the flowchart of the required process. If the user preferred settings are automatically detected, the user will have the option to either modify such settings or not. This process is not applied if there are no predefined user settings. If the default settings are selected, the information is then presented to the module that automatically and dynamically configures the device.

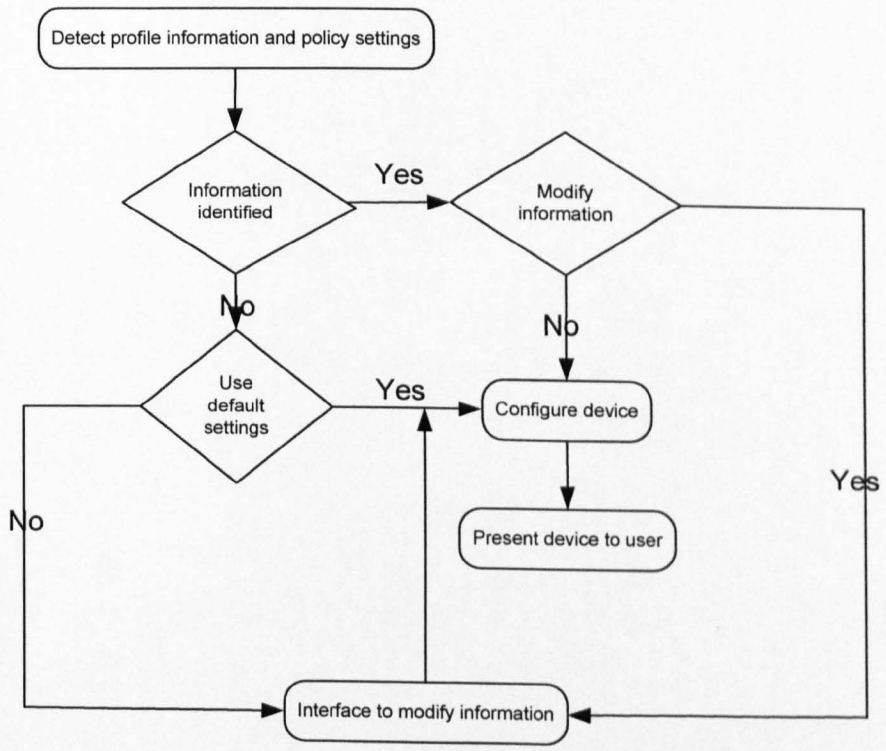


Figure 39 Flow chart for configuration device

Then the configured device should be presented to the user. If the user chooses to modify the default settings or their own specified settings then an interface will be provided for them to modify such information. After such modifications the process of configuration will start.

If the contextual information of the device changes, this information can be changed/updated dynamically with little or no impact on the availability of the service and functionality of the device to the user.

The above can also be implemented in a way that the user is permitted to specify one or more preferences associated with the device configuration settings, including the user's policy, some parts of the user profile, etc. New data fields can also be defined to be used for data entries in terms of new properties, for policy creation or user profile specification. The layout of such an input interface will be based on a Meta data, i.e. using XML files to generate data entry forms dynamically. Some of the computing devices used are designed to interact in a client server setting. Client devices include: PDA's, mobile phones, HTC, palmtop, iPad, etc. The communication between such devices and the server is done wirelessly through communication interfaces, and the communication can be via audio, text, pictures, images and data transfers such as files between the devices or clients and server. Each mode of communication will be automatically detected and determined based on device setups using policies and relevant contextual information provided and the given scenario.

## 6.2 MATTS AND COMPOSITION CLIENT APPLICATION

MATTS has been introduced in section 6.1.1, which was designed principally to allow the testing of secure component composition analysis techniques. In this section we provide further details of MATTS's functionality. We have improved MATTS to allow the testing of UCIM in crisis management scenarios, by adding modules and applications together with other functionalities. The extension integrates together a number of tools as explained in section 6.1.3.

MATTS is implemented as a single native C++ application. Within the single application it is possible to model multiple interacting components by executing a byte-code that can be compiled using a RISC-like Assembly language.

The compiler wraps up executables into neat bundles that include a code section and a block of workspace memory. The MATTS language is unusual in that it works directly in memory, it does not make use of any registers, so that all states are held entirely in the workspace memory. MATTS is therefore able to implements a simple mobile code framework that allows any executing code to automatically transport itself to another machine running MATTS with one simple instruction.

MATTS is able to run multiple pieces of code simultaneously that are also able to communicate with each other. This allows component topologies to be built up using links between components. Simple 'Compose' scripts – which are in an XML format – can be created that allow component topologies to be tested against security criteria. The language used to create Compose scripts is actually itself Turing Complete, and has been designed to suit the task of analysing topologies. A language is Turing complete if for each function defined within the language that can be calculated with a Turing machine, it can be proved that there is a program in this language that performs the same function. A Turing machine is a very simplistic theoretical computer model with an endless tape of cells, which has a given set of symbols, a pointer to the cells and a finite state control. It is also capable of manipulating those symbols according to a given set of rules.

An important design element of the system is that it provides a graphical overview of all the running agents and their communication links. The user can interact with this in order to control the agents. In addition, a console is provided for each agent to allow direct interaction with the agent. There are two levels of interaction: interaction with the agent and interaction with the system of agents (the composition). There is a slight added complication to the actual MATTS application, in that nodes can also be representative of remote devices (*e.g.* portable devices) or even further instances of the application itself. In this case, the remote devices are 'dummy' nodes, since they're not capable of running arbitrary code and instead only have a subset of functionality plus a local on-device control Graphical User Interface (GUI). We have also implemented a

version of the dummy nodes as the CompositionClient application, run based on mobile devices or PDAs. However, this version of dummy also runs functions that help the intermediate process, but actual analysis of the whole system is done in MATTS.

The CompositionClient application has been primarily designed and implemented for the purpose of setting devices/node properties by the users themselves rather than just hard coding such properties in MATTS. This has been further extended to allow devices to be able to send properties files and other files, create connections with the server, and display a dynamic interface using property sets as well as allowing users to define policies dynamically using the policy interface creation tools. All of these components of the CompositionClient will be discussed in the next few sections as well as the rest of the thesis.

Finally, Compose scripts can invoke a formal analysis engine that is built into MATTS that is able to analyse executables based on pre, post and mid conditions. Although it's a relatively simple analysis tool, it allows some quite detailed information to be gained about specific pieces of code and to be used during component composition analysis. We present the code that initialises device connection (from the MATSS point of view) as well as creates a buffer and an XML object to load the relevant properties of the device, as shown in Figure 40.

**Code: Device Connection**

```
/*
=============================================================================
* Initialise the DeviceConnection object
=============================================================================
*/

DeviceConnection::DeviceConnection (SOCKET nAcceptSock, cServer * psServer, cWindow * psWindow) {
        this->nAcceptSock = nAcceptSock;
        this->psServer = psServer;
        this->psWindow = psWindow;
        eCommand = CONCOM_INVALID;
        eState = CONSTATE_COMMAND;
        nFileRead = 0;
        nFileSize = 0;
        nNode = 0;
        // Create a cSaveXML object to load the node XML properties
        psSaveXML = new cSaveXML (psWindow);
        // Allocate some space to read in the property info
        // TODO: Change the property reading to use an input stream (smaller input buffer)
        pcProperties = (char *)malloc (PROPERTIESXML_BUFFER);
        nPropertiesSize = 0;
        pcPropertySet = (char *)malloc (PROPERTIESXML_BUFFER);
        nPropertySetSize = 0;
}
```

Figure 40 Device Connection

The code presented in Figure 40 is used for connecting a device to the network/ MATTS server in this case. The code first creates a device connection function in MATTS, taking three parameters: socket, server and cWindor. It also creates a few local variables such as nFileRead, nFileSize and nNode. It then creates an XML object to load the node/device XML file property. This is done by allocating some space in the memory that will store this information temporarily before the connection is established; MALLOC is used in this case. It further tries to establish the connection via the created socket to the server. After a successful connection, the stored property details in the memory are used to fill up the required property fields of the device within MATTS's environment and the node/device is now ready to interact with other nodes within MATTS as required. If there is an error in establishing connection, this will be notified to both the server and the user to take an appropriate action.

The actual code from the CompositionClient that is responsible for sending the XML file after a successful connection is shown in Figure 41.

Our work has improved and extended some modules in MATTS, which mainly allow different components/systems to be connected to MATTS either virtually or using real devices like the HTC phones and PDAs that we used in our implementation as well as other nodes like netbooks. We have created the code and relevant user interfaces that handle such processes. We have also created the policy and property tools that allow MATTS and other virtual or real nodes to be able to define their own properties and policies. We have provided means for MATTS to receive, interpret, use and send any XML file within the system as well as receiving any other files required. We have utilised the already existing analysis scripts within MATTS to help the testing of the scenarios used in our implementation. The CompositionClient is another tool that our work has developed, which enables MATTS functionality with real devices.

**Code: Send XML File**

```
//send the xml file to Matts
private void SendXMLToMatts()
{
  data = "Initialization;";//beginning of the data to send, indicating the
                //purpose is to send the device profile/initialise
  try
  {
    //fileStrem  read content of a file for sedning
    FileStream readstream = new FileStream(Configuration.DEVICE_PROFILE_FILE_LONG,
FileMode.Open, FileAccess.Read);

    XMLFileData = new byte[readstream.Length];
    readstream.Read(XMLFileData, 0, XMLFileData.Length);
    //attach the name of the profile
    data += Configuration.DEVICE_PROFILE_FILE_LONG + ";";

    sw.Write(data);//sending "Initialization;Profile name;"
    sw.Write(XMLFileData.Length);//send the length of the profile
    sw.Write(XMLFileData);//sending the actual content of the profile file


  }
  catch (Exception ex)
  {
    statusBar.Invoke(new SetStatusBarTextDelegate(this.SetStatusBarText), ex.Message.ToString());
  }

}
```

Figure 41 Send XML File

## 6.3 CONTEXTRANK APPLICATION

### 6.3.1 SCENARIO

We consider a public event such as a conference as an example. In this scenario we suppose that delegates at the event come from various disciplines in the research community. Each delegate requires a system or a way of being able to access facilities provided by the conference or for their own purposes (*e.g.* a PDA or phone). Some delegates might have more than one area of research interest, or might also have undergone some training or have qualifications outside of their speciality. For example, they may be trained in first aid, have been a professional medical doctor before moving into research, or be a member of an approved security agency. During the event, organizers might want to tailor their programme to the needs of delegates as well as trying to utilise delegates with certain skills for other tracks or other unexpected events during the conference such as accidents, medical emergencies, and so on.

By using the dynamic policy creation tool, users will be able to define their policies and personal profiles dynamically and can set up policies and/or partial identities to release different elements of their profiles automatically depending on the context. This can be achieved by the ContextRank algorithm proposed in section 5.7 to filter relevant information for each user and help the organising committee as well as other delegates to make best use of the available resources and help each other when required. The process involved is depicted in the sequence diagram in Figure 42.

The process in Figure 42 first requires the device to send its property file/details to the policy interface module. This is used in generating the policy of the device with user intervention and input as required, or the default policy file from the device can be used instead. The policy of the device is sent to the contextRank algorithm module. Then a loop is initiated between the context information sever module and the contextRank algorithm module. The loop is for computing the weight of each relevant contextual information item within the context server that meets the policy of the device. The weight of each contextual information item is returned to the algorithm. Another loop is initiated to try and find if there is any similarity between the available contextual information items, and the result is also sent back to the algorithm. The final result of the computation is then returned to the device as a list of $n$ relevant contextual information items that meet the policy of the device and can help the user of the device to fulfil its role within the scenario concerned.

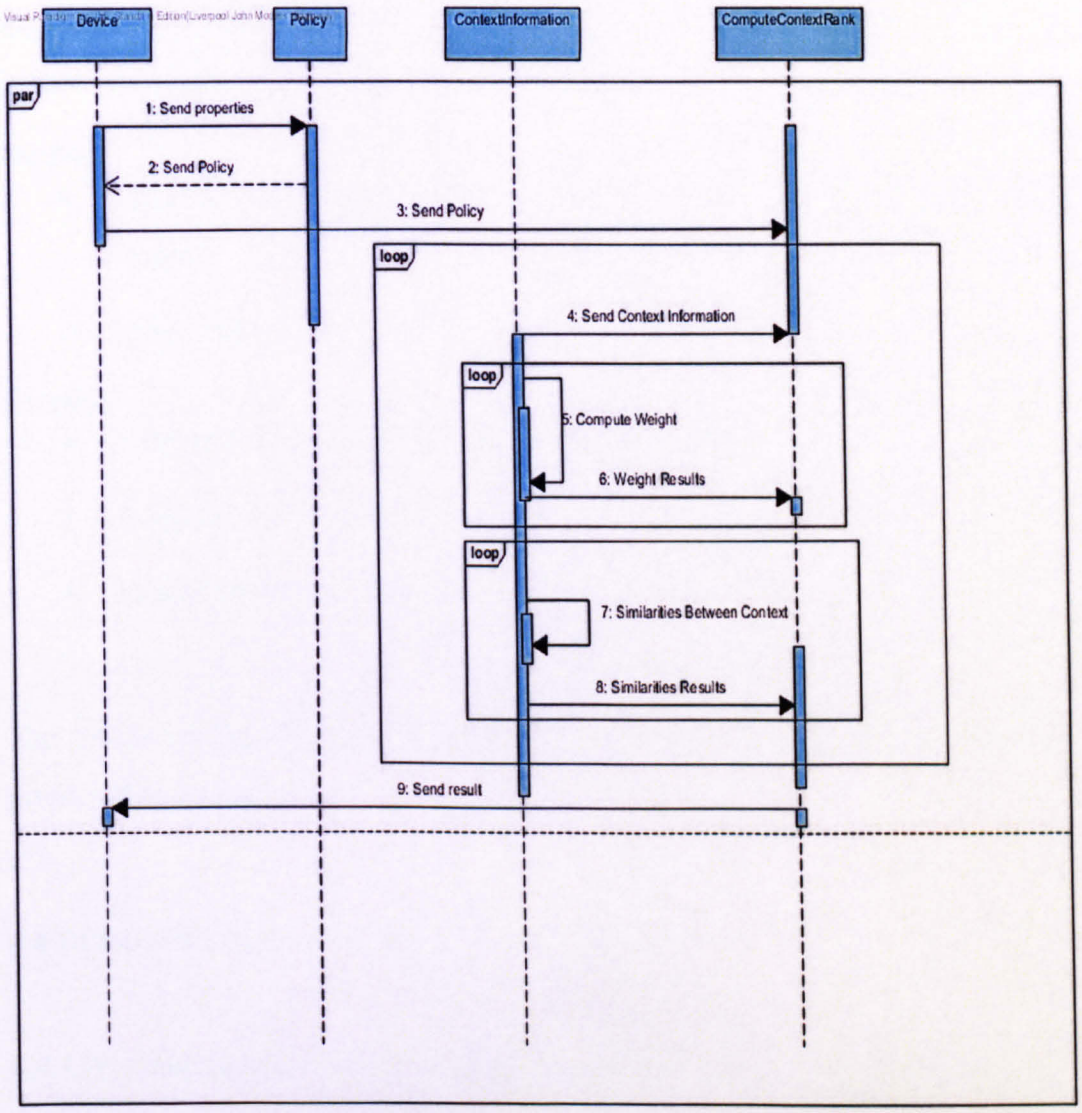


Figure 42 ContextRank Sequence Diagram

### 6.3.2 OVERVIEW

The ContextRank application is designed to function in two main ways: either just as a filtering algorithm considering only the relevant contextual information for the user, or incorporating user defined policies from the Dynamic Policy creation module or any other policy in filtering contextual information. It is designed to take in the criteria under which contextual information can be filtered and produce an output for the users that meets its required criteria.

Hence, the algorithm will be able to filter delegates based on their speciality. For example, after an incident, the organizing committee might want to get

experienced medical doctors or first aiders, this algorithm will filter that information and redirect recourses as needed.

Inputs

- An array or list of all available contextual information within the range of the user.
- One or more policy files.

Outputs

- A list of relevant contextual information for the user.
- Access to users' data or profile info.
- Possibly a request for further information if required.

The details of the process involved in generating the output from the input are presented in section 5.7.

## 6.4 DEMO WITH POLICY INTERFACE APPLICATION

### 6.4.1 DYNAMIC POLICY SPECIFICATION INTERFACE

Defining user policies dynamically is one of the key elements of our proposed UCIM framework. In this section we provide the details on how a user is able to interact with a simplified and implemented user interface to create policies from a given set of property files. The property files define the available properties of individual nodes within the system. Each property will be used as an element within a rule, and combining such rules will create a policy. Other underlying technical functions are hidden away from the user, which have the capability of taking the user's input from the policy interface to create an XML policy file that will be used within the system when needed. Below we provide an overview of the key elements that helps the interface to be user friendly, together with some aspects of its components:

- Users are allowed to dynamically specify policies using predicates and functions.

- Policies are built from predicates (which are just inequalities containing properties and values) and sub-functions, joined together using logical operators.
- The use of sub-functions within the interface eliminates the problem of tackling complex mixtures of conjunctions and disjunctions; however, it is also possible that this will weaken the required flexibility.
- A policy can be easily translated and used in many other applications.
- The methodological design can be implemented for any scenario and is adaptable – it's not designed for a single purpose usage.
- Policies are aimed to be stored in a tree structure in memory or in XML files in a way that is lightweight and portable.
- The design of the system is modular –meaning new modules can be plugged in to improve functionality.
- The methodology could be easily integrated into current existing solutions, *e.g.* Ponder, to act as client applications and pass on created policies to the Ponder engine for processing.
- Policies can also utilize users' partial identities.
- Users can modify policies, e.g. change their areas of speciality or training to help in emergency situations. This will take effect dynamically and with minimal interruption of the system.

Inputs to the interface are defined manually through the interface, or loaded as an XML file defining necessary predicates. These inputs include:

- A list (array) of properties. This is in the form of a PropertySet XML file as output from the PropertySet Tool,
- User defined constraints on elements of the properties and other sub-rules/policies.

The outputs from the interface are currently produced in the form of an XML file containing one or more rules/policies/predicates for which satisfaction can be determined.

### 6.4.2 POLICY XML FILE

Policy XML files specify simple security policies. In this context, a security policy is simply a statement in propositional logic that can be evaluated by substituting the values of properties into free variables. A policy is considered to be satisfied if it evaluates to true, and violated otherwise.

These policies are really intended to be applied to a single device. To tie multiple policies together across multiple devices, Compose XML files should be used.

Although the policies are encoded in XML, the important part is really just the logical statements, and the functions needed for substitution.

Policy XML files are created using the PolicyCreate tool. Policy XML files are used by the ContextRank algorithm defined in section 5.7. They are also referenced by Compose XML files in order to create more complex policies interpreted by MATTS.

```
<Policy name="Risk">
 <Function id="main">
  (LowRiskMedValue AND MedRiskLowValue)
 </Function>
 <Function id="LowRiskMedValue">
  ((Risk~Level &lt; 0.2) AND (Asset~Value &lt;= 2))
 </Function>
 <Function id="MedRiskLowValue">
  ((Risk~Level &lt; 0.5) AND (Asset~Value &lt;= 1))
 </Function>
</Policy>
```

Figure 43 User Policy

As an example, Figure 43 shows a policy requirement defined using some of the predefined properties of the device. These predefined properties serve as the elements to be used in creating a policy. The figure demonstrates only the use of the 'AND' and 'OR' logical operators. A small window in the interface (Figure 46 shows the user the policy defined, while the translated policy is shown as an XML file.

We apply the data abstraction rule to determine the level of details of the delivered data, when a positive access decision is returned from the access control rule.

To access this information at any time from the interface, the following rules will apply:

```
If (profileType == office &(actionType =
        = triggercommunication|actionType =
        = makeAppointment))
{
pID == office
} else
If f(profileType == Home)
{
pID = Home
} else
if(profileType == Social & (actionType =
        = triggercommunication|actionType == bookTable))
{
pID = Social
} else
If (profileType == HealtCare)
{
pID = Healtcare
}
```

The example rule above, checks to see if the profile type is office and the action within the rule is either trigger communication or make appointment. If this case is true, it sets the user pID to office. Otherwise, it checks if the profile type is

home. If the case is true, it sets the pID to home, etc. After setting the pID, other polices relevant to the pID will be enforced automatically, i.e., it allows only users whose pID are the same, e.g. home, to see the user available within the network.

In implementing the policy specification interface, we have implemented two versions. One mainly targets mobile devices and their properties for creating a policy file based on another in-house developed application that allows users to create new properties for a system. The second version is mainly web based but can still be used in mobile devices as it follows our main principles of making the framework lightweight. In doing so, we have divided the implementation into three different modules: the property interface, Policy Logic and Policy Creator. The property interface allows the users to define the data fields/properties that can be used to define the required policies. The policy logic deals with the technical aspects of the policy creator. This contains details on translating the input from the policy creator to create the set of policies for the system. The policy logic hides all the technical details from the user. The user makes use of the policy creator to define policies for its system.

### 6.4.3 THE PROPERTY INTERFACE

The property interface is designed to enable the systems to allow each node's property to be set or determined on an individual basis. Each node can have different property options and therefore capabilities rather than each node having the same set of properties in a static manner. The property interface aims to make the device user interface to be configured dynamically with different property options to meet the challenging needs of ubiquitous environments and the need of individualised policy specification by each node/user based on their profile types. Full details of the property interface can be found in our internal report [137], and a summary of the interface is provided below.

The process for the dynamic creation of a device interface and properties of individual nodes is based on using XML file. This is predefined with basic root elements of node properties, and users are allowed to modify or add new

properties to the files or even create a new set of properties and store them as an XML file. Although it is possible to write the required XML file in a text editor, this is not user friendly and not suitable for non-technical users or in emergency situations. Hence there is a need to develop the interface. The structure of the XML file is presented in Figure 44.

```
<?xml version="1.0" encoding="utf-8" ?>
<PropertySet fileName="\ProgramFiles\PropertySet1.xml">
  <Property id="Name" type="Text"/>

  <Property id="External" type="Binary" />

  <Property id="Sensitivity Level" type="Integer" rangeMin="0" rangeMax="9" />

  <Property id="IDS" type="Floating Point" />

  <Property id="Encryption" type="Option">
    <Option enum="1">WEP-40</Option>
    <Option enum="2">RC2-40</Option>
    <Option enum="3">RC4-40</Option>
    <Option enum="4">WEP-114</Option>
    <Option enum="5">DES-56</Option>
    <Option enum="6">RC5-56</Option>
    <Option enum="7">RC2-128</Option>
    <Option enum="8">RC4-128</Option>
    <Option enum="9">RC5-128</Option>
    <Option enum="10">TDES-168</Option>
    <Option enum="11">IDEA-128</Option>
    <Option enum="12">Skipjack-80</Option>
  </Property>
</PropertySet>
```

Figure 44 Example of different property types

The properties have two main attributes: the *id* attribute gives a property a name, such as "Name" or "Encryption". The *type* property defines the type of the property, which effectively decides how it is represented on a User Interface. There are five different types currently:

- **Text** – the property is represented through a text value, for example, the name might be "Police".
- **Binary** – the property is represented by means of a true or false value (a checkbox on a user interface). For example, "Is the node external? Yes/No".
- **Integer** – the property is represented by an integer. This can be un-ranged without any extra attributes, or includes a minimum value, a maximum value or both.

- **Floating Point** – the property is represented by a floating-point number. In the CompositionClient this is for a specially created "FloatingPointTextBox" control.
- **Option** – this property is represented by a set of options. These options are represented by child nodes in the form shown in Figure 45. The value of an option is represented by the text value of the node.

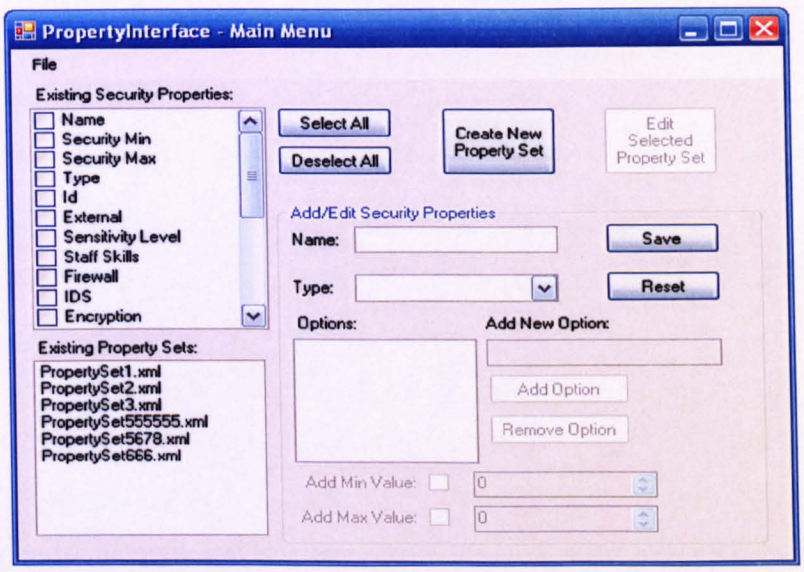


Figure 45 Property Interface Main Menu

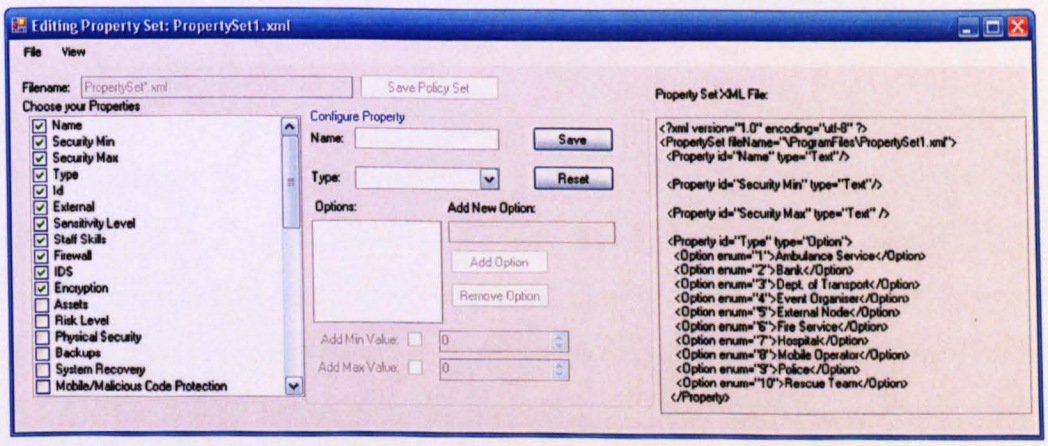


Figure 46 A Fully Populated Property Set

A user can create a variety of XML files similar to the one in Figure 44 using the interface presented in Figure 45 and its sub-interface. The interface in Figure 45 is used by users to create new property sets, add or edit security properties. It also displays the list of available property files so that the user can select the required one, modify or add new properties and save the file. An example of a fully created

property set is shown in Figure 46. The interface in Figure 46 consists of three sections. The left part displays available properties from the loaded property file, and ticking the check box next to a property name means adding this property to the new file. The middle part allows the user to configure new property elements, and this in turn will be added in the left part of the interface. Finally the right part of the interface displays the XML file of the new property file being created.

### 6.4.4 THE POLICY LOGIC

When implementing the policy interface, we have adapted two versions, one for PDA/mobile devices (HTC phones and other phones emulators) and the other for web services. However, both of these two implementations are compatible to PDA's, HTC phones and desktop PCs. The main difference between the two is that the PDA version has both the policy logic and policy interface in one implementation, whereas the web service based implementation provides a separation between the policy logic and policy creator interface. This is needed so as to separate the technical aspect of the policy creation from the user and also to make the interface more lightweight in terms of trying to deploy it using devices with fewer resources. It also helps to make things simpler for the end user of the system.

The main role of the policy logic is to implement the required parser for the web service based policy creation tool. There is no user interface attached to this module, but it has to be running before the user can run the actual policy creator interface. This module consists of two main part, a machine generated parser by peg-sharp 0.3.427.0 from Parser.ged, which is a C# packrat parser generator. Parsing Expression Grammar (PEG) is a type of analytic formal grammar. It is used to describe a formal language in terms of a set of rules for recognising strings in language. Hence, we make use of this parser to transform our grammar in the design section into codes and rules for user defined policies. The second part of the module consists of hand written code, developed to be used by the semantic actions for the machine generated parser. An extraction of the machine generated code is shown in Figure 47, while an extraction of the hand written code is shown in Figure 48.

**Code: Machine Generated Parser Code**

```
// Machine generated by peg-sharp 0.3.427.0 from Parser.peg.
using System;
using System.Collections.Generic;
using System.Globalization;
using System.Runtime.Serialization;
using System.Security.Permissions;

namespace PolicyLogic
{
     [Serializable]
     internal sealed class ParserException : Exception
     {
          public ParserException()
          {
          }

          public ParserException(string message) : base(message)
          {
          }

          public ParserException(int line, int col, string file, string message) : base(string.Format
("{0} at line {1} col {2}{3}", message, line, col, file != null ? (" in " + file) : "."))
          {
          }

          public ParserException(int line, int col, string file, string format, params object[] args) :
this(line, col, file, string.Format(format, args))
          {
          }

          public ParserException(int line, int col, string file, string message, Exception inner) : base
(string.Format("{0} at line {1} col {2}{3}", message, line, col, file != null ? (" in " + file) : "."),
inner)
          {
          }

          [SecurityPermission(SecurityAction.Demand, SerializationFormatter = true)]
          private ParserException(SerializationInfo info, StreamingContext context) : base(info, con-
text)
          {
          }
     }

     // Thread safe if Parser instances are not shared across threads.
     internal sealed partial class Parser
     {
          public Parser()
          {
               m_nonterminals.Add("Start", new ParseMethod[]{this.DoParseStartRule});
               m_nonterminals.Add("Expression", new ParseMethod[]{this.DoParseExpressionRule});
               m_nonterminals.Add("Operation", new ParseMethod[]{this.DoParseOperationRule});
               m_nonterminals.Add("OperationBracketed", new ParseMethod[]
{this.DoParseOperationBracketedRule});
.
.
.
          }

          public Operation Parse(string input)
          {
               return DoParseFile(input, null);
          }

          // File is used for error reporting.
          public Operation Parse(string input, string file)
          {
               return DoParseFile(input, file);
          }
          .
          .
          .
          #region Private Types
               .
               .
               .
          // The result of parsing a literal or non-terminal.
          .
          .
          .
          #endregion
.
.
.
}
```

Figure 47 Machine Generated Parser Code

## Code: Hand written Code

```
using System;
using System.Collections.Generic;

namespace PolicyLogic
{
    // This is the hand-written code used by the semantic actions for the
    // machine generated parser.
    internal sealed partial class Parser
    {
        // Create an operation from a binary operation
        private OperationFunction DoCreateOperationBinary(List<Result> results)
        {
            OperationFunction result;

            result = new OperationFunction(results[1].Text);
            result.AddValue(results[0].Value);
            result.AddValue(results[2].Value);

            return result;
        }


        // Create an operatino from an n-ary operation
        private OperationFunction DoCreateOperationNary(List<Result> results)
        {
            OperationFunction result;
            int Count;

            result = new OperationFunction(results[1].Text);
            for (Count = 0; Count < results.Count; Count += 2)
            {
                result.AddValue(results[Count].Value);
            }

            return result;
        }

        // Create an operation from an n-ary function
        private OperationFunction DoCreateFunctionNary(List<Result> results)
        {
            OperationFunction result;
            int Count;

            result = new OperationFunction(results[0].Text);
            for (Count = 2; Count < results.Count; Count += 2)
            {
                result.AddValue(results[Count].Value);
            }

            return result;
        }

    }
```

Figure 48 Hand Written Code

The code in Figure 47 is machine-generated as a result of using peg-sharp 0.3.427.0 from Parser.ged to enable the proper functionality of the parser function used in creating the policies from the user interface. The code fragment shown in Figure 47 first creates a Parser Exception function to throw out any errors or exceptions when parsing. It also creates few other parser functions and takes only a string to be parsed as input and another one that takes in both a string and a file as parameters. The file parameter is used for error reporting purposes. The code shown in Figure 48 on the other hand is hand-written. It creates a couple of operation functions that create an operation from an n-array.

### 6.4.5 THE POLICY CREATOR

The policy creator interface is the main user interface for interacting with users. It allows users to define policies mainly from given property files, by loading the required property files. Users can also introduce new properties by using the free text entry option of the interface. Figure 49 presents both the text free entry and property based policy interface designed for the use of mobile device, while Figure 50 and Figure 51 present the web based policy creator interface. Another functionality of the web based interface is the ability to create sub-functions within a policy, which can be later reused. Users can also save these sub-functions or the whole policy. As a user is creating a policy, the resulting policy is shown in the info section part of the interface. The user can easily delete rules within a policy from the middle section of the interface as well as entering new rules for sub-functions using the Enter tap. The policy can be navigated based on the level shown in the info section of the interface. The code used for configuring the interface in Figure 49 is presented in Figure 52.

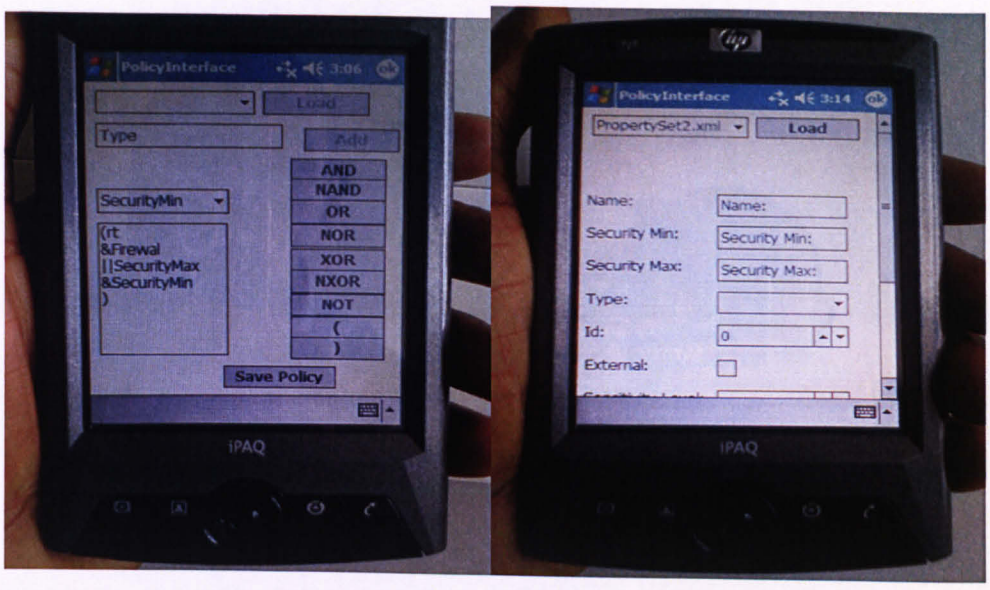


Figure 49 PDA based policy interface (text free entry based on property file)

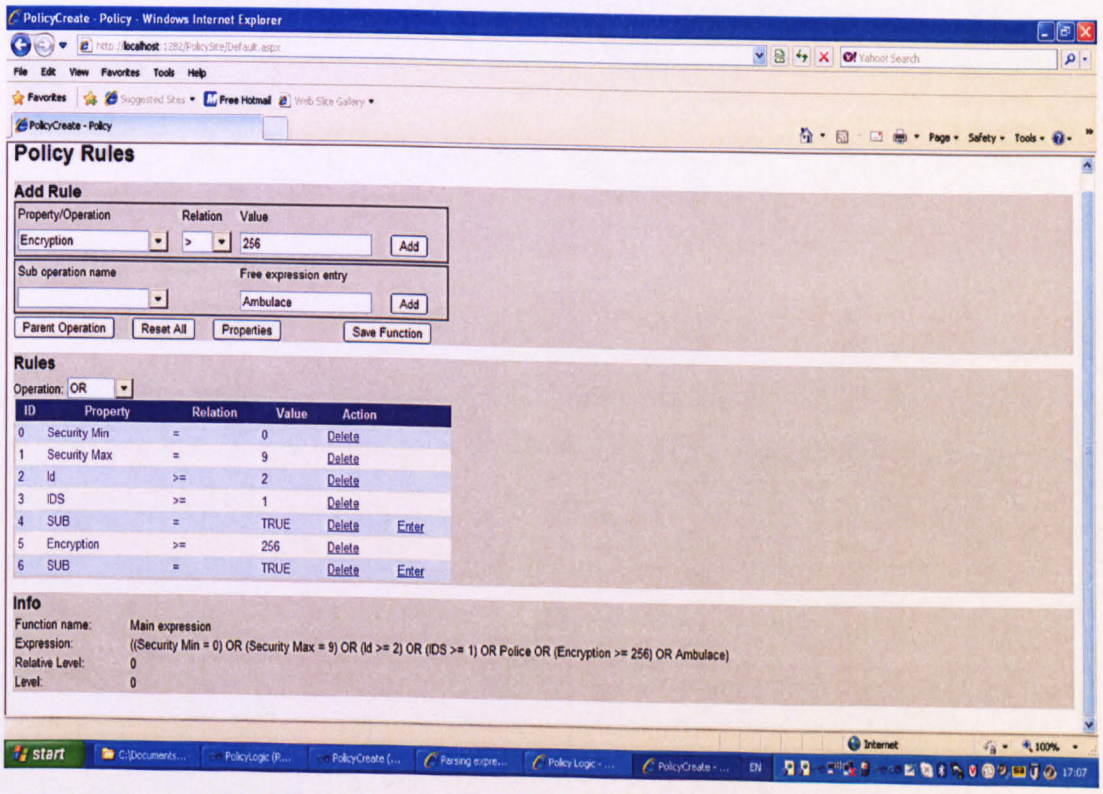


Figure 50 Web Based Policy Creation Interface

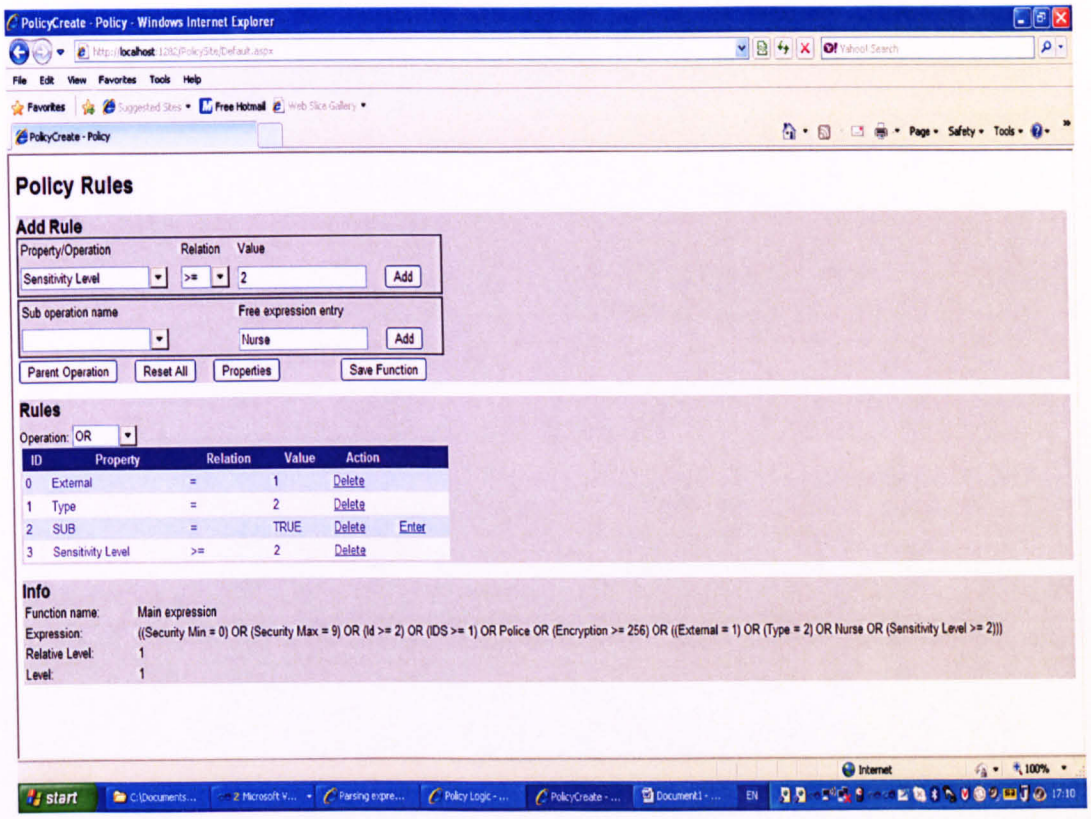


Figure 51 Web Based Policy Creation Interface for Sub-Functions Level 1

The code presented in Figure 52 is used to configure each node or device automatically when the modules of the framework are loaded. It first checks the version or type of the operating system (OS) the device is running. This enables the framework to decide how to configure each device. It then sets the relevant file path to locate the required property set file for the device. It then creates an XMLNodeList to contain all the XML attributes, which represents the properties in the property file. It also reads the file into a file stream reader and goes through each loop statement to configure the devices with all the relevant properties found in the file. The final device configuration is then displayed to the user.

```
Code: Configure UI
//Configures key components of the UI
    //this includes the combo box that holds the Properties
    //this combo box is filled with Controls that are then
    //configured.
    private void ConfigureUI()
    {
      PropertiesCB.DisplayMember = "Name";
      string filePath = "";
      //load properties
      if (System.Environment.OSVersion.ToString().StartsWith("Microsoft Windows NT"))
        filePath = "";
      else filePath = "\\Program Files\\PolicyInterface\\";

      XmlNodeList nodeList = FileStreaming.GetXmlNodeList(filePath + propertySet, "/PropertySet/Property");

      for (int pos = 0; pos < nodeList.Count; pos++)
      {
        foreach (XmlAttribute attrib in nodeList[pos].Attributes)
        {
          switch (attrib.Name.ToLower())
          {
            case "type":
              switch (attrib.Value.ToLower())
              {
                case "binary":
                  CheckBox cb = new CheckBox();
                  cb.Name = nodeList[pos].Attributes["id"].Value.ToString();
                  //NewCB.Visible = true;
                  //NewCB = nodeList[pos].Attributes["id"].Value.ToString();
                  cb.Location = new Point(92, 59);
                  //NewCB.Location = new Point(92, 59);
                  PropertiesCB.Items.Add(cb);
                  //PropertiesCB.Items.Add(NewCB);
                  cb.Visible = false;
                  //NewCB.Visible = false;
                  break;
                .
                .
                .
              break;
          }
        }
        if (PropertiesCB.Items[pos] != null)
          this.Controls.Add((Control)PropertiesCB.Items[pos]);
      }
    }
```

Figure 52 Configure UI

## 6.5 SECURITY MODULE FOR HOME GATEWAY FRAMEWORK

In this section, we present another contribution of our work by creating an addition to another framework developed by a colleague in the School. The framework is aimed at and making good use of home appliances in a peer to peer network (refer to our published work for full details [138]). The design and implementation of the framework lacks the security requirement to insure that home appliances can communicate in an effective and secure manner. Hence, using our methodology discussed in chapter five, we added new modules that provide this required security.

The framework is presented in Figure 53, where peers A-L and R-V are members of the normal peer network, while peers M-Q are acting as a Gateway Peer Overlay Network for the peers. As the state of the system is dynamic and

constantly changing, it is possible that the network might later grow considerably and one of the peers in the P2P network may become part of the Gateway Peer Overlay Network to provide 'gateway' functionality to a group of peers who wish to share and offer ad-hoc services. As mentioned earlier, the overlay network forms a logical layer on the top of the Internet utilising the Internet base transport protocols. In this context, our overlay network of gateway peers belongs to the same level as a peer but their functionality acts as a logical layer above the normal peer network.

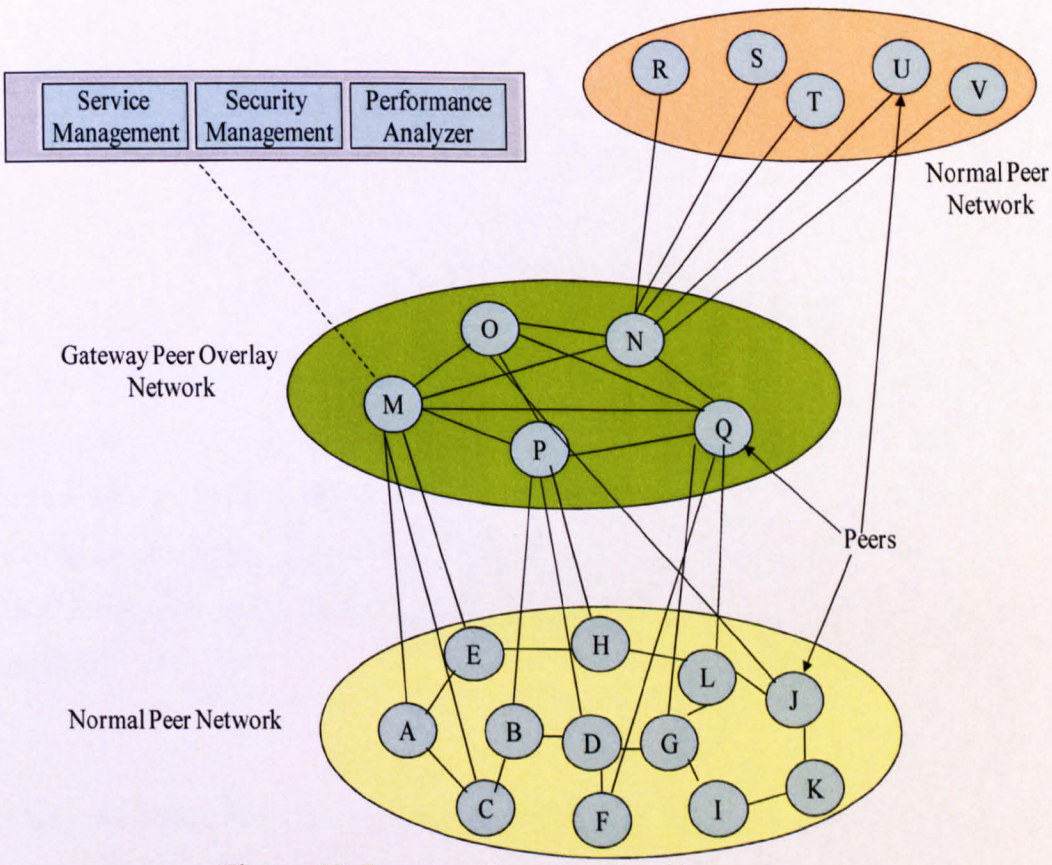


Figure 53 Gateway Peer Overlay Network

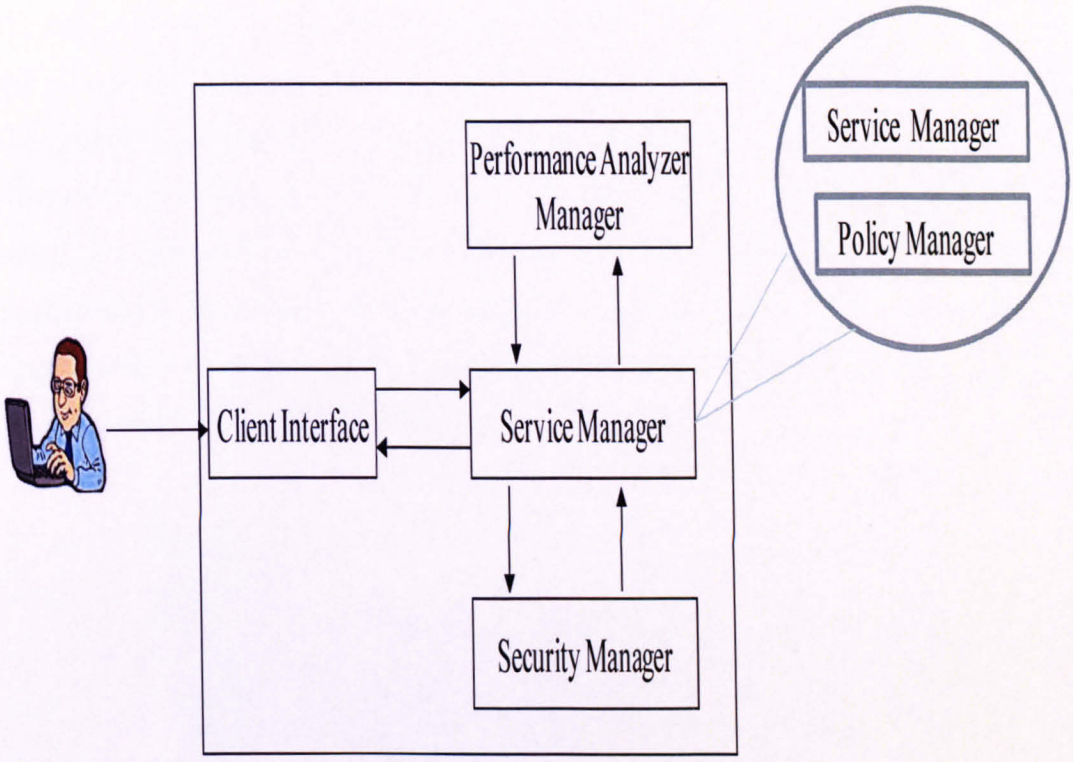

Figure 54 Policy Manager

Our enhancement is in the form of the process of controlling access to services to only authorised devices within the environment. To do so, we extended the framework to incorporate a policy manager within the service manager as shown in Figure 54. When a service is requested by a device, the service manager will first check the policy relating to the service before relinquishing the service to the device.

Here, we allow the home owner to be able to define policies that will decide automatically and dynamically who is allowed to access his/her music sharing environment (see Figure 55). The implantation shown in Figure 55 is an extension to the work done by Muhammad et al [138], by adding a module that extends the used UPnP Device template in their work. The required policy(s) is defined for each device via the extension of the XML Devices Description used by UPnP DeviceType: V Device Template as shown in Figure 55. Here, the policy can be defined as a policy set for each device. In Figure 55, we have defined policy list elements that have a list of possible service rules for the home network. When other devices are trying to access the service provided by a certain device, the device service policy as shown in Figure 55 will first be consulted to make sure

that there is no violation of the device policy before granting or denying access to the requested service. The implementation of a user interface that will allow such definitions and translations of input from the interface into an XML policy file was discussed in section 6.4 when we dealt with the dynamic policy creation. In the case of a personal usage of the framework, i.e. within a home or personal environment, the definition and modification of a policy can be based on a simple user interface as shown in Figure 49 using the mobile application version or web service based one as shown in Figure 50.

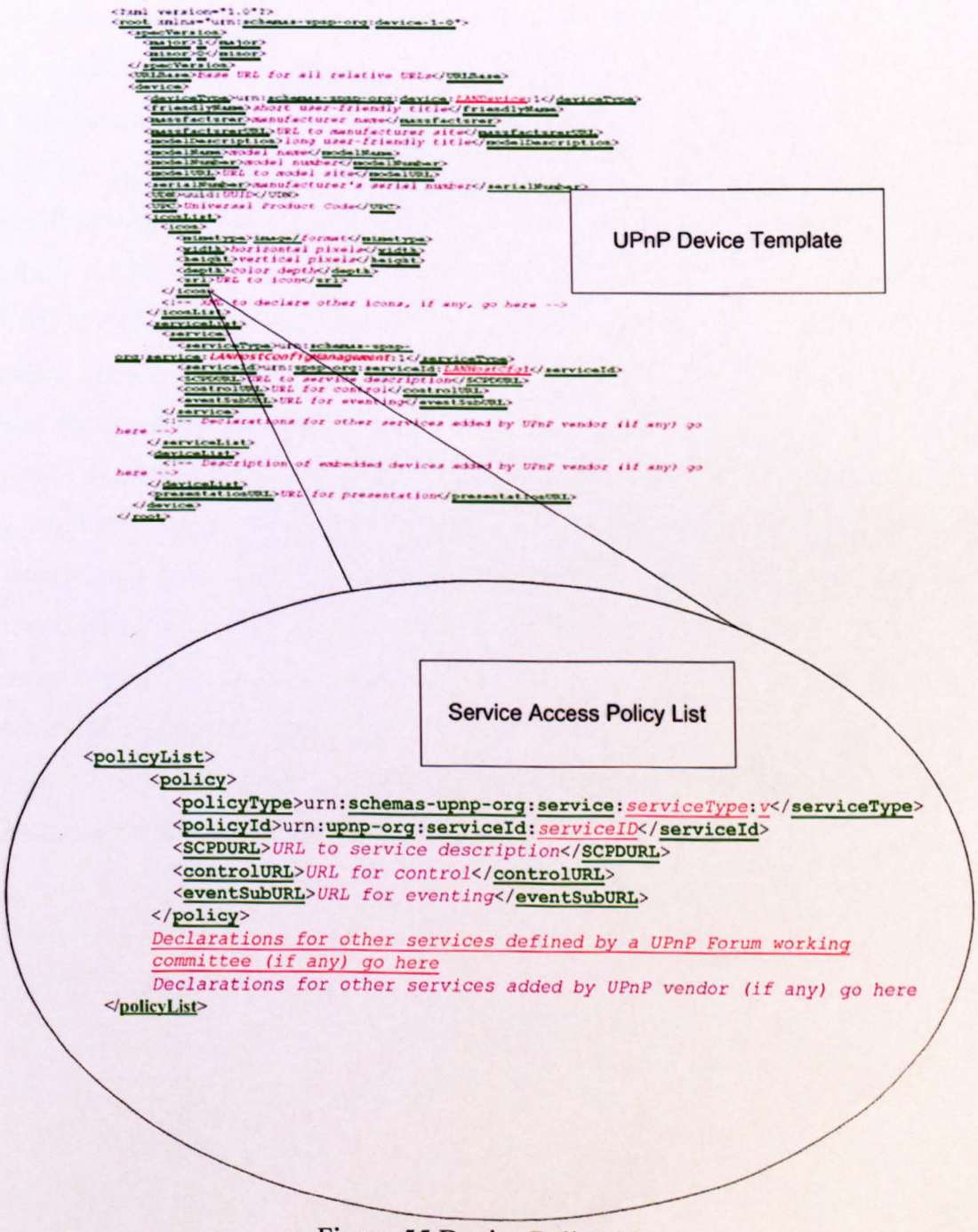


Figure 55 Device Policy List

## 6.6 SUMMARY

In this chapter, our main focus is on the implementation of UCIM together with its sub-modules and provides the results from our lab test. We have presented the implementation phases, simulation of the framework, screenshots as well as some code fragments of the modules of UCIM. Firstly we have explained the implementation phases including a motivational scenario(s) used for the implementation. We have also provided a more in-depth reasoning behind our chosen methodology and presented some of the work that has been carried out to help with the understating of the framework. We have demonstrated through simulations that our proposed framework and methodology have provided a more user-centric and context-aware interface, our policy creation tool has provided a more dynamic and portable application compared to other existing schemes as well as solving the problem of using nested brackets in terms of policy specifications. One of the lessons that can be drawn from this chapter amongst others is that our UCIM framework provides a dynamic modular architecture. Each module is fully functional on its own, and it is portable and can be used as added features to other solutions as demonstrated in section 6.5. The solutions and methodology are applicable to other environments in addition to ad-hoc ones, but the main design is ad-hoc centric. Our simulation results have provided a proof of concept and an implementation of identity and personal information management both within social and critical infrastructures crisis management situations. The creation of policies can be achieved in a dynamic way and in a more user centric manner, as we have provided a means of separation from the technical aspects and eliminated the existing problem of creating and modifying policies. The next chapter will provide more details on the evaluation of UCIM based on the laws of identity and lab results.

# CHAPTER SEVEN

## EVALUATION

In this chapter, we start by reviewing the requirements of identity management in MANets and ubiquitous computing in general. We then evaluate the performance of UCIM against the requirements based on the Law of Identity and lab test results of the applications developed. We have also made use of scenarios to demonstrate and evaluate the functionality of our proposed methodologies as presented in chapter five. A successful identity management system in ubiquitous computing must have the following features: context-awareness, user-centricity, dynamic user policy specification, real-time device configuration, scalability, adaptability, resource efficiency, security, privacy assurance and effectiveness. Comparing with existing solutions, UCIM has addressed all of these issues from the start of its design. For example, it achieves user-centricity by giving users an interface to define policies in relation to available contextual information. UCIM also considers capacity constrained nodes in MANets by adopting proxies, where a novel hybrid metric is designed to balance the system resources for proxy selection.

### 7.1 Analysis of Proposed Framework

In this section we analyse the IM module of our framework based on the seven laws of identity which has been proposed by *Cameron* [113]. This provides a good starting point for our analysis criteria. Before that we first re-cap on the key features of UCIM.

The framework has capitalized and addressed some of the weaknesses identified in the related work section (chapters three, four and five). For example, the generic police toolkit developed by *Ricardo [139]* relies on the role of a system administrator. Its policies and rules are statically defined and require some level of knowledge before such definition. In UCIM, we extend the rule and policy definition by eliminating the role of the system administrator and allowing users to define policies themselves via the use of an intuitive graphical interface.

The overlay networks proposed by *Chen et al. [86]* divide nodes to perform various independent tasks, and each node depends on the existence of another, which causes the problem of a single point of failure. UCIM allows each node within the topology to have the full functionality of the framework. Although this might require more resources compared to having individual nodes performing separate functions, it allows individual nodes to only query nodes that are next to them for required information, eliminates the need of an Segment-Tree Virtual Network (STVN for short), that is used to construct the overlay network, and hence reduces computation power for improved efficiency. We also make use of the push method only during an initial connection so that all nodes can receive real-time information. This reduces the effect of heavy overheads in terms of data transfer that has been one of the drawbacks of the framework by *Chen et al.* Also the push approach is applied to individual nodes based on their demands, and when real-time information is available users are notified. Furthermore, we make use of XML schemas for storing and acquiring information in a more portable way, compared to the conventional usage of databases, to reduce the level of processing and storage needed, which is one of the requirements of devices with less processing power.

Additionally, in UCIM we apply the concept of RBAC but put users in control of access policies and rules. This is achieved by integrating contextual information (mainly users' status/commitments) in an access decision process instead of using traditional RBAC. Hence, users' contextual information and roles have a significant and active influence on the access decision making module. Hence, we call our approach to access control as Contextual Based Access Control (CBAC), which takes context information (mainly the users' statuses) as a deciding factor of either allowing users to access the required information or denying such access. We further extend the idea of CBAC by developing a new dynamic policy interface tool that eliminates the problem of tracking a complex mixture of conjunctions and disjunctions.

The 7 laws of identity by Cameron [113] try to define a unifying identity Meta system for addressing the problem with an absent identity layer of the Internet, which makes it vulnerable. Hence, these laws try to define a new standard that identity providers and researchers should try to adhere to in order to eliminate the problem or the side effect that undermines most technological development. These laws are developed and enhanced through various discussions on Cameron's blog that involves various readers and developers from both the academia and industry. Hence, as widely debated and adopted laws [140], it is of paramount importance that we try to compare and evaluate our framework in accordance with them. Although the laws are mainly developed for the Internet, we can see their application to MANet and SoS composition environments that we are working in. These laws and the related discussion on the blog provide a wider concept on the needs and importance of protecting personal information and identity management. Small devices are often used in MANets and SoS as well as other smart environments, and they are interconnected with access to the Internet. The current trend of the Internet development is moving towards Internet of Things (IoT) to allow small and smart devices to be interconnected seamlessly. The vulnerability of identity theft and misuse of personal information in such environments more critical than or as dangerous as it is on the Internet.

***Law 1: User Control and Consent:*** *An IM system must put users in control of what digital identities are used and released, protect users against deception, and verify the identity of any party who asks for user information.* We have defined user control as being able to determine when, where, how and whom their information is sent to or viewed by. Hence, when a user joins a network with a given set of profile information, he/she has permitted other users within the network, who meet the set rules and policies with respect to the given profile information, to view such information. Hence, we are able to meet this law as well as eliminate the multiple consent solution proposed by Eap et al. for service oriented architectures [56]. The policy interface, property interface, data mishandling and profile building in section 6.1.3 provide more details on how to give users full control of their digital identities. Consequently, our implementations as demonstrated in Figure 29, Figure 35, Figure 43, Figure 45, Figure 46, and other

results in chapter 6 provide the proof of the framework's concept. It is worth pointing out that we have not tested the issue of deception as stated in this law.

***Law 2: Minimal Disclosure:*** *An IM system should limit the disclosure of identity information for a constraint use, in case of a security breach.* The domain centric IM research strongly supports this view by applying the principles of anonymous access in an attempt to prevent the correlation of IDs. In our proposed framework, we also provide the facility of enabling a user to specify which of their personal information should be included within the profile for other users to access or view, using the policy interface as described in section 6. Hence, UCIM not only supports anonymity but also limits the amount of personal data to be disclosed (see section 5.5.3 and Figure 15). These are achieved by allowing users to define and have different profile types (see Figure 18 and section 5.6.1) and decide which to use at any given time to best suit their commitments. Our notion of allowing users to create and impose an expiry date/time for their information fully provides the requirement of this law.

***Law 3: Justifiable Parties:*** *An IM system must only disclose identity information to parties having a necessary and justifiable place in a given identity relationship. Cameron* has not discussed the issue of trust. Instead, he suggests that users should determine whom they can trust, and the IM system should provide the necessary information for the users to make these decisions. Hence, in UCIM this law is used to determine that the user should decide whom they should trust and when a particular profile is made available within the network. This implies that any user that fulfils the set rules and policies of the profile, as defined in the policy interface module in sections 5.4.1, 6.4.4 and 6.4.5 as well as in Figure 13 and analysed by MATTS as presented in section 6.2 and Figure 32 using the defined polices, is trusted.

***Law 4: Directed Identity:*** *An IM system must support both omni-directional (public) identity to facilitate the discovery and unidirectional (private) identity to prevent unnecessary release of correlation identity information.* In our framework we provide the ability of dividing personal information into partial identities and sub-profiles as presented in section 5.6.1 and Figures 16 and 17. Hence users are able to separate which information about them should be made public and private.

***Law 5: Pluralism of Operators and Technologies:*** *An IM system should be able to work with multiple identity providers (IdPs).* This requires that an IM framework must support multiple IdPs, but this does not imply that a user must have more than one IdP. For this particular law, we have not tested our framework to see if it is capable of working with multiple identity providers. But based on our simulation, users can define various identities that are associated to them and interact with other users who are from different agencies. This principle is demonstrated in the crisis management scenario presented in section 6.1, and the results of the implemented scenario presented in Figures 20 and 24.

***Law 6: Human Integration:*** *An IM system must include human user to be a component of the distributed system integrated through unambiguous human/machine communication mechanisms.* In UCIM user participation is a central point of the framework design, and this design principle is reflected in the user control and consent of the UCIM system as well as users' role in defining rules, policies and profile information. Some relevant details of the user interaction as defined within the various UCIM modules are presented in sections 6.4.3, 6.4.4 and 6.4.5 as well as in Figure 13.

***Law 7: Consistent Experience across Contexts:*** *An IM system must provide users with a simple and consistent experience while enabling separation of contexts through multiple operators and technologies.* UCIM provides consistent user experience across multiple contexts via the use of a simple user interface. Also users are able to access resources and contextual information from other users/nodes after successfully establishing connection within the network topology. Our user interface implementation of the framework provides the required consistency for the user. For example, the user interfaces in Figures 45, 46, 49, 50 and 51 show the consistency in terms of usage with similar keywords and interface layouts for user defined policy interfaces. We have tried to achieve such consistence on both device based and web based interfaces. Also the same property interface is used for both the policy interface and the two groups (property and policy) interface, designed to be consistent. This also applies to the simulation environments as presented in Figures 27 and 28, while making use of icons in both the implementation and scenario diagrams.

## 7.2 Simulation and Lap Test Results

In the first part of this chapter our focus was on evaluating UCIM based on the seven laws of identity. This section of the chapter turns the focus to evaluating UCIM based on the lab test results as shown in chapter six. Before going into the details, our results are divided into two main sections: the use of simulated nodes/devices within our in-house developed simulator, and using real devices, i.e. PDA's, HTC mobile phones and desktop PCs. We have presented in chapter six how to make use of these simulated nodes and real devices.

### 7.2.1 Simulation

For simulating our framework, we make use of MATTS to create virtual devices and create connections between them either manually or automatically. A given scenario can be loaded dynamically using an XML script that contains basic details of the required scenario. Each device is automatically configured based on the given details in the XML scenario file. However, devices/node properties can be modified after the scenario has been loaded so as to create situations in which the security of the devices can be compromised while allowing the devices to make proper use of contextual information. An example of a simulated environment based on MATTS with nodes representing a crisis management scenario introduced in chapter six is shown in Figure 27 and Figure 28 respectively. A portable version of the implementation on small portable devices with various virtual icons representing individual devices is shown in Figure 35. Also profile building and data misuse scenarios are illustrated in Figure 36 and Figure 37.

There is no limit to the number of either virtual or real devices that can be connected to the MATTS simulator, although our trials only used a maximum of ten virtual devices and up to three physically connected devices. Any of the virtual devices can also be configured to behave as an external device. We have not noticed any connectivity or performance impact as a result of increasing the number of devices (maximum of ten virtual devices are used) in our simulator.

However, when the number of devices is increased significantly, it is possible to have some impact on the connectivity or performance.

We have also deployed our solutions to a real test bed using a scenario to evaluate the functionality and usability of the systems developed. Figure 56 provides a 100 square meter representation of our outdoor experiment using a mobile ad-hoc network on five netbooks representing individuals from the CoI as discussed in section 6.1.2.

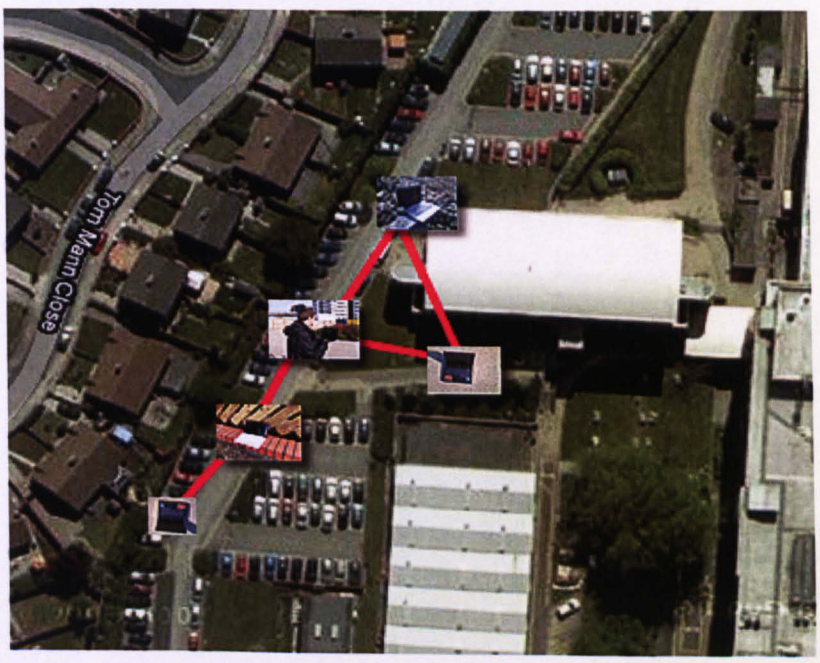


Figure 56 Mobile ad-hoc Network in a 100 Square metre area using five netbooks

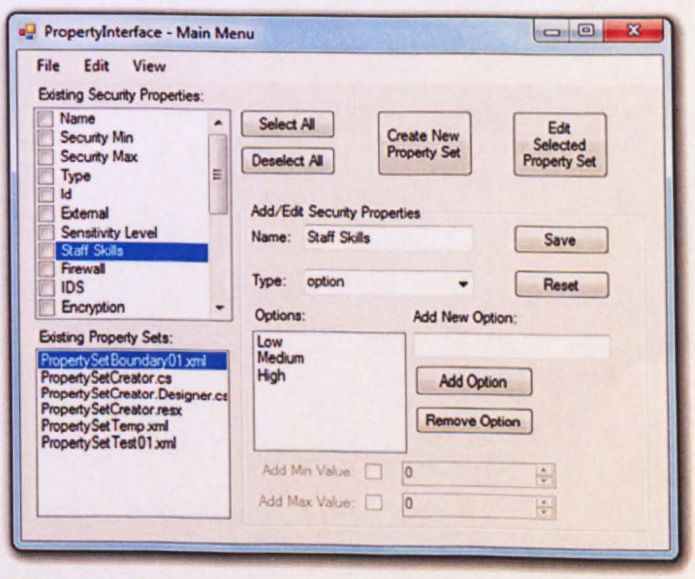


Figure 57 Property Interface for Defining Node Properties

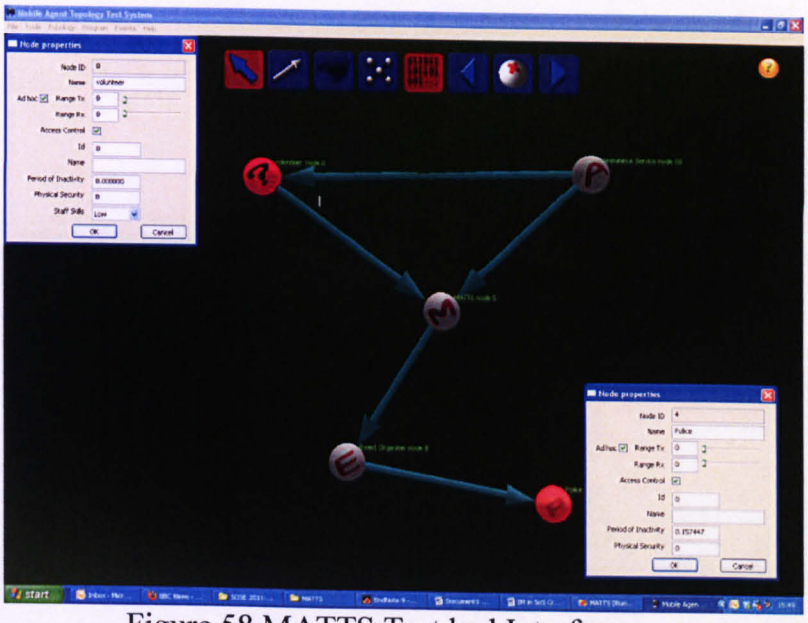

Figure 58 MATTS Test bed Interface

### 7.2.2 PORTABLE DEVICES

As it has been hinted above, not only have we been able to implement UCIM on portable devices but also we are able to run a simple simulation of eight virtual devices based on the scenario and implementation of the ContextRank algorithm and profile building. Although the implementations of the algorithm is based on using two constraint variables: the user profileType and its location (using the screen size of the device and the parameter for the surrounding contextual information), we can see no reason why the portable version of the simulation will not work with more variables or a more complex policy as well as more than eight virtual or real devices.

The proposed framework and its implementations have also been deployed and tested using various portable devices either as real devices or using the available device emulators in the .NET Compact development environment. Figure 59 shows an example of some devices on which we have deployed our framework and tested it. We have also used the Samsung mobile emulator deployed on three versions of HTC phones, a PDA emulator shown in Figure 37 and a real PDA shown in Figure 49. Hence, the scope of the portability and heterogeneity of

UCIM has been demonstrated. We can see no reason why it can't be deployed and used on other kinds of portable devices, desktops, laptops or netbooks.

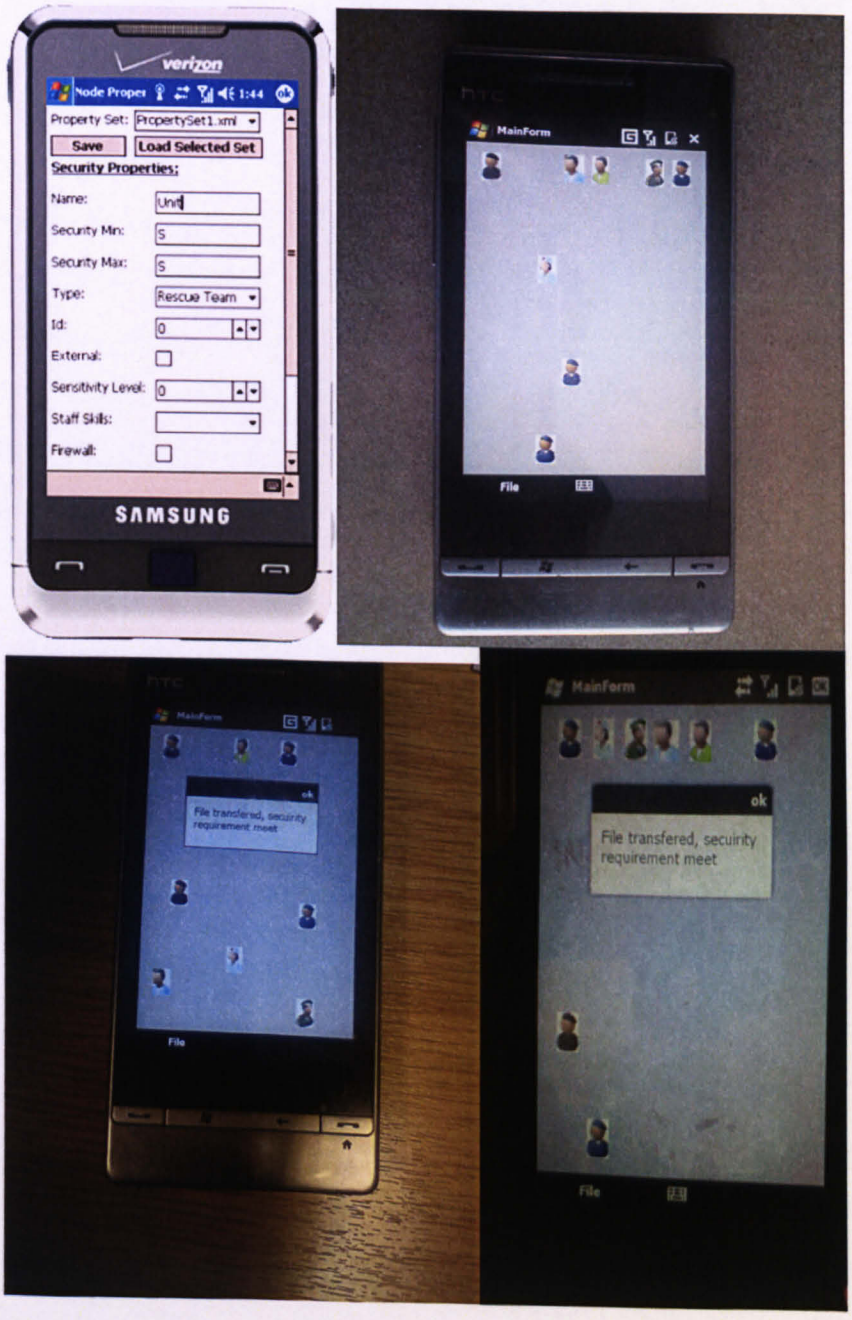


Figure 59 UCIM deployed in Various Portable Devices

### 7.2.3 DEVICES COMMUNICATION

In our current implementation of UCIM, external devices are connected to each other or the MATTS simulator by making use of the available wireless technology communication within our laboratory rather than ad-hoc protocols, while devices in the virtual simulation environment are either connected using a link

automatically or manually. We have developed a simple socket connection mechanism to allow devices to send information within the simulator to real devices. Hence, users are allowed to select the required information to send, then create a socket connection and send the file or information. There are drawbacks to this form of communication but our thesis is not aimed at the development of communication or ad-hoc network protocols. Hence this can be considered as something for future work to help improve the efficiency and communication security of the framework. Currently we have also developed an ad-hoc test bed based on Ad-hoc Wireless Distribution Service (AWDS) wireless multi-hop routing protocol and have tested the modules of our framework for functionality. AWDS is a Layer 2 routing protocol for wireless mesh networks. It provides transparent Ethernet-like access to all participating nodes, thus easily allowing the deployment of different higher-level protocols like IP (with DHCP), IPv6, AppleTalk.

### 7.2.4 Security Analysis

In this section we analyse UCIM and its mechanism in terms of providing a secure platform for identity management within ubiquitous environments. This is done by rendering a tabular contrast of UCIM with reference to other related frameworks or systems as discussed in previous chapters. The comparison and analysis is mainly focused on the key security features and aims of UCIM to provide a more secure platform for ubiquitous environments. In order to achieve that, we have explored some of the factors that will enhance user adaptability and encourage them to put more focus on security related issues. These include amongst others: interoperability, user-centricity, context-awareness, dynamic policy creation, lightweight, information granularity, identity provision and preservation and authorisation & authentication.

Before going into the details of the comparison of our framework with others, our security analysis makes use of attack scenarios as presented in the implementation section in chapter 6 of the thesis. For the attacker model we assumed that the attacker is able to execute code or misuse the system. It can be an insider or/and

outsider attacker. Consequently, in our scenario in section 6.1 for SoS composition in crisis management, our implementation looks at the possibility of attacks from a volunteer node/device within the composed system as well as other nodes within or outside the boundary of the composed system. Subsequently, the boundary check problem presented in section 6.1 as well as the data mishandling and misuse scenarios provide a proof of concept and security analysis of our framework. The scenario of identity management in SoS also presented in section 6.1 focuses more on the issue of identity management in a real life scenario using a test bed so as to analyse how secure our proposed framework is. To further analyse the security of our framework, we extended the work on a home gateway framework as presented in section 6.5.

Table 4 presents a tabular summary for the comparison of UCIM and other related frameworks. The table lists some of the main features and goals that we initially setup for the UCIM project. Rows in the table represent the names of the frameworks or the name of the authors of relevant proposed methodologies. The columns represent the features we are interested in for providing a secure context-aware and user-centred identity management platform for MANets within ubiquitous environments.

The entries of the table are either a Y, N or ?. Y represents the availability of such a feature within the framework concerned, which is either fully or partially supported or addressed theoretically. N represents a situation whereby such a features has not been supported or even addressed theoretically. ? is used when we are not sure if such a feature is available or not, because none of them have claimed to provide such functionality and we have not seen any other evidence that they have done so. It is worth also pointing out that we have tested or verified it ourselves.

**Expressiveness:** this feature mainly deals with how readable the policies used in such a framework are or if such policies are addressed within the framework concerned. As we can see from Table 4, only our proposed UCIM framework and P3P are able to provide an easy to read and understand policies for most uses. The

rest of the studied frameworks have either not provided any policy usage or offered the policies that are not expressive.

Table 4 Framework Comparison

| Frameworks | Expressiveness | User-centricity | Conjunctions and Disjunctions | Ubiquitous | Context-awareness | Dynamic Policy Creation | Partial Identities | Multiple Identities |
|---|---|---|---|---|---|---|---|---|
| UCIM | Y | Y | Y | Y | Y | Y | Y | Y |
| SOUPA | N | N | N | ? | N | N | N | N |
| RBAC | N | N | N | N | N | N | N | N |
| Ponder | N | N | N | N | N | N | N | N |
| P3P | Y | N | N | N | N | N | N | N |
| KAOS | N | N | ? | N | ? | N | ? | N |
| Personal IM | N | Y | N | N | N | N | Y | ? |
| Google latitude | N | N | N | N | Y | N | N | N |
| Chen et al | N | Y | N | N | Y | N | N | ? |
| EMERGE | N | N | N | N | Y | N | ? | ? |
| Cradock et al | N | N | N | N | Y | N | N | N |
| Helmhout et al | N | N | N | N | Y | N | N | N |
| Google + Project | ? | Y | ? | N | ? | N | Y | N |

The Google+ project has designed and developed its interface with considerations given to the need and requirement of users, but it fails to meet the other security elements listed in the table. For any entry with a question mark '?', there has been no proof or claim by the authors of such frameworks that they have addressed the issues either in their design or implementation.

**User-centricity:** this manly focuses on looking at if the frameworks have been able to put users in full control of defining, editing and using security elements in

terms of protecting users' personal information. Again it is only UCIM that provides such capability. With regards to KAOS, we are not sure (as it is not stated or claimed in the paper) whether it offers the capability, while the rest have not.

**Conjunctions and disjunctions:** this deals with if any of the frameworks has provided users with means of defining policies without requiring the use of nested brackets to distinguish between policy rules using 'and' and 'or', while at the same time making it simple and easy for a lay user to read and define such policies. It is only UCIM that addressed the needs of this requirement; the rest of the studied work has not addressed the issue.

**Ubiquitous:** this refers to if a framework is designed purposely for ubiquitous environments or it can be easily tailored to meet the requirements of such environments. Only UCIM is mainly designed for usage with devices found in ubiquitous environments. We are not sure of KAOS, while the rest target at either the Internet use or specialized applications. And their adaptableness for ubiquitous devices is limited if not impossible.

**Context-awareness:** most of the applications studied have been able to address some elements of contextual information with regards to helping provide good security for users. Six out of the twelve frameworks researched either have provided contextual information to users or can incorporate such functionality. But their considerations do not treat users as the central focus to help display only suitable information to the users. Additionally, five of the frameworks do not support contextual information, while we are not sure about one framework in terms of such capability. On the other hand, UCIM has utilized this feature to allow users to define policies that can filter contextual information based on their requirements and situations.

**Dynamic Policy Creation:** This feature is aimed to give users more control on defining some policies required in terms of the usage of their personal information and identities. This will in turn help improve the security issues of data and information misuse as we have described in the scenario in the previous chapters. It is only our proposed UCIM that provides such capability. The rest of the studied frameworks have all hardcoded their policies within the systems.

There has been no separation between the policies and the actual functionalities of the systems.

**Partial identities:** use of partial identities is of crucial importance in improving the security of systems within ubiquitous environments. This can help prevent attackers from building a full profile of a user on their devices by deleting profile information from the drives or misusing the users' information, when such information has been compromised. It also gives a user total control on which of its partial identities the user wants to make use of at any given scenario, while all such partial identities are unanimously linked to that user. Only one of the studied frameworks provides such capability, and UCIM builds on such work to improve it. We are not sure (no proof or claim made by the authors) if two of these frameworks have addressed this issue.

Table 5 Comparison of Models

| Model | ID Type | Service Composition | Cross Domain | User control | Privacy protection |
|---|---|---|---|---|---|
| Isolated | Single ID | No support | No support | No control | Few and very weak protection |
| Centralized | Single ID | Limited support | Limited support | Few control | Much but weak protection |
| Federated | Multiple ID's | Multi domain support | Nearly fully support | Much control | Much and strong protection |
| UCIM | Multiple ID's | Multi domain support | Yes | Full control | Strong and full protection |

Looking back on the classifications of IM into three main models [47], we now provide a comparison of UCIM with these models, and demonstrate where

UCIM fits within the three categories and the levels of security the models in question provide. This comparison is summarized in a tabular form in Table 5.

In Table 5, isolated models make use of a single identity type, have some support for services composition, cross domain support and user control, and provide few and very weak mechanisms in terms of privacy protection. Centralized models on the other hand provide limited support for both services composition and cross domain support, make use of a single identity type, and provide more privacy protection but still very weak one. A federated model tries to address the limitations of the isolated and centralized models. It makes use of multiple identities, provides services composition, and nearly fully supports cross-domain operation and much stronger privacy protection. Our proposed UCIM can be compared with the federated model, and resolves some weaknesses of the federated model by providing full support for cross domain operation, giving users full control of their identity information, and defining their own policies. The use of user defined policies helps to provide a stronger support for privacy protection.

A more closely related system to compare with our work is the Google+ project [141] that tries to address some of the security issues in terms of who amongst the system users your information is shared with. This is done by providing a means of defining sub-groups called 'circles' of your contacts. As shown in Figure 60, you can create new circles and drag and drop your contacts to a given circle. The circle feature allows you to share what matters with the people who matter most or whom to share it with. It aims to eliminate the shortcomings of the current social networking medium of using just friends. Your contacts can also be added to more than one circle. The whole aim of the Google+ project is to help users to be able to manage their contacts and information to be revealed to each contact within various circles.

However, looking deeply into the Google+ project, the user has no control on defining policies or defining which information each circle member can view. The

circles created are static. The privacy window setting for the Google+ project is very complicated, and it does not provide users with much functionality as depicted in Figure 61. Another drawback of the circle idea is the fact that, whatever you share within a circle goes to all users within the circle, you cannot change the settings dynamically, and you cannot control information flows, data sharing and mishandling between your circles of contacts and others that they might share contacts with.

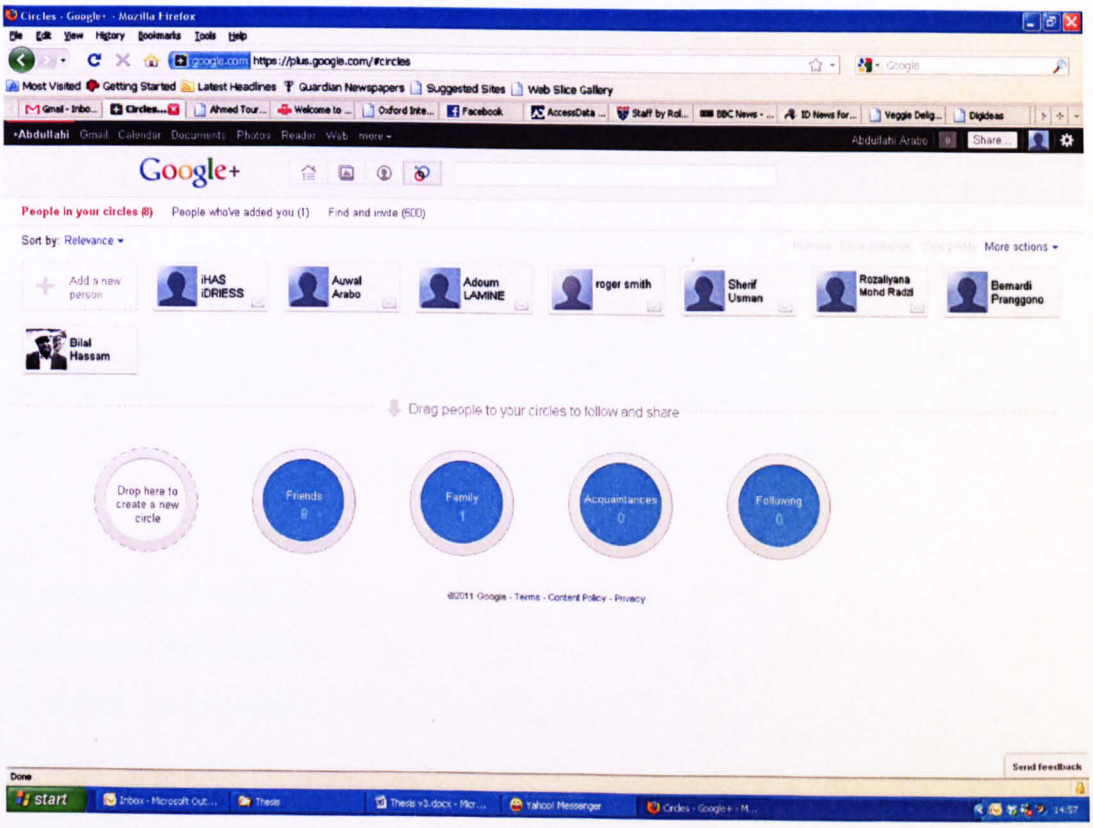


Figure 60 Google+ Circles

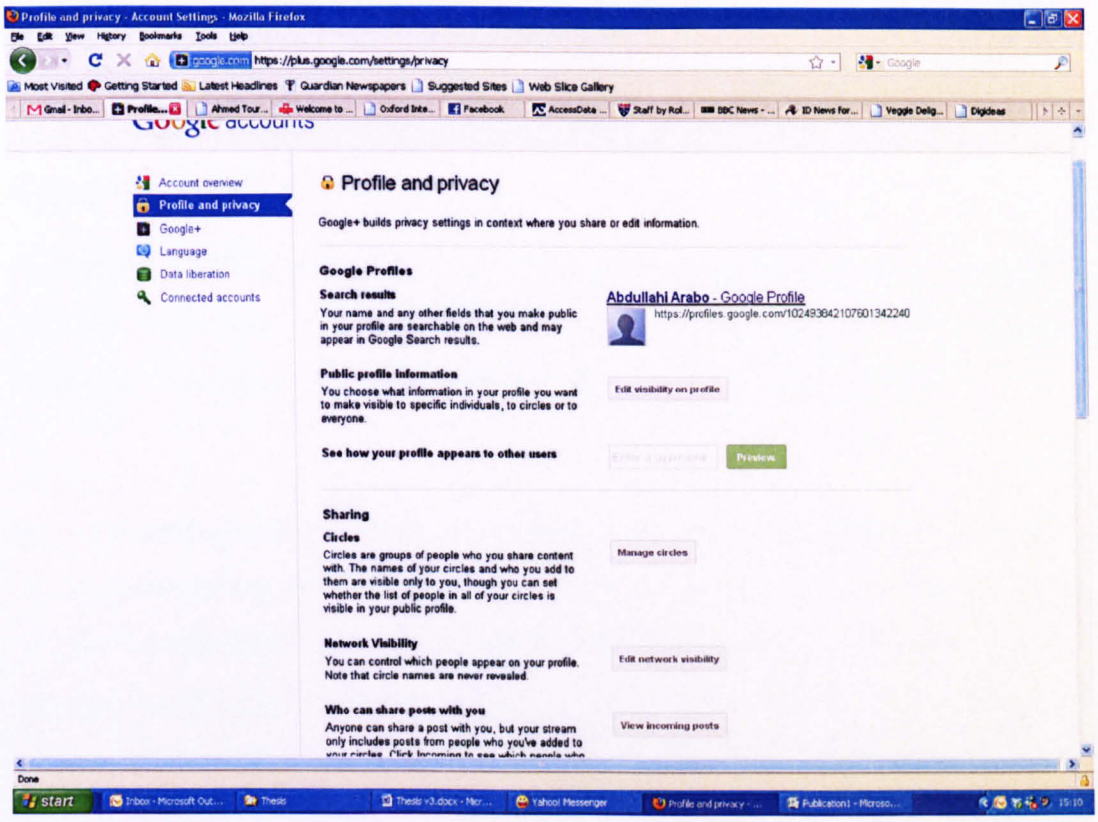


Figure 61 Google+ Profile and Privacy Settings

In comparison with the Google+ project in Figure 61, our privacy interface and policy creation interface in Figures 49, 50 and 51 give users more control on how to define their policies and are more initiative and functional. As the Google+ project allows adding users in more than one circle, this also creates the problem of duplicates of data and possible information misuse, whereas our solution has clearly provided a separation and minimised such s problem. Although the Google+ project has implemented some elements that are related to our proposed solution of restricting access to personal information and managing identities based on who are in contact with whom, our solution is more dynamic than the Google+ project's static circle creation and addition.

### 7.2.5 Reflection on Aims and Objectives

This subsection provides a review of our aims and objectives in comparison to what the project and thesis have achieved based on the implementation shown in chapter six.

We believe that our design and implementation in chapters five and six together with the evaluations in previous subsections can demonstrate that our project has been able to accomplish its aims and objectives as summarized below:

- Heterogeneity: There are thousands of diverse devices in ubiquitous computing environments. The classification of network nodes could help us effectively organise the entire system. Moreover, by cooperating with a context-aware system, UCIM is able to generate contextual information relevant to specific devices. These provide better understanding of the need of individual nodes on both software and hardware levels. The ContextRank algorithm further strengthens this means.
- User centricity and control: By making use of the policy interface model, we have given users the ability to control how their information is used by defining their policies dynamically. The interface itself is a simple one that a lay man can utilise with no difficulties.

## 7.3 Summary

In this chapter we have presented the analysis of UCIM based on the seven laws of identity, test results from our lab and test bed. We have also provided a run through on how UCIM is able to satisfy the set requirements as well as the aims and objectives of the project outlined in both chapters one and five. The chapter has also provided a detailed analysis of the framework in comparison to other frameworks and models. In the next chapter, we will present some concluding remarks and point to areas of future research that might be of paramount importance to improve the framework and meet the expected requirements of future identity management in ubiquitous environments.

# CHAPTER EIGHT

# CONCLUSION AND FUTURE WORK

## 8.1 CONCLUSIONS

The thesis has first established the history of computer networks. From the early years, computers were seen as standalone, stored in large rooms, inflexible, and providing computing resources within localised environs. This situation has changed with the advent of ARPANet in the late 1960's, where a set of computers were connected together to allow remote access to computer resources. This led to a stage where computers are more portable and can be placed on the desk. Since then, millions of computers have joined the network forming the biggest computer society - Internet. By enabling us to shop, work and study remotely, the Internet changes our daily lives in many ways. We have pointed out that the development of computers has undergone three major waves: mainframe computers, personal computers and the introduction of the concept of ubiquitous computing.

The concept of ubiquitous computing was introduced as a prospective view about the future usage of computers. Smaller and cheaper computer chips have enabled us to embed computing ability into any appliances, e.g. a cup, lighter, cloths and even a piece of paper. People's daily activities become closely connected with computers and beneficially become more convenient. The development of computer networks also has rapidly changed, where we have MANets used for various purposes from being limited to the military to a stage where it is available for public usage.

However, the great features of ubiquitous computing inevitably expose its inherent vulnerabilities; our main focus in the thesis is providing a secure identity management framework via context-awareness and user-centricity. The convenience brought by ubiquitous computing could also be taken advantage by intruders, trying to misappropriate users' private and confidential information. It

makes things too easy for malicious people to build a system to spy on others. For example, an intruder may compromise the identity, integrity and confidentiality of an information system by using a stolen ID to modify or access valuable information, or compromise the availability of an information system by possessing the system resources in order to interfere with authorised users' normal access. Like any other computer systems, one of the main prerequisites for the wide adoption of a ubiquitous network is security. The network has to be properly secured so that it can be relied upon. When it comes to the use of convenient devices and before users can agree to adopt such applications, they need to be persuaded that they will be able to have full control of their personal information from misuse. Hence, they can permit the usage of such information, but at the same time they will like to maintain the right to be able to revoke such permissions, keep an eye on how such information is used, etc.

An overview of existing identity management schemes provided in section 3.1 has revealed several weaknesses that hinder their direct application to ubiquitous networks and MANets environments in particular. These shortcomings are caused by their lack of considerations about the heterogeneity, flexibility, context-awareness, user-centricity and resource constraints of ubiquitous networks and devices.

As demonstrated earlier, to overcome the above problems, we have proposed a novel secure context-aware and user-centric identity management system – UCIM. UCIM is a very flexible, adaptive and resource-efficient identity management framework with novel techniques, which utilises contextual information, user policy creation, a service-oriented mechanism and flexible user-centric design. By working together with user defined policies, UCIM can reliably and effectively deter users' personal information from misuse and mishandling as well as giving the user full control of their partial identities by allowing them to define and enforce the expiry date/time of their information. UCIM comprises the following main components: Contextual Information, Privacy Identity Manager and Privacy Manager as the main modules of the framework. It also consists of a user policy

creation tool, utilisation of contextual information to help enforce security and experience of users, a user-control mechanism, a resource-efficient mechanism on how data is stored and transmitted between nodes in the network, and flexible system architecture. Our future work will focus on the further examination of UCIM and the refinement of its models. In the next section, we will also recommend some possible extension of UCIM to incorporate simple practical scenarios, implement it in real-life situations and extend it to issues of cyber security situation awareness.

### 8.2 FUTURE WORK

We have identified the weaknesses of current identity management frameworks and related work as well as provided the requirements for a new generation of identity management solutions that protect ubiquitous computing networks and devices in a resource-efficient way and give users full control of their personal private information/identities. UCIM is presented in this thesis as such a solution. It can provide a number of capabilities as summarised in the previous sub-section. However, this solution is not fully complete, and it also points out other areas that research and developments efforts need to be concentrated on to make the solution more effective and address other identified areas of improvements. More specifically, the future work includes the following main directions:

- Contextual Information Module: as part of our future work, we recommend an implementation of the Contextual Information Module as a standalone server that will store all contextual information as well as act as a backup facility when nodes are out of certain ranges where machine learning techniques will be implemented to help refine the accuracy of users' privacy policies.

- Similarities between two contextual information or profile types: as future work we also recommend extending the algorithm designed in section 5.7 to include ways of calculating the similarities between two objects of the same category (i.e. two contextual information or two profile types) when

they are introduced to the system. This will further enhance the effectiveness and usability of the framework. The extension will also incorporate ways of calculating the ContextRank algorithm by making use of the new similarities function/algorithm.

- Implementation of a simple classroom scenario: currently in the student lab seasons that I am running, it is evident that quite a few students are struggling but at the same time they are shy or self-conscious to ask for help because of various reasons. Another possible application scenario for UCIM will be to incorporate the work of *Chen et al* [85] on paper based leaning with a supporting mobile policy based solution. Particularly the policy interface module of UCIM can be implemented for the student lab scenario, which allows students to use policies to request help and decide who can see it, e.g. only staff or just some students. It should also display if a student can help others on what sections of lab work, etc. Not only will this help evaluate the usability of UCIM, but it will also help students improve their understanding of the lab sessions. This provides a way that UCIM can be further enhanced to handle other similar scenarios and real life events that are not anticipated.

- SoS cyber security scenario for the Ministry of Defence (MoD): one of the main areas of interest in the MoD Defence Science and Technology (DSTL) cyber security call for research proposals is on the area of Enhanced Situation Awareness (ESA). Its target is to provide tools and techniques that will allow computer network defence operators to dynamically query a distributed information system or repository and visualise the results in support of agile risk based decision making in real-time. UCIM has the scope of being enhanced by extending the existing modules and adding few ones so as to provide a proof of concept for the required tool to fulfil the need of future cyber security issues and trends.

- Ad-hoc network protocols: currently we have tested UCIM based on wireless communication technologies, i.e. using the available wireless

network within the lab and the AWDS protocol to connect devices. We think it is important to integrate one of the ad-hoc protocols into UCIM, or even improve the protocol to make it more lightweight and efficient.

It will also be possible to extend the usage of UCIM to other areas like future home networks and services by extending the work that we have done on the secure gateway for P2P. This will incorporate more home networked or healthcare appliances. By doing so, we can further demonstrate the concept of the portability and heterogeneity of UCIM and also evaluate it within various ubiquitous environments.

# REFERENCES


1. Weiser, M., *The computer for the 21st century.* ACM SIGMOBILE Mobile Computing and Communications Review, 1999. **3**(3): p. 3 - 11
2. Arabo, A., et al., *Identity Management in Mobile Ad-hoc Networks (IMMANets): A Survey*, in *9th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (PGNet 2008).* 2008: Liverpool, UK. p. 289-294.
3. Mercuri, R.T., *Scoping identity theft.* Communications of the ACM 2006. **49**(5): p. 17 - 21.
4. Bertino, E., F. Paci, and N. Shang, *Digital Identity Protection - Concepts and Issues.* Proc. International Conference on Availability, Reliability and Security (ARES 09.), 2009: p. lxix-lxxviii.
5. Andersson, C., L.A. Martucci, and S. Fischer-Hübner, *Privacy and Anonymity in Mobile Ad Hoc Networks*, in *Handbook of Research on Wireless Security.* 2008, Information science reference, Hershey, New York. p. 431-448.
6. Bulusu, N., J. Heidemann, and D. Estrin, *GPS-less low-cost outdoor localization for very small devices.* Personal Communications, IEEE 2000. **7**(5): p. 28 - 34.
7. Guanling, C., David, K., *A Survey of Context-Aware Mobile Computing Research.* Department of Computer Science, Dartmouth College; Technical Report: TR2000-381 2000.
8. Sheikh, K., Wegdam, M., Van Sinderen, M. *Middleware Support for Quality of Context in Pervasive Context-Aware Systems.* in *Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07.* . 2007.
9. Stuart, K., Tracy, Camp, Michael, Colagrosso, *MANET Simulation Studies: The Incredibles.* ACM SIGMOBILE Mobile Computing and Communications Review, 2005. **9**: p. 50 - 61.
10. Altman, E. and T. Jienez, *MS Simulator for beginners.* Lecture notes, 2003-2004, University de Los Andes, Merida, Venezuela and ESSI, Sophia-Antipolis, France, 2003.
11. Dr. George F, R., *Using the Georgia Tech Network Simulator.* Department of Electrical and Computer Engineering Georgia Institute of Technology.
12. Lucio, G.F., Paredes-Farrera, M., Jammeh, E., Fleury, M., Reed, M.J. , *OPNET modeler and Ns-2: comparing the accuracy of network simulators for packet-level analysis using a network testbed.* WSEAS Transactions on Computers, 2003. **2**(3): p. 700-7.
13. Terry, D., *Ubiquitous Computing 20 Years Later -(A Whitepaper for the NSF Sponsored Workshop on Pervasive Computing at Scale(PeCS)).* Acessed 12/06/2011, http://sensorlab.cs.dartmouth.edu/NSFPervasiveComputingAtScale/pdf/1569392061.pdf.
14. Vertegaal, R., *Attentive User Interfaces.* COMMUNICATIONS OF THE ACM, 2003. **46**(3): p. 30-33.
15. Peter T., K., *Early Experiences with the ARPANET and INTERNET in the UK.* Department of Computer Science, University College London, 1998.
16. Casanova, F., *Advanced Prototyping Lab and Director of Apple's Advanced Systems Group*, Apple Computer Inc.'s 2010
17. Weinberg, M., *It will be awesome if the don't screw it up*, in *Public Knowledge.* November 2010.
18. Taneja, K. and R.B. Patel. *An Overview of Mobile Ad hoc Networks: Challenges and Future.* in *Proceedings of National Conference on Challenges & Opportunities in Information Technology (COIT-2007).* 2007. RIMT-IET, Mandi Gobindgarh. .
19. Tobagi, F., R. Binder, and B. Leiner, *Packet radio and satellite networks.* IEEE Communication Magazine, 1984. **22**(11): p. 22-40.
20. Leiner, B.M., R.J. Ruth, and A.R. Sastry, *Goals and challenges of the DARPA GloMo program [global mobile information systems].* IEEE Personal Communications, 1996. **3**(6): p. 34 - 43.
21. Royer, E.M. and C.-K. Toh, *A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks.* IEEE Personal Communications, 1999. **6**(2): p. 46-55.
22. Cheswick, W.R. and S.M. Bellovin, *Firewalls and Internet Security: Repelling The Wily Hacker* April 30, 1994: Addison-Wesley Professional Computing- Paperback.
23. Farmer, D. and W. Venema, *Improving the Security of Your Site by Breaking Into It.* Internet White paper. http://gd.tuwien.ac.at/infosys/security/wietsearchive/ admin-guide-to-cracking.101.Z, 1993.
24. Day, J.D. and H. Zimmerman. *The OS1 Reference Model.* in *Proceedings of the IEEE.* 1983.

25. SCMagazine, *Perspective: 20 years of security.* http://media.scmagazineus.com/Archives/11/1109_P_perspective_V2_WEB_ONLY_2671.pdf, 2009.
26. Pfleeger, C.P.C., D.M. , *Security and Privacy: Promising Advances.* IEEE Software. , 1997. **14**(5 ): p. 27-32.
27. Carlson, T., *Information Security Management: Understanding ISO 17799*, in *Information Security Management.* 2001, Lucent Technologies Worldwide Services. p. 1-23.
28. Dewdney, A.K., *Computer Recreations: Of Worms, Viruses and Core War.* Scientific American, March 1989: p. 110.
29. Gartner, G., *Emerging trends and Technologies scenario.* 2003.
30. DNV, R., *Technology Outlook 2003- DNV Research's assessment of development trends.* 2003.
31. Lahey, B., et al. *PaperPhone: Understanding the Use of Bend Gestures in Mobile Devices with Flexible Electronic Paper Displays.*
32. João Pedro, S., David, Garlan, *Aura: An Architectural Framework for User Mobility in Ubiquitous Computing Environments.* Proceedings of the 3rd Working IEEE/IFIP Conference on Software Architecture, 2002: p. 29-43.
33. Satyanarayanan, M., *Pervasive computing: vision and challenges.* IEEE Personal Communications, 2001. **8**(4): p. 10–17.
34. Satyanarayanan, M., *Fundamental challenges in mobile computing.* Proceedings of the 15th Annual ACM Symposium on Principles of Distributed Computing (PODC'96), 1996: p. 1–7.
35. Goeden E., M., *Cramming More Components onto Integrated Circuits.* Electronics 1965. **38**(8): p. 114-117.
36. Martin, S., Thomas, Schoch, *Today's impact of ubiquitous computing on business processes.* First International Conference on Pervasive Computing 2002: p. 62--74.
37. Genevieve, B., Paul, Dourish, *Yesterday's tomorrows: notes on ubiquitous computing's dominant vision.* Personal and Ubiquitous Computing, 2007. **11**(2): p. 133-143.
38. Robinson, P., *An Application-led Approach for Security Research in Ubicomp* A Pervasive 2005 Workshop 11 May 2005, Munich, Germany, 2005.
39. Madhavapeddy, A. and N. Ludlam, *Ubiquitous Computing needs to catch up with Ubiquitous Media.* A Pervasive 2005 Workshop 11 May 2005, Munich, Germany, 2005.
40. K. Fishkin, B.J., M. Philipose, and S. Roy, *I sense a disturbance in the force: Long-range detection of interactions with rfid-tagged objects.* In Ubicomp, September 2004.
41. Ciarletta, L., *Emulating the Future with/of Pervasive Computing Research and Development* A Pervasive 2005 Workshop 11 May 2005, Munich, Germany, 2005.
42. Salber, D., Dey, A.K., Abowd, G.D, *Ubiquitous Computing: Defining an HCI Research Agenda for an Emerging Interaction Paradigm.* Tech. Report GIT-GVU-98-01, Feb. (1998).
43. Müller, G. and S. Wohlgemuth, *Study on Mobile Identity Management.* 2005, FIDIS (No 507512).
44. Alpar, G., J. Hoepman, and J. Siljee, *The identity crisis security, privacy, and usability issues in identity management.* Identity Management on Mobile Devices, January 2011.
45. Ashford, W., *ISSE 2010: Police are playing catch-up as criminals embrace IT*, in *ComputerWeekly.* 2010. p. 3.
46. *University Scorecard Balanced Scorecard Metrics Template*, h.w.s.a.c.s.u.s.e.h.a. 19/01/2011], Editor. 2011.
47. Cao, Y. and L. Yang, *A survey of Identity Management technology.* IEEE International Conference on Information Theory and Information Security (ICITIS), 2010: p. 287-293.
48. Enck, W., et al. *TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones.* in *9th USENIX Symposium on Operating Systems Design and Implementation (OSDI'10).* 2010. Vancouver, BC, Canada.
49. Smith, E., *iPhone Applications & Privacy Issues: An Analysis of Application Transmission of iPhone Unique Device Identifiers (UDIDs)*, in *http://www.pskl.us/wp/wp-content/uploads/2010/09/iPhone-Applications-Privacy-Issues.pdf (online).* 2010.
50. Holwerda, T., *Studies Show Android, iOS Transmit Private Data to 3rd Parties*, in *http://www.osnews.com/story/23865/Studies_Show_Android_iOS_Transmit_Private_Data_to_3rd_Parties (online).* 07/10/2010.
51. BBCNews, *Facebook uncovers user data sales*, in *BBC News Technology http://www.bbc.co.uk/news/technology-11665120 [accessed 06/11/2010].* 2010.

52. Mohammad, M.R.C., Josef, Noll. *Distributed Identity for Secure Service Interaction.* in *Proceedings of the Third International Conference on Wireless and Mobile Communications (ICWMC'07).* 2007.
53. Jøsang, A. and S. Pope. *User Centric Identity Management.* in *AusCERT Conference 2005.* 2005. Australia.
54. Balasubramaniam, S., et al. *Identity management and its impact on federation in a system-of-systems context.* in *IEEE SysCon 2009 —3rd Annual IEEE International Systems Conference.* 2009. Vancouver, Canada.
55. Hermans, B., *Desperately seeking: Helping hands and human touch,* in *Zeist, The Netherlands, http://www.hermans.org/agents2/index.html (last updated on: the 8th of July, 2000., Visited 23/09/2010).* May 1998.
56. Eap, T.M., M. Hatala, and D. Gašević. *Enabling User Control with Personal Identity Management.* in *IEEE International Conference on Services Computing, 2007. SCC 2007.* . 2007.
57. BhargavSpantzel, A., et al., *User centricity: a taxonomy and open issues.* Journal of Computer Security, The Second ACM Workshop on Digital Identity Management - DIM 2006, 2007. **15**(5): p. 493-527
58. Bartolomeo, G., S. Salsano, and N. Blefari-Melazzi. *Reconfigurable Systems with a User-Centric Focus.* in *Proceedings of the 2007 International Symposium on Applications and the Internet Workshops (SAINTW'07).* 2007: IEEE.
59. Claycomb, W., Dongwan, S., Hareland, D. . *Towards privacy in enterprise directory services: a user-centric approach to attribute management.* in *2007 41st Annual IEEE International Carnahan Conference on Security Technology,* . 2007.
60. Altmann, J. and B. Sampath, *UNIQuE: A User-Centric Framework for Network Identity Management* 2006 IEEE/IFIP Network Operations and Management Symposium (IEEE Cat. No. 06CH37765C),, 2006.
61. Bramhall, P., Hansen, M., Rannenberg, K., Roessler, T. , *User-centric identity management: new trends in standardization and regulation.* IEEE Security & Privacy, 2007. **5**(4): p. 84-7.
62. Pashalidis, A. and C. Mitchell, *Privacy in Identity and access management systems.* Digital Identity and Access Management: Technologies and Framework, February 2011.
63. Taylor, N.K., et al., *Pervasive computing in Daidalos.* IEEE Pervasive Computing, January-March 2011. **10**: p. 74-81.
64. B. Schilit, N.A., R. Want, , *Context-Aware Computing Applications.* First Workshop on Mobile Computing Systems and Applications, 1994: p. 85-90.
65. Dey, A.K. and G.D. Abowd. *Towards a better understanding of context and context-awareness.* in *Proceedings of the Workshop on the What, Who, Where, When and How of Context-Awareness.* 2000: ACM Press, New York.
66. G. Chen, D.K., *A survey of context-aware mobile computing research. Technical Report TR2000-381, Department of Computer Science, Dartmouth College.* 2000.
67. Brown, P.J. and G.J.F. Jones, *Context-aware Retrieval: Exploring a New Environment for Information Retrieval and Information Filtering.* Personal and Ubiquitous Computing, 2001. **5** (4): p. 253 - 263.
68. Verkasalo, H. *Contextual Usage-Level Analysis of Mobile Services.* in *4th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MOBIQUITOUS 2007).* 2008.
69. Gregory. D. Abowd, E.D.M., *Charting past, present, and future research in ubiquitous computing.* ACM Transactions on Computer-Human Interaction, 2000. 7(1): p. 29–58.
70. Dan, H., Dickson K.W. ,Chiu, Vincent Y. ,Shen. *Requirements elicitation for the design of context-aware applications in a ubiquitous environment.* in *ACM International Conference Proceeding Series; Proceedings of the 7th international conference on Electronic commerce* 2005: ACM.
71. Wyse, J.E. *Applying Location-Aware Linkcell-Based Data Management to Context-Aware Mobile Business Services.* in *International Conference on the Management of Mobile Business, 2007. ICMB 2007.* . 2007.
72. Alduan, M., et al., *System architecture for enriched semantic personalized media search and retrival in future media internet.* IEEE Communications Magazine, March 2001. **49**(3): p. 144-151.
73. Chai, W.K., et al., *Curling: content-ubiquitous resolution and delivery infrastructure for next-generation services* IEEE Communication Magazine, March 2011. **49**(3): p. 112-120.

74. Choi, J., et al., *A survey on content oriented networking for efficient content delivery.* IEEE Communication Magazine, March 2011. **49**(3): p. 121-127.
75. Dingledine, R., Mathewson, N. Syverson, P. Tor. *The Second Generation Onion Router.* in *Published in Proceedings of the 13th USENIX Security Symposium,.* 2004. San Diego, USA,.
76. Par, J.A., Jonas Sjostrom, *The Principle of Identity Cultivation on the Web.* ECRIM News, 2008. **72**: p. 31-32.
77. Tanachaiwiwat, S., Dave, P., Bhindwale, R., Helmy, A. . *Location-centric isolation of misbehavior and trust routing in energy-constrained sensor networks.* in *2004 IEEE International Conference on Performance, Computing, and Communications.* 2004.
78. Karp, B. and H.T. Kung. *GPSR: greedy perimeter stateless routing for wireless networks.* in *International Conference on Mobile Computing and Networking archive Proceedings of the 6th annual international conference on Mobile computing and networking* 2000. Boston, Massachusetts, United States ACM New York, NY, USA.
79. Moloney, M., Weber, S. *A context-aware trust-based security system for ad hoc networks.* in *Workshop of the 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks, 2005.* . 2005.
80. *Google Latitude.* http://www.google.com/mobile/default/latitude.html (Accessed 05/02/2009).
81. Yahoo, *Introduction to Fire Eagle*, in *http://fireeagle.yahoo.net/developer/documentation.* [Accessed 11/05/09].
82. *IYOUIT*, in *https://www.iyouit.eu/portal/Manual.aspx#privacy.* [Accessed 11/05/09].
83. Hadjiantonis, A.M., Malatras, A., Pavlou, G. *A context-aware, policy-based framework for the management of MANETs.* in *Seventh IEEE International Workshop on Policies for Distributed Systems and Networks, 2006.* . 2006.
84. Davide, F., Gruia-Catalin, Roman. *Context-aware publish subscribe in mobile ad hoc networks.* in *9th International Conference, COORDINATION 2007.* 2007.
85. Chen, G.-D., Chao, P.-Y., *Augmenting Traditional Books with Context-Aware Learning Supports from Online Learning Communities.* Educational Technology & Society,, 2008. **11**(2): p. 27-40.
86. Tzung-Shi, C., Gwo-Jong, Yu, Hsin-Ju,Chen. *A framework of mobile context management for supporting context-aware environments in mobile ad hoc networks*
in *International Conference On Communications And Mobile Computing* 2007. USA.
87. Chen, Y., Schwan, Karsten. *Opportunistic Overlays: Efficient Content Delivery in Mobile Ad Hoc Networks.* in *Proceedings of the 6th ACM/IFIP/USENIX International Middleware Conference (Middleware 2005).* 2005. Grenoble France, November 2005.
88. Gui, C. and P. Mohapatra. *Efficient Overlay Multicast for Mobile Ad Hoc Networks.* in *Proceedings of IEEE Wireless Communications and Networking Conference.* 2003. New Orleans, Louisiana, USA, March 2003.
89. Rachel, F., Giovanni, Iachello, Jehan, Moghazy, Zachary, Pousman, John, Stasko, *The design and evaluation of a mobile locationaware handheld event planner.* MobileHCI, 2003: p. 145–160.
90. Palle, K., Jens, Krösche, Susanne, Boll, *Accessights – a multimodal location-aware mobile tourist information system* International Conference on Computers Helping People with Special Needs(ICCHP) 2004: p. 187–294.
91. Forest, F., et al. *Psycho-Social Aspects of Context Awareness in Ambient Intelligent Mobile Systems.* in *IST Summit - Workshop - Capturing Cotext and Context Aware Systems and Plateforms.* 2006. Myconos.
92. EMERGE, *EMERGE (Emergency Monitoring and Prevention) Factsheet.* 2008.
93. Blandford, A. and B.L.W. Wong, *Situation Awareness in Emergency Medical Dispatch.* International Journal of Human–Computer Studies, 2004. **61**(4): p. 421-452.
94. Rachel, C., David, Harmer, *Emergency services situation awareness*, in *Contingency Today,* *http://www.contingencytoday.com/online_article/Emergency-services-situation-awareness/827*. Accessed 22/04/09.
95. Leen, G., et al., *Novel personal digital support systems for Emergency Responders to a crises occurring in a Critical Infrastructure*, in *The Institute of Electrical and Electronics Engineers, Incorporated (IEEE) pervasive computing - "Pervasive Computing in Hostile Environments".* 2010.
96. Research, A., *Alternative Positioning Technologies.* 2010, ABI Research http://www.abiresearch.com/research/1005214 [accessed 29/10/2010].

97. Badrul, S., George, Karypis, Joseph, Konstan, John, Reidl, *Item-based collaborative filtering recommendation algorithms.* Proceedings of the 10th international conference on World Wide Web, 2001: p. 285 - 295.
98. Mukund, D., George, Karypis, *Item-Based Top-N Recommendation Algorithms.* ACM Transactions on Information Systems, 2004. **22**(1): p. 143 - 177.
99. Page, L., Brin, Sergey, Motwani, Rajeev, Winograd, Terry *The PageRank Citation Ranking: Bringing Order to the Web*, in *Technical Report. Stanford InfoLab.* 1999.
100. Glen, J., Jennifer, Widom, *SimRank: a measure of structural-context similarity.* Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining 2002: p. 538 - 543.
101. Pariser, E., *The Filter Bubble: What the Internet is Hiding from You.* 2011: Viking. 304.
102. Mitchell, G., *Click listeners test 'filter bubble'*, in *Click on BBC World Service: http://www.bbc.co.uk/news/technology-14155845 [accessed 14/07/2011].* 2011.
103. Jason I., H., James A., Landay, *An Infrastructure Approach to Context-Aware Computing.* Human-Computer Interaction, 2001. **16**: p. 287–303.
104. Harry, C., Tim, Finin, Anupam, Joshi, *An Ontology for Context-Aware Pervasive Computing Environments.* The Knowledge Engineering Review, 2003. **18**(3 ): p. 197 - 207.
105. Tao, G., Hung Keng, Punga, Da Qing, Zhang, *A service-oriented middleware for building context-aware services.* Journal of Network and Computer Applications, 2005. **28**(1): p. 1-18.
106. Martin, H., Alisdair, McDiarmid, Allan, Tomlinson, James, Irvine, Craig, Saunders, John, MacDonald, Nigel, Jeeries. *Instant Knowledge: a Secure Mobile Context-Aware Distributed Recommender System.* in *ICT-MobileSummit 2009 Conference Proceedings, Paul Cunningham and Miriam Cunningham (Eds), IIMC International Information Management Corporation.* 2009.
107. James, I., *Instant Knowledge: Leveraging Information on Portable Devices.* 2nd IEEE International Interdisciplinary Conference Portable Information Devices 2008: p. 1 - 5.
108. Nigel, J., James, Irvine, *Virtual Centre of Excellence in Mobile and Personal Communications -Mobile VCECORE 4 Research Area: 'Instant Knowledge' (A Secure Autonomous Business Collaboration Service).* Instant Knowledge Info Sheet from http://www.mobilevce.co.uk/frames.htm?core5research.htm, [Accessed 19/04/2010].
109. Ferraiolo, D.F. and D.R. Kuhn. *Role Based Access Control.* in *15th National Computer Security Conference* 1992.
110. Ravi S, S., Edward J., Coyne, Hal L., Feinstein, Charles E., Youman *Role-Based Access Control Models.* IEEE Computer, IEEE Press, 1996. **29**(2): p. 38-47.
111. R, S., D F, Ferraiolo, D R, Kuhn. *The NIST Model for Role Based Access Control: Towards a Unified Standard.* in *Proceedings, 5th ACM Workshop on Role Based Access Control.* 2000. Berlin
112. Gustaf, N., Mark, Strembeck. *An approach to engineer and enforce context constraints in an RBAC environment.* in *In Proceedings of the Eighth ACM Symposium on Access Control Models and Technologies (SACMAT-03)* Jun 2003. New York: ACM Press.
113. Cameron, K. *The Laws of Identity.* in *http://identityblog.com/*. Accessed 15/10/2008.
114. Sebastian, C., Marit, Köhntopp *Identity management and its support of multilateral security.* Computer Networks: The International Journal of Computer and Telecommunications Networking 2001. **37**(2): p. 205 - 219.
115. Freedman, M.J., Morris, R. *A Peer-to-Peer Anonymizing Network Layer.* in *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002).* 2002. Washington, DC, USA.
116. Rennhard, M., Platter, B. *MorphMix: Peer-to-Peer based Anonymous Internet usage with Collusion Detection.* in *Proceedings of the Workshop on Privacy in Electronic Society (WPES).* 2002. Washington, DC, USA.
117. Arabo, A., Q. Shi, and M. Merabti, *User-Centred Identity Management in Mobile Ad-hoc Networks*, in *10th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (PGNet 2009).* 2009: Liverpool, UK. p. 147-153.
118. Jan E., S., Peter J., Burke, *Identity Theory and Social Identity Theory.* Social Psychology Quarterly, 2000. **63**(3): p. 224-237.

119. Stryker, S., Burke, Peter J., *The past, present, and future of an identity theory.* Social Psychology Quarterly, 2000. **63**(4): p. 284-297.
120. Graf, A., *TP Lex and Yacc - The Compiler Writer's Tools for Turbo Pascal Version 4.1 User Manual.* 1998.
121. Hat, R., *JBoss Enterprise BRMS Business Rules Management System.* http://www.jboss.com/products/platforms/brms/, [Accesses 20/04/09].
122. Ould-Ahmed-Vall, E., Riley, G F, Heck, B S, Reddy, D *Simulation of large-scale sensor networks using GTSNetS.* in *Proceedings of the 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'05).* 2005: IEEE.
123. Thomas, S., Claudi,a LinnhoffPopien. *A Context Modeling Survey.* in *Proceedings of the Sixth International Conference on Ubiquitous Computing - Workshop on Advanced Context Modeling, Reasoning and Management (UbiComp 2004),.* 2004. Nottingham/England.
124. Helge, J., Linda, Finch, *The role of dynamic security policy in military scenarios.* 6th European Conference on Information Warfare and Security, 2007: p. 121-130.
125. Moritz Y., B., *Information governance in NHS's NPfIT: A case for policy specification.* International Journal of Medical Informatics, 2007. **76**(5): p. 432-437.
126. Moritz Y., B.P., Sewell, *Cassandra: Distributed Access Control Policies with Tunable Expressiveness.* 5th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY), 2004: p. 159–168.
127. J. M. , S., *The Z Notation: A Reference Manual*, ed. S. Edition. 1992: Prentice Hall International (UK) Ltd.
128. Pfitzmann, A., *Anonymity, Unobservability, Pseudonymity, and Identity Management – A Proposal for Terminology.* http://dud.inf.tu-dresden.de/Literatur_V1.shtml (v0.5 and all succeeding versions).
129. Dimitris, G., Panayiotis, Kotzanikolaou, Christos, Douligeris. *Preventing Impersonation Attacks in MANET with Multi-factor Authentication.* in *Third International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 2005. WIOPT 2005.* . 2005.
130. Garfinkel, S., Spafford, G., and Schwartz, A, *Practical Unix and Internet Security.* 3rd ed. 2003: O'Reilly Media, Inc.
131. Merabti, M., et al., *MATTS Instruction Manual.* 2007, School of Computing and MAthematical Sciences, Liverpool John Moores University.
132. Zhou, B., et al., *System-of-Systems Boundary Check in a Public Event Scenario.* 5th IEEE International Conference on Systems of Systems Engineering, 2010.
133. Balasubramaniam, S., et al. *Identity management and its impact on federation in a system-of-systems context.* in *IEEE SysCon 2009 —3rd Annual IEEE International Systems Conference, 2009.* 2009. Vancouver, Canada: IEEE.
134. Arabo, A., et al., *Identity Management in System-of-Systems Crisis Management Situation.* 6th IEEE International Conference on System of Systems Engineering SoSE 2011 (2011) 2011: p. 37 - 42.
135. Q. Shi, N.Z., *A general approach to secure components composition*, in *IEEE Computer Society.* 1996 p. Page: 263
136. Portmann, M. and A.A. Pirzada, *Wireless Mesh Networks for Public Safety and Crisis Management Applications* IEEE Internet Computing, 2008. **12**(1): p. 18-25.
137. Drew, O., *Dynamic Security Properties for Systems-of-Systems Security: Documentation.* 2009, School of Computing and Mathematical Sciences, Liverpool John-Moores University.
138. Muhammad, A., et al. *A Secure Gateway Service for Accessing Networked Appliances.* in *Fifth International Conference on Systems and Networks Communications.* 2010: IEEE Computer Society.
139. Ricardo, N., Patr´ıcia, Dockhorn Costa, Maarten, Wegdam , and Marten, van Sinderen. *An Information Model and Architecture for Context-Aware Management Domains.* in *IEEE Workshop on Policies for Distributed Systems and Networks.* 2008: IEEE Computer Society.
140. Cameron, K., *Introduction to the Laws of Identity*, in *http://www.identityblog.com/?p=354*. 8 January 2006.
141. Gundotra, V., *Introducing the Google+ project: Real-life sharing, rethought for the web*, in *http://googleblog.blogspot.com/2011/06/introducing-google-project-real-life.html [accessed 18/07/2011].* 2011.

BLANK PAGE
IN
ORIGINAL