A Proof-of-Proximity Framework for Device Pairing in Ubiquitous Computing Environments

Yasir Arfat Malkani
y.a.malkani@sussex.ac.uk

October, 2010

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy (DPhil) in the School of Informatics, University of Sussex, Brighton, UK

THESIS COMMITTEE

SUPERVISORS

Dr. Dan Chalmers
School of Informatics
University of Sussex, UK

Dr. Ian Wakeman
School of Informatics
University of Sussex, UK

EXAMINERS

Dr. Kun Yang
School of Computer Science & Electronic Engineering (CSEE)
University of Essex, UK

Prof. Chris Chatwin
School of Engineering and Design
University of Sussex, UK

DEDICATED

TO

MY WHOLE FAMILY

SPECIALLY TO MY MOTHER

AND

LATE BROTHER ZAFAR JAMIL MALKANI

ABSTRACT

Ad hoc interactions between devices over wireless networks in ubiquitous computing environments present a security problem: the generation of shared secrets to initialize secure communication over a medium that is inherently vulnerable to various attacks. However, these ad hoc scenarios also offer the potential for physical security of spaces and the use of protocols in which users must visibly demonstrate their presence and/or involvement to generate an association. As a consequence, recently secure device pairing has had significant attention from a wide community of academic as well as industrial researchers and a plethora of schemes and protocols have been proposed, which use various forms of out-of-band exchange to form an association between two unassociated devices. These protocols and schemes have different strengths and weaknesses – often in hardware requirements, strength against various attacks or usability in particular scenarios. From ordinary user's point of view, the problem then becomes which to choose or which is the best possible scheme in a particular scenario.

We advocate that in a world of modern heterogeneous devices and requirements, there is a need for mechanisms that allow automated selection of the best protocols without requiring the user to have an in-depth knowledge of the minutiae of the underlying technologies. Towards this, the main argument forming the basis of this dissertation is that the integration of a discovery mechanism and several pairing schemes into a single system is more efficient from a usability point of view as well as security point of view in terms of dynamic choice of pairing schemes. In pursuit of this, we have proposed a generic system for secure device pairing by demonstration of physical proximity. Our main contribution is the design and prototype implementation of Proof-of-Proximity framework along with a novel Co-Location protocol. Other contributions include a detailed analysis of existing device pairing schemes, a simple device discovery mechanism, a protocol selection mechanism that is used to find out the best possible scheme to demonstrate the physical proximity of the devices according to the scenario, and a usability study of eight pairing schemes and the proposed system.

ACKNOWLEDGEMENTS

First of all I would like to thank my advisors, Dr. Dan Chalmers and Dr. Ian Wakeman for their guidance and inspiration during this work. Without their support this dissertation would not have happened. Secondly, I am indebted to my whole family for their support throughout my educational career.

I am thankful to Dr. Des Watson for his valuable feedback during my annual review interviews. I am also thankful to all of the members of Foundations of Software Systems Group. Specially, I am indebted to the following people for their insightful comments, helpful discussions, and proof reading of this whole dissertation: Simon Fleming, Danny Matthew, James Stanier, Anirban Basu, Lachhman Das Dhomeja, Renan Krishna, Roya Feizy and Aeshah Alsiyami.

I am also grateful to the vice chancellor and the registrar of the University of Sindh, Jamshoro, Pakistan (i.e. source of PhD studies funding), who awarded me scholarship to pursue my DPhil studies here in University of Sussex for four years.

Finally, I am also very much thankful and indebted to those anonymous people, who might have given me some valuable suggestions/feedback at some point in this research, and my heartfelt thanks to those who voluntarily participated in the usability study.

DECLARATION

I hereby declare that this thesis has not been and will not be, submitted in whole or in part to another University for the award of PhD or any other degree.

Signature

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

This first chapter of the dissertation presents an overview of the research area along with motivation towards the need for the proposed system. In this chapter, we also describe our contributions followed by a list of our publications. At the end of the chapter, we have presented the organization of the remaining parts of the dissertation.

1.1 INTRODUCTION AND MOTIVATION

In ubiquitous computing, computing devices are spread around us, whereby they are interconnected with each other through either wireless or wired connectivity. They do not require continuous attention from the users in order to perform tasks as they are seamlessly integrated into the background. Ubiquitous computing environments are becoming popular and a common-place nowadays. It is due to the continuous advancements in communication technologies and proliferation of modern small hand-held devices. Many modern devices (e.g. smart printers, PDAs, smart phones and cameras) support multiple communication channels and almost all of them use wireless technology in some form, such as Bluetooth, Infrared, Wibree, Zigbee, or 802.11. Having wireless technology in these devices does not guarantee that all of these devices can also take advantage of Internet technology. However, those wireless enabled devices that cannot connect to Internet, can still take advantage of other co-located devices in the vicinity by creating short-term or long-term associations. For example pairing a laptop with a printer or an access point in an airport lounge through the use of WiFi or Bluetooth (i.e. short-term association), and pairing a PDA with home devices in order to control them wirelessly (i.e. long-term association). Some other examples of pairing from everyday life include pairing a Bluetooth enabled headset with a mobile phone or MP3 player, pairing of Bluetooth keyboard with the desktop computer, and pairing of two mobile phones to exchange music files or other data. Since wireless communication is susceptible to eavesdropping, one can easily launch man-in-the-middle (MiTM), denial-of-service (DoS) or bidding-down attacks to break the secure pairing process. MiTM attack is a kind of active eavesdropping, in which an adversary can fully intercept the messages moving in both directions, modify or, corrupt the message, store messages for later replay, or insert new messages; In DoS attack, an adversary prevent communication between two legitimate communicating partners; and in bidding-down attack, the goal of an adversary is to fool (i.e. bid-down) the intended communicating partners to use weaker security than is possible.

Over the last ten years significant research efforts have addressed the issue of secure device pairing. The main goal of the research community working on the secure device pairing issue is to provide mechanisms that give assurance of the identity of the devices participating in the pairing process and to secure them from being victims of eavesdropping attacks, such as MiTM attack. Achieving this goal is a challenging problem from both the security and the usability points of view.

Security challenges emerge due to the ad hoc and dynamic nature of ubiquitous computing environments, in which devices do not know each other a priori, but still need to develop spontaneous interactions between themselves. This precludes the idea of pre-shared secret keys. Further, traditional key exchange or key agreement approaches – such as Diffie-Hellman [1] in its original form – are not applicable in wireless environments due to their vulnerability to a MiTM attack.

From a usability point of view, since most of the device owners are non-technical, they want to have minimal and easy interactions with their devices during the pairing process. They do not want to remember a list of PIN numbers or secret passwords to establish the secure communication channel between a pair of devices for several scenarios or situations. Since many users do not have a deep technical understanding of the risks of pairing and there is a substantial cognitive overhead in remembering the different kinds of steps of secure pairing for several categories of devices and situations, many users may either deactivate security of the devices or select an inappropriate pairing method that may cause poor security. Therefore it is also challenging to develop more general, standardized and user-friendly interaction methods that might increase the usability of pairing schemes. Some other challenges are due to the devices' heterogeneity in terms of their communication channels, user interfaces, power requirements and sensing technology that make it hard to give a common or standard solution for secure pairing of devices.

As a result of these challenges, a wide community of researchers has proposed many protocols and schemes [2-31] to deal with this issue. These protocols vary in their assumptions about the required capabilities in the devices, required human intervention, and in the way they utilize out-of-band or location-limited side channels including physical, audio, visual, short-range wireless channels like Near Field Communications (NFC), and also combinations of these. Consequently, there currently exists many options for an ordinary user to establish a secure channel between the devices from entering pins and passwords to verifying hashes of public keys and pressing buttons simultaneously on the two devices. This notion contradicts with the usability goal of secure device pairing schemes. As a motivating example towards this, consider the following scenario.

Let us introduce Angela, who is working in a reputable organization. She organizes a meeting with representatives of some customers to give them a confidential briefing about a new product that her company is launching in the near future. The meeting is organized in a hotel equipped with modern smart devices, but which is unfamiliar to Angela. On the meeting day, Angela is getting late, so she leaves her office in hurry and forgets to print some important documents required during the meeting. When she reaches the hotel, she wants to pair her laptop with a nearby printer to print the documents, without having to gain special permissions on the hotel network or pass files to a receptionist. That she has been allowed into the room with the printer is sufficient credentials. Next she goes to the meeting room, where she wants to pair her laptop with the projector securely, since the presentation carries some sensitive data. In addition to preventing eavesdroppers on a connection expected to last for several hours, Angela's laptop selects a mechanism that allows her to demonstrate to the room that the data is coming from her laptop. After her meeting and before leaving, she needs to discuss a confidential issue with her boss. At this time, she wants to pair her Bluetooth enabled headset with her mobile phone. Finally, when she finishes everything and needs to leave the hotel, she wants to provide the hotel with a signature stored on her work smart-ID card to use in authenticating their invoice.

The scenario presented above embodies common problems in ubiquitous computing of ad-hoc interactions with unfamiliar devices and institutions, but can also make use of physical presence. It gives rise to two major concerns regarding the pairing process. First is how Angela makes sure that no one else can modify or read the sensitive data sent to the various devices. This requires setting up of keys for encryption, but also correct device selection in an unfamiliar environment. Second, while pairing the devices she needs to discover which pairing process can be applied in each situation. To the best of our knowledge, there is no common secure pairing system that best fits in all four situations of the scenario. For example accelerometer based techniques (e.g. [13, 15, 18]) are not practical for large devices, and in a large room with a roof mounted projector radio signal and close-range techniques are likely to fail (e.g. [14, 28]). Where a choice of pairing techniques is available not all users are capable to judge which one is the best to use. Further, a pairing system must not increase the complexity and the cost of the devices by requiring expensive dedicated hardware in all devices, but should accommodate the existing capabilities of the pairing partners and should be flexible enough to accommodate future technologies.

In view of above facts, we believe that a common pairing infrastructure for ubiquitous computing environments can improve the usability of the pairing process. In this dissertation, we are presenting such a system. The proposed system integrates device discovery, several pairing schemes and a selection mechanism into a single model that facilitates association of any pair of devices in a wide range of scenarios by using the devices' existing capabilities and user preferences, and also assists the user to select an appropriate pairing protocols and relieves him/her from choosing between more than two dozen [2-31] of pairing schemes. The detailed analysis of these schemes is given in chapter 2.

1.2 CONTRIBUTIONS

The major goal of this research work is to investigate and examine the feasibility of a framework based approach to device pairing. The main contribution of this dissertation can be summarized as the design and implementation of the proof-of-proximity (PoP) framework and the Co-Location (CoLoc) protocol that facilitates a generic device pairing system, which can be used in a wide-range of device pairing scenarios in ubiquitous computing environments. In pursuance of main contribution, we have also made several other contributions. We have summarized our overall contributions as below:

- A proof-of-proximity framework and its prototype implementation.
- The design and implementation of a novel Co-Location (CoLoc) protocol.
- Critical and comparative analysis of existing device pairing schemes.
- A simple device discovery mechanism for co-located devices.
- A PoP protocol specification and selection mechanism to demonstrate the physical proximity.
- A usability study of eight pairing schemes and the proposed system.

Note that we built our solution on the advances already made in the field of device pairing. Since there had been a number of schemes already developed for providing secure device pairing in ad hoc networks; we believed in integrating these schemes into the proposed system either in their original form or with some minor variations (subject to satisfying certain minimum requirements concerning their integration) to achieve our objectives. Majority of the researchers exploit the property of physical presence of devices in same space/location in some way to achieve their goal of device pairing. Consequently, in this dissertation, the term "PoP or Proof-of-Proximity Protocols" refer to the set of those device pairing protocols that are implemented in the proposed system either in their original form or with some variations.

1.3 RELEVANT PUBLICATIONS

Following list of publications describe the author's prior work published, which is relevant to this dissertation. However, in this dissertation a more comprehensive description of the ideas and concepts involved and work undertaken is given as compared to the sum of these publications.

1. **Malkani, Y.A.,** D. Chalmers, and I. Wakeman. Towards a General System for Secure Device Pairing by Demonstration of Physical Proximity (Poster). in UBICOMP Grand Challenge: Workshop on Ubiquitous Computing at a Crossroads: Art, Science, Politics and Design. 6th and 7th January, 2009. Huxley Building, Imperial College, London.

2. **Malkani, Y.A.,** D. Chalmers, I. Wakeman, and L. D. Dhomeja. Towards a General System for Secure Device Pairing by Demonstration of Physical Proximity, in MWNS-09 co-located with IFIP Networking 2009 Conference, Shaker Verlag: Aachen, Germany. ISBN: 978-3-8322-8177-9. pg. 13-24.

3. **Malkani, Y.A.** and L.D. Dhomeja, PSIM: A Tool for Analysis of Device Pairing Methods. International Journal of Network Security & Its Applications (IJNSA). ISSN: Print - 0975 - 2307, Online - 0974 - 9330, October 2009. 1(3).

4. **Malkani, Y.A.** and L. Das Dhomeja. Secure device association for ad hoc and ubiquitous computing environments. in IEEE 5th International Conference on Emerging Technologies, ICET 2009. pg. 437-442.

5. **Malkani, Y.A.,** D. Chalmers, and I. Wakeman, Secure Device Association: Trends and Issues, Book Chapter in Security of Self-Organizing Networks: MANET, WSN, WMN, VANET, October 2010. ISBN: 978-1-4398-1919-7, Editor: A.-S.K. Pathan, Auerbach Publications. url:http://www.routledge.com/books/Security-of-Self-Organizing-Networks-isbn9781439819197.

6. **Malkani, Y.A.,** D. Chalmers, and I. Wakeman. A Framework for Secure Device Pairing by Demonstration of Physical Proximity (Under review). in Frontiers of Information Technology (FIT-2010), Proceedings will be published by ACM.

In (1 and 2), we presented our initial ideas and the position of our research to a wider community of researchers in order to get their feedback. Some content presented in chapter 1 and 6 is from (2). In (3), we presented the details of a tool designed to test the usability of pairing schemes and also presented the results of an early usability study that become the basis for the usability study presented in chapter 5. In (4), we have presented the short survey of the state of the art in secure device pairing and (5) is the extended version of the (4). A significant part of the contents presented in chapter 3 is from (5). In (6), we have described the details of the overall framework. In fact this paper covers the work presented in chapter 3 and 4 of this dissertation.

1.4 OTHER PUBLICATIONS

7. Khuhro, Z.-u.-A., A. Harrison, and **Y.A. Malkani**. RNA Structures Comparison using Graphs and Matrices (Short Paper). in IET BioSysBio'09 Conference. 2009. Cambridge, UK.

8. **Malkani, Y.A**. and L.D. Dhomeja. Location aware device discovery for physically constrained environments. in IEEE 2nd International Conference on Computer, Control and Communication (IC4 09) 2009: ISBN: 978-1-4244-3313-1.

9. Elahi, M.A., **Y.A. Malkani**, and M. Fraz. Design and implementation of real time vehicle tracking system. in IEEE 2nd International Conference on Computer, Control and Communication (IC4-09) 2009: ISBN: 978-1-4244-3313-1.

10. Fraz, M., **Y.A. Malkani**, and M.A. Elahi. Design and implementation of real time video streaming and ROI transmission system using RTP on an embedded digital signal processing (DSP) platform. in IEEE 2nd International Conference on Computer, Control and Communication (IC4-09). 2009: ISBN: 978-1-4244-3313-1.

1.5 DISSERTATION OUTLINE

Chapter 2 provides a detailed analysis of the device pairing schemes along with some basic concepts and terminology. The main focus of this chapter is a detailed discussion of the state-of-the-art in device pairing along with an evaluation of existing solutions based on a comparative usability and security analysis. This analysis proved to be helpful in drawing a scope for the problem we are addressing in this dissertation.

The focus of chapter 3 is the details of the proposed PoP framework. It gives the reader an understanding of the architectural view of our system. In this chapter, we have also presented the details of Co-Location (CoLoc) protocol that is one of our main contributions and the core part of the system followed by the PoP protocol selection mechanism and some of the additional features of the proposed system.

In chapter 4, the focus is prototype implementation of the proposed framework. In this chapter, we describe the software components of the proposed system and discuss the structure of the CoLoc protocol messages. Further, in this chapter we also present the classification and description of the integrated PoP protocols followed by the demonstration of the prototype implementation.

In chapter 5, we provide the details of a usability study that is carried out in pursuance of the hypothesis of the dissertation and to evaluate the proposed system. In this chapter, we present the evaluation of the proposed system through the analysis of the usability study results and its own design features followed by describing the wider view of the usability study results in general and their impact on the PoP protocol selection criteria in particular.

Finally, chapter 6 provides the summary of the dissertation, achievements and contributions followed by the future work and closing remarks.

CHAPTER 2

SECURE DEVICE PAIRING: TRENDS AND ISSUES

The main focus of this chapter is the survey and analysis of the protocols and schemes that use various forms of out-of-band exchange to form an association preceded by describing some of the basic concepts and terminology and several possible attacks in device pairing model. At the end of this chapter, we present motivation towards the need of framework-based approach to secure device pairing.

2.1 BASIC CONCEPTS AND TERMINOLOGY

2.1.1 RECAPITULATION OF CRYPTOGRAPHIC PRELIMINARIES

Cryptography is the science of hiding data in order to provide information security at numerous levels and in several disciplines. The main objective is to control people/entities access to the information for which they must show their legitimacy.

To achieve this goal, over a long period many schemes [32-34] have been proposed, which are collectively known as cryptographic primitives. In cryptography, the original data is called the plaintext; the transformed or altered data is called the ciphertext. The process of transforming plaintext into cihpertext and ciphertext into plaintext is known as encryption and decryption. The algorithm which performs the transformation is called a cipher. Conventionally, a cryptosystem may consist of three basic components: keys, algorithms, and key management schemes. Symmetric cryptosystems use the same key for encryption/decryption; while asymmetric cryptosystems use a pair of keys (public/private) for encryption/decryption. In practice, symmetric-key based systems are much more efficient than asymmetric-key based systems; however asymmetric cryptography provides more efficient key management. Recently, another trend (known as hybrid cryptography) has began, in which asymmetric cryptography is combined with symmetric cryptography by transferring a secret key between communicating parties using an asymmetric scheme, and then performing encryption/decryption using symmetric schemes. Hash functions are cryptographic algorithms that take a string of any length as an input parameter and produce a short fixed-length hash code; while MACs are similar to hash functions, except that they require a secret key to authenticate the hash code on the recipient-side. A digital signature is an alternative to MACs, which also provides data integrity and authenticity while the public key of the entity who signed the message is trusted. The difference between digital signatures and MACs is that MACs are generated and verified using the same secret key, while this is not the case for digital signatures. Further, digital signatures also provide non-repudiation. In scenarios where there is some doubt in the ownership of the sender's public key, digital certificates are used. Digital certificates are issued by a trusted third party and are messages that associate an identity to a public key. Interested readers can find further details of cryptographic primitives in [32-34].

2.1.2 COMMUNICATION CHANNEL

In computer and communication systems, a communication channel, or simply channel, refers to a transmission medium that is used to transfer an information signal from one point to another; thus allowing two or more entities to communicate with each other. A communication channel could be physical, such as a wired-channel, or wireless, such as a radio-channel. Since wired-channels use a physical connection between the sender (i.e. transmitter) and the receiver points, these channels are less vulnerable to interference, and are more secure and private. Wireless channels are much more open without any physical connection between sender and receiver points and are more vulnerable to noise and interference as compared to wired-channels. Thus, the major drawback of using wireless channels is the ease of interception.

2.1.3 OUT-OF-BAND CHANNEL

An out-of-band (OOB) channel is also known as location-limited side channel or simply physically constrained channel. In the literature of device pairing, the term communication channel or in-band channel is used for a fast and high bandwidth, but unreliable and insecure channel, such as 802.11 or Bluetooth; while the term physically constrained channel is used for a slow and very low bandwidth secondary communication channel. Such a channel usually has additional security guarantees (e.g. confidentiality or message integrity) that help to create a secure association between a pair of devices. In many cases, the additional security comes through the absence of vulnerability to attacks on the network and/or a requirement that engagement with the channel is physically visible to the users, and possibly being as simple as direct person-to-person verbal exchange. One of the major uses of OOB channel is to transfer messages for authentication during the pairing process. These channels can be categorized into two broad categories: input OOB channels and output OOB channels. The first category is usually used to enter some data into the device(s) during the pairing process, such as entering a PIN code or Passkey using a keypad. The latter category is used for verification purposes through the use of some output capability of the device, such as display.

2.1.4 TRADITIONAL VS. UBIQUITOUS COMPUTING ENVIRONMENTS

Traditional computing environments are usually composed of static devices (such as computers, printers, scanners), which use fiber-optic or copper wires, along with hubs, switches and routers to establish the communication network amongst them and the most widely used communication mechanisms is through the use of traditional TCP/IP model. In these networks, devices rarely change their physical location, so they are not dynamic in nature. Since devices in traditional networks are static, they are connected with some power-outlet, and do not face the problem of battery exhaustion. Further, traditional networks usually work with the support of some infrastructure, such as on-line servers, which provide several useful services for the management and survival of these networks.

Ubiquitous computing environments are usually composed of modern small, hand-held and embedded devices (such as PDAs, mobile phones, MP3 players, wireless gadgets etc), which use short-range wireless technology to establish the communication network amongst them. These networks are built spontaneously without relying on some fixed infrastructure. In contrast to traditional computing environments, ubiquitous computing environments are wireless, ad hoc and dynamic in nature. As a consequence, the security mechanisms and solutions proposed and developed for the traditional computing environments – in their original form – are not applicable in the scenarios of ubiquitous computing environments. Therefore, in order to provide secure communication between devices in ubiquitous computing environments, we need new security mechanisms that others have already started to propose and develop. For example, to ensure security and privacy of ubiquitous computing systems, several approaches, from authentication [35-37] and access control [38, 39] to distributed trust [40-43] have been proposed. Consequently, the issue of secure device pairing has also received significant attention from many researchers and a large set of device pairing schemes and protocols have been proposed [2-4, 7-16, 19-29, 31, 44-47]. Towards this, in this dissertation, we have attempted to provide a light-weight infrastructure for the secure pairing of devices by integrating several pairing schemes along with a discovery mechanism into a generic framework.

2.1.5 SECURE DEVICE PAIRING

Secure device pairing (also known as security initialization, secure first-connect, secure device association or simply device pairing in the literature) is the process of establishing a secure channel between two unassociated human-operated devices over a short range wireless channel, such as Bluetooth, Infrared or 802.11. In the context of this dissertation, short range refers to a close proximity or single space (such as room) in which devices are operating.

2.1.6 SECURE GROUP COMMUNICATION

Apart from addressing the issue of establishing a secure session between two devices, we have also shown in this dissertation how to achieve secure group association through the demonstration of physical proximity. The details of the proposed scheme for secure group association are presented in chapter 3.

A secure group is formed by the members who play roles as client, resource or both. Since secure groups are dynamic in nature and may vary over time, a group key management protocol is required to ensure the secrecy of the group. The main goal of group key management protocols is to timely provide the latest security relevant information to the legitimate group members in order to maintain the high standard of group security. There are two broad categories of group key management protocols, which are: centrally managed group key distribution protocols, and group key agreement protocols. In centrally managed group key distribution protocols a single entity (i.e. key server or group controller) is exploited for controlling the whole group; while group key agreement protocols do not require or concentrate on any central group controller or key server, but all the group members participate in the generation of group key material in a distributed manner [48]. Additionally, Rafaeli and Hutchison in their survey [49] highlighted the third category of group key management protocols, which is protocols based on decentralized architectures. The protocols designed within this category consider the division of a large group into more manageable subgroups, and each subgroup has a subgroup-manager; thus, trying to minimize the problem of concentrating the work in a single place [49].

2.2 ATTACK TYPES IN DEVICE PAIRING MODEL

As stated earlier, device pairing is the process of security initiation, which enables two entities/devices to establish a secure communication link between them in close proximity. However, achieving this goal in ubiquitous computing environments is a challenging task due to the wireless, ad hoc and spontaneous interaction of devices. Since wireless communication is open to everyone, these systems are highly vulnerable to security risks, such as eavesdropping. Consequently, there are also similar kinds of security threats or attacks in device pairing scenarios. In this section, we briefly describe them.

2.2.1 EAVESDROPPING

The most significant risk in device pairing models is that the underlying communication channel is wireless (e.g. Bluetooth, 802.11, etc), which is open to everyone including bona-fide users as well as intruders or adversaries, and thus pairing partners cannot be physically secured the same way as two peers in a point-to-point wired network. In an eavesdropping attack an adversary secretly listens to the conversation between pairing partners. The adversary's main goal is to obtain confidential information, including: public/private keys, location information, contact details, data of commercial value, or even devices' capabilities. To reduce the risk of eavesdropping general solutions include encryption, and physically securing the medium (line of sight transmission, frequency hopping etc.).

2.2.2 MAN-IN-THE-MIDDLE (MITM) ATTACK

Simple eavesdropping is a passive attack, in which an adversary's goal is to steal some confidential information. However, active attacks are more dangerous, in which the main goal of an adversary is to fool the legitimate device(s) to associate with the adversary's device. A "Man-in-the-Middle" (MitM) attack is the most widespread and well known active attack against device pairing protocols. It is a kind of active eavesdropping, in which an adversary can fully intercept the messages moving in both directions, modify or, corrupt the message, store messages for later replay, or insert new messages. To successfully launch this attack an adversary should be able to establish two independent connections with the victims. In the event of a successful attack victims believe that they are communicating with each other and the messages received by them are from the legitimate source; while, it is not the case. In fact all conversation is passed through the adversary, who is able to illegitimately analyze and modify the real data, launch denial-of-service (DoS) attack, and even impersonate one partner to gain control over the victim's device(s) or gain access to data or resources. Figure 2.1 depicts the scenario of a MiTM attack.

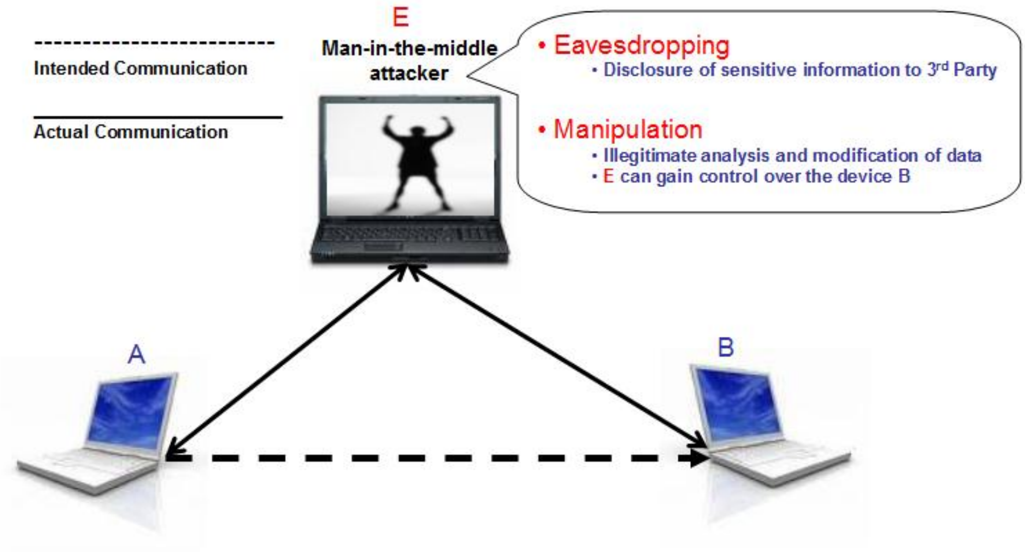

Figure 2.1: Man-in-the-middle attack scenario

2.2.3 DENIAL-OF-SERVICE (DOS) ATTACK

The general goal of an adversary launching a denial-of-service (DoS) attack is to prevent communication between wirelessly connected nodes. However, in the case of device pairing a DoS attack can prevent two legitimate pairing partners from establishing a secure channel. It is a general concept that this is the easiest attack that can be launched by an adversary in wireless environments. Since there has been less emphasis on the prevention of DoS attack in pairing scenarios, many of the pairing schemes are susceptible to a DoS attack. For example, in pairing schemes that use audio as an out-of-band channel, an attacker can launch DoS attack by creating a noisy environment for the user/devices. The noisy environment may prevent the user from properly participating in the pairing process. In the case of visual out-of-band channels, this attack can be launched by manipulating the lights (dark, bright, flashing etc) so that bar codes, screens etc used to contain secure pairing information cannot be read. However, these kinds of DoS attacks can be recognized by the user, who can then try to eliminate them by changing the environment or by forcing the adversary not to do so in case of source detection.

2.2.4 BIDDING-DOWN ATTACK

The bidding-down attack is possible in scenarios where a list of choices to establish a secure channel is available, and the selection of the best pairing protocol is negotiated based on some criteria, such as device capabilities or user preferences. In this kind of attack, the goal of an adversary is to fool (i.e. bid-down) the intended pair-able devices to use weaker security than is possible. For instance, when pairing two display and camera-equipped devices, an adversary could modify the capabilities of one of the devices into display-less and/or camera-less device (i.e. bidding-down) to force a radio-based pairing protocol to be used, which is easier to intercept without being detected.

2.2.5 COMPROMISED DEVICES

Compromised devices are a risk in any wireless system and are difficult to prevent at the protocol level. In the case of secure device pairing, it is possible that an adversary may install malicious code on the device(s). Then an adversary can access confidential information (e.g. shared secret) stored on the device or use it to gain authorized access to other available services. Further, a compromised device could suggest pairing with only the adversary's device or could run a weak pairing protocol. It is the user's responsibility to eliminate the chance of this attack by some mechanism, such as deploying security software to detect the malicious code or to restrict the physical access of the device to only those people whom he/she trusts.

2.3 DEVICE PAIRING IN AD HOC AND UBIQUITOUS COMPUTING ENVIRONMENTS

The problem of secure device pairing continues to be a very active area of research in ad hoc and ubiquitous computing environments. The issue got significant attention from many researchers after Stajano et al. [2-4] highlighted the challenges inherent in secure device pairing. As a result, currently we have more than two dozen device pairing schemes including their variations. In this section, we present a survey of several approaches to device pairing along with a detailed comparative analysis (section 2.4).

2.3.1 DEVICE PAIRING SCHEMES PROPOSED BY ACADEMIA

2.3.1.1 DEVICE PAIRING REQUIRING WIRED OR CONSTRAINED CHANNEL

In their seminal Resurrecting Duckling paper [3] Stajano and Anderson presented a policy-based mother-duckling security model that played an important role in raising the issue of secure device pairing among a wide community of researchers. Their work [2-4] has been considered as the first effort towards secure transient association between devices for ad hoc and ubiquitous computing environments. The proposed mother-duckling model maps the relationships between devices. Mother is a master device that imprints a duckling that is a slave device. The slave device remains in one of the two states: imprinted or imprintable. The slave device is in the imprintable state at the beginning or bootstrapping time. However, it switches from imprintable to the imprinted (paired) state once it has got the shared secret from its master device. The slave remains in this state until its death (i.e. while it keeps the shared secret provided by its master device). In fact the shared secret binds the slave device to its master device. As a consequence, the slave device remains faithful to the master device and obeys no one else. Since the shared secret is transferred from master to slave over a physical connection (such as using a cable) in plain-text form the proposed approach does not require complex cryptographic methods, such as Diffie-Hellman [1].

Balfanz et al. [21] extended Stajano and Anderson's work and proposed a two-phase authentication method for pairing of co-located devices using infrared as a location limited side channel (also known as out-of-band channel). In their proposed solution pre-authentication information is exchanged over the infrared channel and then the user switches to the common wireless channel. Pre-authentication data contains cryptographic material as well as the complete address of the device. The proposed method exploited public key cryptography in which devices exchange their public keys over an insecure wireless channel followed by exchanging the hashes of respective public keys over the location limited side channel (i.e. infrared). Further, they are the first to introduce the concept of demonstrative identification (i.e. identification in the form of a representation of an object, e.g. printer in this room, display in front of me, etc) for authentication purposes in pairing process. Slightly different variations, of Balfanz et al.'s [21] approach are proposed in [24-26, 50], which use infrared, laser and ultrasound as location limited side channels to transfer the pre-authentication data.

2.3.1.2 DEVICE PAIRING USING SENSORS TECHNOLOGY

Unlike the approaches described above, the idea of shaking devices together to pair them has become more common. Smart-its-Friends [13] was the first effort that proposed pairing of two devices using a common movement pattern and used accelerometers as an out-of-band channel. In this approach, two devices are held and shaken together simultaneously. Then common readings from the embedded accelerometers in the devices are utilized to establish the communication channel between the two devices. However, security has not been the major concern of Smart-its-Friends. The follow-on methods to Smart-its-Friends are Are You With Me [45] and Shake Well Before Use [15]. In Are You With Me [45], the main goal was to show that accelerometer's data can be used to reliably determine that a set of devices are being carried by the same person. The authors showed that one can reliably determine whether the two devices are being carried by the same person or not using only eight seconds of walking data. However, one of the major limitations of the proposed system is that they require the user(s) to walk [45].

Mayrhofer and Gellersen [15] extended Holmquist et al.'s [13] approach and proposed two protocols to securely pair the devices. Both of the proposed protocols exploit cryptographic primitives with accelerometer data analysis for secure device-to-device authentication. The first protocol use public key cryptography and is more secure as compared to the second protocol, which is more efficient and computes a secret key directly from the accelerometer's data. In second scheme, the user is required to hold and shake the devices together for approximately twenty seconds to generate a 128-bit shared secret [15]. Kirovski et al. proposed Martini Synch [51], another accelerometer based approach to securely pair the devices that use the idea of joint fuzzy hashing [44].

Another approach that requires shaking or moving patterns is Shake Them Up [14]. Authors suggest a movement-based technique for pairing two resource-constrained devices that involves shaking and twirling them in very close proximity to each other. Unlike accelerometer-based schemes, this approach exploits the source indistinguishability property of radio signals and does not require embedded accelerometers. While being shaken, two devices exchange radio packets and agree on a key one bit at a time, relying on the adversary's inability to determine the source of each radio packet (i.e. the sending device).

Recently, Varshavsky et al. [28] proposed Amigo a proximity-based technique for secure pairing of co-located devices. They extended the Diffie-Hellman key exchange protocol with the addition of a key verification stage. The proposed approach utilizes commonality of radio signals from locally available wireless access points to establish the secure channel between the devices. Any attacker who is not physically very close would see a different pattern of access point signal strengths. Radio-based approaches to secure device pairing either require no or minimal hardware and user involvement during the pairing process. However these schemes are not applicable in the scenarios where devices support only Bluetooth technology.

Biometrics are a common technique for identifying human beings. Due to the success of biometric-based user authentication systems, researchers realized that many benefits could be achieved by combining biometrics with cryptography. As a consequence, Buhan et al. proposed two systems [20, 23] that utilize biometric data to establish a secure channel between the devices. Both of the proposed systems are based on the Balfanz et al. model [21], and biometrics is used as an out-of-band channel. In Feeling-is-Believing (FiB) [23], Buhan et al. investigated grip pattern and proposed to generate a shared secret key from biometric data using quantization and cryptanalysis. In SAfE [20], keys are extracted from images during the pre-authentication phase that are used for authentication in subsequent phase.

2.3.1.3 DEVICE PAIRING USING NEAR FIELD COMMUNICATION TECHNOLOGY

Near-Field Communication (NFC) is a short-range, high-frequency, low-bandwidth wireless connectivity technology defined by the NFC Forum [36]. Since NFC uses magnetic field induction to enable communication between devices it allows users to securely pair the NFC-enabled devices by simply touching them together or holding them in very close proximity of up to 10 centimeters. NFC enabled devices are capable of establishing a peer-to-peer network to exchange content and access services. It operates on the 13.56 MHz frequency with data transfer rate of up to 424 kilobits per second, with a bandwidth of 14 KHz. However, NFC in combination with other wireless technologies, such as Bluetooth or WiFi, can be used for exchanging a huge amount of data or can support longer communication.

In NFC, there are two kinds of devices - active-devices that generate their own field, and passive-devices that retrieve power from the field generated by active-devices. NFC supports two basic modes of communication: active-mode and passive-mode. In active-mode, both of the devices generate their own magnetic field and require a power supply in each of them. While in passive-mode one of the devices (an active-device) generates its magnetic field and the other device (i.e. passive-device, such as a contactless smart card) is powered by the active-device. There are many scenarios where NFC can be used. One such common scenario is the pairing of a NFC enabled camera and computer. In that scenario user could transfer all the photos in camera into his/her computer just touching them together or putting them in very close proximity. The touch mechanism makes it clear for the user which two devices are selected for intended association and takes away the burden of selecting the right devices (i.e. discovery and device identification) from a long list of available devices.

Other possible applications/uses of NFC include smart posters, replacement of contactless-credit-cards with NFC-enabled mobile phones, and support services (through the use of voice clips) for the visually impaired people. Wi-Fi protected setup also incorporates one of the methods that use NFC as an out-of-band channel. Recently there has been much greater availability of this technology in commercial devices including Nokia 6131, Motorola L7, SAGEM my700X Contactless, LG600V and Samsung D500E.

2.3.1.4 DEVICE PAIRING REQUIRING AUDIO/VIDEO TECHNOLOGY

Based on the pairing protocol of Balfanz et al. [21], some other schemes are proposed through the use of audio and visual out-of-band channels. One such system is Seeing-is-Believing (SiB) [29]. SiB takes advantage of the common presence of cameras in modern handheld devices, and utilizes two dimensional bar codes for exchanging pre-authentication data (i.e. public keys) between the devices. In the proposed approach, device A encodes cryptographic material into a two-dimensional barcode and displays it on the screen, then device B reads it through a camera to setup an authenticated channel. In the simplest case SiB requires the first device (A) to have a display to show the 2D barcodes and the second device (B) a camera. Then the user is required to focus and place the camera of device B at the first device's (device A) screen properly to take a photograph of the displayed bar code. SiB supports several use cases based on the device capabilities. For example, when the first device has a camera and the other device has only a display, then only the first device (camera-equipped) can authenticate the other device – i.e. display only device (1-way authentication). In the second use case, when both devices are camera and display equipped, then both of the devices can authenticate each other by two protocol runs, one in each direction (2-way authentication). In another use case, when only one device has a camera and the other device has neither a camera nor a display, user can then print a two dimensional barcode on a sticker, containing the cryptographic material, and attach the sticker to the other (camera-less and display-less device) device. In this case, the user takes a photo of the sticker and performs the SiB protocol as usual.

Another pairing method that uses a visual out-of-band channel is proposed by Sexana et al. [7]. To reduce the camera requirement in one of the pairing devices in SiB, they extended the work of McCune et al. [29] and proposed an improvement to it through the use of simple light source, such as an LED, and short authenticated integrity checksums. In fact, they showed that mutual authentication can be achieved with a one-way visual channel, while SiB requires two visual channels, one in each direction (for full functionality). In the proposed scheme [7], device A needs to be equipped with a camera and device B with a single LED. Device A takes a video clip of a blinking pattern on device B's LED. Then the video clip is parsed to extract an authentication string.

Loud and Clear (L&C) [31] and Human-Assisted Pure Audio Device Pairing (HAPADEP) [19] use audio as an out-of-band channel to establish a secure channel between the devices. The main idea of the L&C [31] scheme is to encode the hash of the first device's public key into a MadLib sentence (i.e. grammatically correct but nonsensical sentence) and transmit it over a device-to-human channel using a speaker or a display. The second device also encodes the hash of the received public key from the first device into the MadLib sentence and transmits it over a device-to-human channel using a speaker or a display. The user is then responsible for comparing the two sentences and accepting or rejecting the pairing. There are four variants of this approach: speaker-to-speaker, speaker-to-display, display-to-speaker, and display-to-display. In the first variant, the user is required to compare and verify the two sentences vocalized by the pairing candidate devices. In the second variant, the user is required to compare the vocalized MadLib sentence with the sentence displayed on the other device. In the third variant, user is required to compare the displayed MadLib sentence on one device with the vocalized MadLib sentence from the other device. In the fourth variant, user is required to compare the MadLib sentences displayed on both of the devices. In all of the variants, the user is responsible for accepting or rejecting the pairing based on the results of comparison.

Soriente et al. proposed HAPADEP [19], which is a follow-on from L&C [31]. Soriente et al. consider the problem of pairing two devices that have no common wireless communication channel, such as Bluetooth or WiFi, at the time of pairing. The proposed scheme uses only audio to exchange both public keys and hashes of public keys. The proposed system consists of two phases: key transfer and key-verification. In the key-transfer phase, first device (Device A) encodes cryptographic material along with protocol messages into a fast audio codec and plays the resulting audio sequence. The other device (Device B) records and decodes this audio sequence in order to obtain the key. This process is also repeated in the reverse direction so that Device A could get the key from Device B. In the second phase, each device computes a hash of the received public key and encodes it into a pleasant audio sequence, such as a melody. Then user is required to listen and compare the audio sequences played by both of the devices and accept or reject the pairing based on the results of comparison. This scheme is only applicable to those scenarios where both devices have a microphone and a speaker.

2.3.1.5 DEVICE PAIRING REQUIRING HUMAN-TO-DEVICE OR DEVICE-TO-HUMAN INTERACTIONS

Soriente et al proposed Button-Enabled Device Association (BEDA). The main idea of the proposed approach is to transfer the short secret key from one device to the other using 'button-presses' and then use that key to authenticate the public keys of the devices. A short secret key (21-bits) is agreed upon between the two devices via one of its four variants. These variants are called button-to-button (B-to-B), display-to-button (D-to-B), short vibration-to-button (SV-to-B) and long vibration-to-button (LV-to-B). In fact, the only difference between these variants is the way first device (device A) transfers the bits of the generated short secret to the other device (device B). Bits of a short secret are encoded by the devices using the time-interval between two events, such as a button-press-event. For example, the first and basic variant (i.e. B-to-B) involves the user simultaneously pressing buttons on both of the devices within certain random time-intervals and each of these intervals are used to derive 3-bits of the short secret key. In the D-to-B variant an event is a square that blinks on device-A's display, in the SV-to-B variant an event is a short vibration, while in LV-to-B an event is either the start or the end of a long vibration. For every event-notification that the user receives from device A, he has to press or release a single button at the same time on device B. It enables device B to calculate the same bits of shared secret that are transmitted from device A.

2.3.2 INDUSTRY EFFORTS FOR PROVIDING SECURE DEVICE PAIRING MECHANISMS

2.3.2.1 BLUETOOTH

Bluetooth [52] is a short range wireless technology that allows modern devices, such as mobile phones, PDAs, Cameras and other handheld devices, to communicate with each other over a distance of up to 100 meters. It works in the 2.4 GHz ISM band, and is considered to be one of the simplest ways to wirelessly exchange information between two devices in close proximity. In order to establish a secure communication link between intended pairing devices, the user needs to go through the Bluetooth pairing set up procedure. In Bluetooth pairing, devices need to exchange a short passkey or PIN code to prove that the owners of both devices are agreed to pair the devices with each other. Below are the general steps involved in the Bluetooth pairing process:

1. The pairing process starts when the first device (device-A), such as Bluetooth-enabled mobile phone or PDA, searches for other Bluetooth-enabled devices in the vicinity. The list of found Bluetooth devices would be shown on the screen of device-A. Note that only those devices can be found that are already in Bluetooth discoverable mode and their visibility option is turned ON.
2. Device-A selects the device-B (such as other mobile phone or PDA) from the available list of devices. Then, device-A asks the user to enter a PIN code or passkey. It could be any special code of the user’s choice; however it must be remembered, because it needs to be entered on the other device (device-B). Note that in some of the resources/interface constrained device scenarios, it is not possible to enter the Passkey or PIN code. In that case, there is a fixed code, such as 0000, which the user is required to enter onto the other device.
3. Once the user has entered the passkey on device-A, it sends it to the device-B.
4. If device-B is not a resource constrained devices, it asks the user to enter the same PIN code or passkey; otherwise it simply uses its own standard/fixed passkey (e.g. 0000).

5. Finally, device-B sends back the user-entered passkey to device-A. If device-B's passkey is the same as entered by device-A, then automatically a trusted association takes place between the devices.

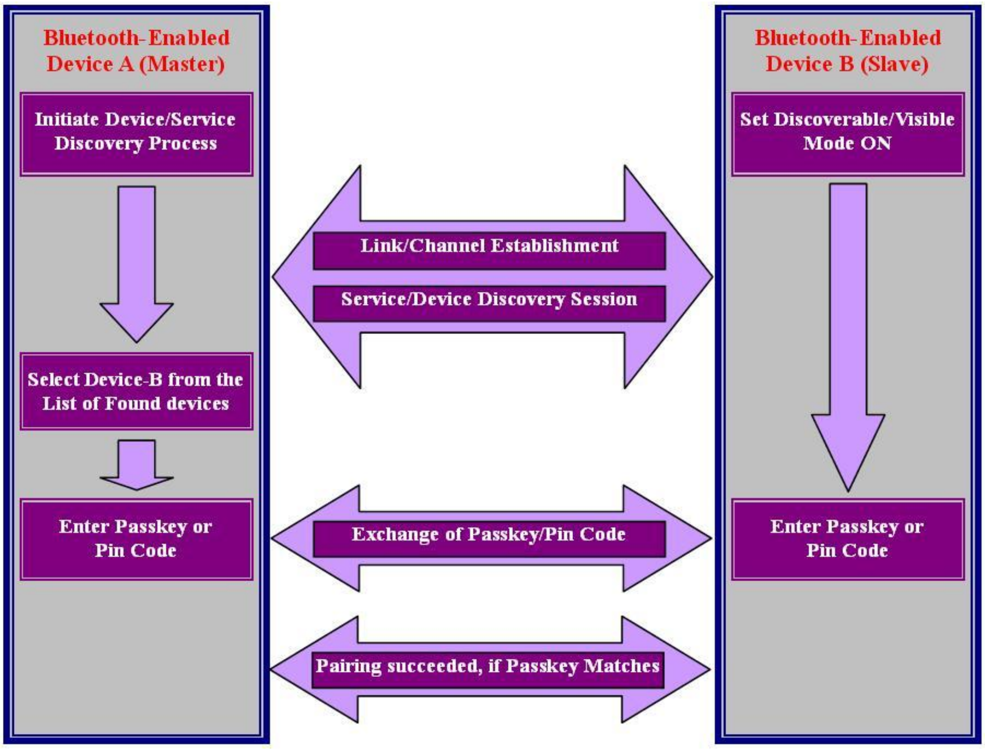

Figure 2.2: Bluetooth pairing process

2.3.2.2 Wireless USB Association (WUSB)

The Wireless USB (WUSB) group was formed in 2004 to define the WUSB specifications that took about one year to complete. WUSB is a short-range (up to 10 meters), high-bandwidth (110 Mbit/s) wireless radio communication technology, which is developed to simplify the process of establishing associations between a pair of wireless-enabled devices. The main goal of this technology is to replace wired USB. In WUSB, device-A (i.e. host-device) and device-B exchanges connection-host-ID, connection-device-ID, and connection-key during the association process. This information is utilized later on to setup secure communication between device-A and the device-B. WUSB supports two types of association models: cable-association model and numeric-association model. Device-A or host-device supports both of the models; while the other device having only USB ports supports the cable association model, and the device with only a display supports the numeric association model. The cable association model utilizes a USB cable to perform the first-time association between a host and a device. Once the association has been completed, the cable is no longer needed and future communications with the device can be entirely wireless. In numeric association model, the first-time association is performed over the ultra-wide-band radio.

2.3.2.3 Wi-Fi Protected Setup (WPS) and Windows Connect now-Net

The Wi-Fi Alliance officially launched Wi-Fi Protected Setups (WPS) in early 2007. The goal was to provide a standard and simple way for easy and secure establishment and configuration of wireless home networks. Another effort for standardization of secure device association is Microsoft's Windows Connect now-NET technology. It provides a way to set up secure wireless network, and works for both in-band wireless devices and out-of-band Ethernet devices.

2.4 Comparative Analysis of Device Pairing Schemes

As described above the issue of secure device pairing got significant attention from many researchers, after Stajano and Anderson in their seminal paper [3] highlighted the challenges inherent in secure device pairing. Since the secret key is transferred in plain-text in their proposed approach, it is susceptible to dictionary attacks. It also requires the same physical interface in both of the devices to transfer the secret, which makes such an approach inapplicable in scenarios where the devices do not have a common physical interface. Further, it is also difficult to carry the cables all of the time. However, Resurrecting Duckling and Talking to Strangers [21] both require minimal user interaction, which is an advantage from usability point of view.

The common drawback of Talking to Strangers [21] and some other similar approaches [24-26] (in terms of use of secondary-location-limited-side channel) is that they need some kind of physical interface (e.g. IrDA, laser, ultrasound, etc) for the pre-authentication phase and are vulnerable to a passive eavesdropping attack in the location limited side channels, e.g. two remotes and one projector. Further, some of the location limited side channels, such as infrared and laser, are highly vulnerable to denial of service (DoS) attacks. Those schemes which use audio and/or visual out of band channels [7, 29, 31] for secure device pairing also suffers from a few problems. For example, SiB [29] requires that one of the peers must be equipped with camera; while in L&C [31] a speaker and/or display is required. Camera equipped devices are usually prohibited in high security areas; while the latter is not suitable for hearing-impaired users. Further, bar code scanning requires sufficient proximity and light in SiB; while L&C and HAPADEP [19] places some burden on the user for comparison of MadLib sentences and audible sequences respectively. An adversary can easily subvert bar code stickers on devices in SiB to launch the successful attack, while ambient noise makes authentication either weak or difficult in L&C as well as in HAPADEP. For example in SiB, a user wants to pair his/her handheld device with a display-less printer to print a confidential document. Since the printer is display-less, a bar code sticker is attached to it. It is possible that an adversary subverts the bar code or swaps it with another printer available in the next building. In that scenario, once the pairing is established, and user sends the document to the printer, it is printed by the adversary's printer in next building. However, this scheme is more secure in the scenarios where both of the devices are camera-equipped and also have displays. Since [7] is a variation of SiB, so this scheme has some of the same limitations as SiB, such as requiring close proximity and a camera in at least one of the devices.

Further, in the case of L&C and HAPADEP more research and development is required in the areas of speech engines, audio codec technology as well as in L&C Dictionary. Moreover, L&C and HAPADEP also suffer from the fact that users cannot be forced to carefully listen to the audio played by the devices. It means a user who does not understand the importance of security might not pay proper attention to the sound played by the devices, and thus can easily ignore the verification stage, and may confirm a false match. Secure pairing of devices by shaking them together is an interesting approach. However, these schemes require embedded accelerometers in both of the devices. Further, shaking devices together is always not possible, since there is large variety of devices, such as printers, projectors and laptops that cannot be held and shaken together simultaneously.

In contrast to the above approaches, AMIGO [28] and Shake Them Up [14] exploit radio signals to establish the secure channel. Since AMIGO uses the similarity of radio signals from the nearby access points, it is not applicable in scenarios where the radio data is not available to process, or where the wireless network is easy to eavesdrop on while remaining physically hidden to the bona-fide users. Further, it is hard to identify the intended device in AMIGO when many other devices surround it, because in the proposed scheme calculated physical proximity is of coarse granular. Moreover, it is also a fact that in many developing countries 802.11-based wireless technology is less popular compared to Bluetooth technology that is common due to the widespread use of mobile phones. Shake Them Up is susceptible to attack by an eavesdropper that exploits the differences in the baseband frequencies of the two radio sources. Biometric based solutions to device pairing are considered to be good from the usability point of view in which biometrics is used as an out-of-band channel. The reason is that biometric-based channels put little cognitive load on the users. However, the calculations required to accurately recognize the biometric-patterns are a heavy burden on its systems. Since no two biometric measurements, even coming from the same user and using the same measurement setup are identical; the issues regarding the accuracy of recognition techniques still need more research and improvement. Another drawback of this approach is that it requires biometric readers in both of the devices.

Bluetooth pairing requires the human operator to put the communicating partners into discovery mode. After discovery and selection of a device, the channel is secured by entering the same PIN or password into both devices that gives rise to a number of usability and security issues [17, 53]. For example, a short password or PIN number makes it vulnerable to dictionary or exhaustive search attacks. In [17], it was shown that an adversary can easily derive a 4 digit PIN from an eavesdropped communication during pairing process in less than 0.06 seconds on a common computer by mounting brute force attack. Further, in Bluetooth pairing an adversary can eavesdrop to break the security from a long distance using powerful antennas. As a consequence, the Bluetooth Special Interest Group (SIG) reacted to these concerns by developing Secure Simple Pairing (SSP) [54]. The SSP supports four association modes: passkey entry, numeric comparison, just works, and an out-of-band model.

Passkey entry mode is designed for two kinds of scenarios: first, where one of the devices has a display and the other an input capability (such as numeric keypad). Second, where both of the devices are capable of entering numeric input through a simple numeric keypad. In former case, a 6-digit number is shown on the display of the first device, which is then entered into the second device by the user. In the latter case, the users of the intended pairing devices are required to enter the same 6-digit number in each device.

Numeric comparison mode is designed for the scenarios where both of the devices have displays, which are capable of showing a 6-digit number and allowing the user to enter a binary input (i.e. 'yes' or 'no') during the pairing process. In the pairing process, user is shown 6-digit number on displays of both of the devices, then user is responsible for comparing the numbers and accepting or rejecting the pairing based on his/her observation.

Just work mode is designed for the scenarios where either one or both of the device has neither an output (display) nor an input (keypad) capability for entering or displaying the numbers. This scheme does not require any user interactions apart from asking/prompting the user to accept a connection. This scheme is suitable for resource constrained devices, such as headsets; however it does not provide any protection against MiTM attack.

Out-of-band mode uses a secondary source of communication (such as NFC) to exchange the security relevant information required during the pairing process. This is designed for the situations where pairing devices use wireless technology other than Bluetooth for the purpose of device discovery and exchange of cryptographic material. Since in this mode the security relevant information required for pairing is exchanged through out-of-band channel, the level of protection against eavesdropping and MiTM attacks is dependent on the out-of-band channel and the mechanisms used for exchanging that information. SSP addresses the two main concerns of the users community using Bluetooth technology, which are: simplicity of the pairing process and security, however recently some of the security vulnerabilities are found in SSP [55, 56]. So far as NFC is concerned, it is an extremely short-range technology compared to other short-range technologies, such as Infrared and Bluetooth. Therefore in many scenarios NFC is used in combination with Bluetooth, where NFC is used for authenticating (pairing) a Bluetooth session used for the transfer of data. NFC setup time is much shorter than Bluetooth. NFC requires less than 0.2ms to set up the connection; while Bluetooth requires approximately 6 seconds. In [57], authors described different possible type of attacks on NFC. For example, NFC offers no protection against eavesdropping and is also vulnerable to data corruption and data modifications [57]. However, it is practically impossible to launch MiTM attack in NFC, especially when Active-Passive communication mode is used [57]. The WUSB project is perceived to have failed at the end of 2008 after the withdrawal of Intel. Two major reasons that play a role in its failure are the need of a power supply cable for the WUSB devices and the consumption of a large amount of energy.

Some other efforts toward providing secure device pairing include Lokey [22], manual authentication [27], and some of the older approaches [58-61] that involves image comparisons. LoKey uses SMS messages to authenticate key exchanged over the Internet, which incurs substantial monetary cost and delay. Gehrmann et al [27] proposed several manual schemes that enable handheld devices to authenticate their public keys by some kind of user interaction. In the proposed schemes, the user manually exchanges short message authentication codes between the devices. These short message authentication codes are strings of very short length consists between 16 to 20 bits. For example, in one of the proposed method user is required to compare the short strings displayed on the screens of intended pair able devices. While, in another case in which one of the device is display-less, user is required to type the short string displayed on first device onto the other device (i.e. display-less device). The early approaches [58-61] encode cryptographic material, such as hash codes, into images and ask the user to compare them on both of the devices. These approaches exempted the user from erring and burdensome process of byte-by-byte comparing of cryptographic hashes, however they require high-resolution displays, which restrict these approaches to only specific types of devices, such as desktop computers, laptops, PDAs and other high-end devices. Finally in table 2.1, we have summarized the features of some of the device pairing schemes, which are described in this chapter and also well known in the literature of device pairing.

Pairing Scheme	Minimum hardware or equipment required in each of the device		Human/User effort required	Out-of-band/Location-limited secondary channel
	Device A	Device B		
Resurrecting Duckling Security Model	A cable and the same physical interface (e.g. USB port) on both of the devices		Set up cable connection between the devices	Cable
Talking to Strangers	Infrared (IrDA) port on both of the devices		Set up infrared (IrDA) connection between the devices	Infrared (IrDA)
Smart-its-Friends	2D accelerometers on both of the devices		Move/shake devices together simultaneously until response signal received	Accelerometer/Motion/ Tactile
Are You with Me?	2D accelerometers on both of the devices		Walk around to shake the devices (sensors) for certain time period	Accelerometer/Motion
Shake Well Before Use	2D accelerometers on both of the devices		Move/shake devices together simultaneously until response signal received	Accelerometer/Motion/ Tactile
Seeing-is-Believing	Display	Camera	Properly place camera of device B at the displayed bar code on device A with sufficient proximity and take the photograph	Visual
L&C (Display-Speaker)	Display	Speaker	Compare the MadLib sentence displayed on the screen of device A with the vocalized MadLib sentence from device B	Combination of audio and visual
L&C (Speaker-Speaker)	Speaker	Speaker	Compare the two vocalized MadLib sentences from both of the devices	Audio
HAPADEP	Speaker	Microphone	Compare two audible sequences/melodies	Audio
Shake Them Up	802.11 network card/interface	802.11 network card/interface	Shake/twirl/move devices around until pairing is done or response signal received	Combination of 802.11 and motion
AMIGO	802.11 network card/interface	802.11 network card/interface	Shake/wave hand near the device until pairing is done or response signal received	Combination of 802.11 and tactile
BEDA (Button-to-Button)	A single button on both of the devices		Press button on both of the devices simultaneously with random time-intervals until response signal received	Tactile
BEDA (Display-to-Button)	Display	A single button	Press and release button on device B whenever display of device A flashes	Tactile
BEDA (Short Vibrations-to-Button)	Vibration capability	A single button	Press and release button on device B whenever device A vibrates	Tactile
BEDA (Long Vibrations-to-Button)	Vibration capability	A single button	Press and hold the button on device B while the device A vibrates	Tactile

Table 2.1: Features summary of the well known device pairing schemes

2.5 THE NEED FOR A FRAMEWORK-BASED APPROACH TO SECURE DEVICE PAIRING

Each of the proposed schemes we have surveyed has strengths and weaknesses – often in hardware requirements, strength against various attacks or usability in particular scenarios. Therefore, we can conclude that no one has yet devised a pairing protocol, which is generic enough to accommodate a very large set of device pairing scenarios and can be considered as a standard solution for ubiquitous computing environments. Currently available schemes for secure device pairing vary in the strength of their security, the level of required user intervention, their susceptibility to environmental conditions and in the required physical capabilities of the devices as well as the required proximity between the devices. Some of these techniques consider devices equipped with infrared, laser or ultrasound transceivers, whilst others require embedded accelerometers, cameras and/or LEDs, display, microphone and/or speakers. Some techniques exploit the knowledge of radio environment to securely pair the devices; others require the user's careful attention and significant manual intervention in pairing process. Further, most of the prior work on secure device pairing considered demonstrative approach (i.e. requires user involvement and/or manual efforts to identify the intended partner) to identification and discovery of the intended pair able co-located device. For example in SiB [29] and the Resurrecting Duckling Security Model [3], the discovery of the intended pair able device is performed manually; while in Talking to Strangers [21] communicating partners exchange their connectivity information over the secondary channel (i.e. infrared). However, in many situations automatic device discovery is required [7]. If we continue to multiply the number of manual or out-of-band discovery mechanisms, users will become confused about the selection of device discovery method during pairing process. For instance, a user wanting to create an association of a mobile phone having a microphone, speaker, camera, display and infrared with another mobile phone having microphone, speaker, display, no camera and no infrared might be confused about the varied types of manual or out-of-band possibilities for device discovery [7]. We therefore agree with the view proposed by Saxena et al. [7] that it should not be the user's responsibility to figure out how and which method to use for device discovery each time; instead an automatic device discovery should take place.

It is therefore appropriate to investigate ways of integrating different pairing protocols and discovery mechanism within a general architecture for providing secure and usable pairing mechanisms for a large set of ad hoc scenarios in ubiquitous computing environments. Such an architecture should facilitate choice of the best pairing scheme, considering device capabilities, environmental limitations, user preferences and the balance between security and usability. We realized this need and proposed a framework-based approach to deal with this issue. In next chapter, we present the details of the proposed framework.

CHAPTER 3

THE PROOF-OF-PROXIMITY FRAMEWORK

In this chapter, we present the design goals, requirements and assumptions along with details of the system architecture of the proposed framework. We also present the details of the Co-location (CoLoc) protocol, which is core part of the proposed framework. Further, we describe the protocol selection mechanism that enables the devices to agree on a common PoP protocol. At the end of this chapter we also describe some of the additional features of the proposed system.

3.1 DESIGN GOALS

The major goal of this research is to design a system that facilitates association of any two co-located devices by demonstration of physical proximity through the integration of discovery mechanism and PoP schemes. Note that PoP schemes are either derived/extended from existing pairing protocols or taken in their original form to provide the authenticity of the physical proximity of devices. These pairing schemes exploit various forms of OOB channels. Note that in the literature of device pairing OOB channel refers to a secondary channel, that work along with the primary in-band channel, such as Wifi or Bluetooth, with additional security guarantees. Due to these features, OOB channels are helpful in developing secure device pairing protocols/schemes.

The three main goals of the proposed system are described below.

- **Generality:** Generality is one of the main goals of the proposed system. The system should be applicable in a wide range of device pairing scenarios in ubiquitous computing environments, capable of incorporating existing pairing schemes and can be extended without major modifications in the design.

- **Usability:** From a usability point of view, the system should be simple to understand, and easy to use for an ordinary user.

- **Security:** Our security goal is twofold. Firstly, the system should be capable of establishing the secure session between two previously unassociated devices through proving the physical proximity of the devices involved in the pairing process. Secondly, all the communication between the entities of the system must be secured.

3.2 DESIGN REQUIREMENTS

To achieve the above mentioned goals, we have identified some of the major requirements described below:

(A) A mechanism is required that facilitates the discovery of possibly co-located devices in the vicinity.

To meet this requirement, we have designed a simple registration and discovery mechanism, which is presented in section 3.4.1.4.

(B) A set of protocols or schemes is required that demonstrates the physical proximity of the two devices.

We have already presented a detailed survey of the state-of-the-art in device pairing in the previous chapter, in which a detailed list of pairing protocols is presented. These pairing protocols and/or their variations could be used to demonstrate the physical proximity of devices. Hereafter in this thesis, the term 'PoP protocols' refers to the set of pairing protocols that are implemented in the proposed system either in their original form or with some variations to facilitate the demonstration of physical proximity through the use of out-of-band channels. This set is chosen to demonstrate the concept, not to limit the use of others.

(C) A generic protocol is required that integrates the discovery mechanism and a set of PoP protocols and exchanges all the other required information/messages between several entities of the system in an encrypted form.

We have designed the Co-Location (CoLoc) protocol to meet this requirement, which is presented in section 3.5.

(D) Users should have some control on the selection of PoP protocols, and the level of required user interaction.

The proposed system is capable of getting user's preferences and considers them during the PoP protocols selection phase. We have presented the details of the selection mechanism of PoP protocols in section 3.6.

(E) The ability to modify PoP protocol selection behaviour at run-time.

The protocol selection mechanism uses an XML-based policy as PoP protocols selection criteria, which is defined in terms of required device capabilities and constraints over PoP protocols. Since, the criterion for the selection of PoP protocols is described in an XML-based protocol specification and selection policy file; it can be changed / modified at run-time.

3.3 DESIGN ASSUMPTIONS

We are considering ubiquitous computing environments, in which devices communicate with each other through short-range wireless technology, such as 802.11 or Bluetooth. They discover each other using our proposed registration and discovery mechanism. We are not considering extremely resource constrained devices, such as sensor nodes. Instead, we are considering those ubiquitous computing devices, which have reasonable battery power and computational capabilities, e.g. mobile phones, cameras, PDAs, laptops, printers etc. These devices are capable of symmetric encryption/decryption, public key based encryption, hashing, signature verification, and have unique device-id or address. Further, devices know their location through some location system already installed in the environment or through their own hardware/software, such as GPS (Global Positioning System). The location information is useful in the discovery process. We assume that the co-location server is a trusted, uncompromised and tamper resistant (or at least tamper evident) device. It is also capable of performing symmetric and asymmetric cryptographic operations. Since, the co-location server is very light-weight; it might be run with other local services (e.g. DNS, print) or any other server, which is part of some existing security infrastructure to limit the deployment costs. Alternatively, it could also be installed into a dedicated low-cost small device. Then each device needs to perform one time demonstrative discovery of the server device in order to build trust. We are considering all the devices registered with the same co-location server as potentially co-located and each co-location server is responsible for handling a particular domain or location. We believe that due to the modern low-cost small ubiquitous computing devices that have now reasonable battery and computational power, one co-location server per scope is feasible.

3.4 SYSTEM DESIGN

In figure 3.1, we have shown the high-level architecture of the proposed system, which illustrates three phases. The first two phases are registration and discovery of the device(s), and the third phase is selection, initiation and execution of the PoP protocol. Figure 3.2 shows the two major components of the system, which are the co-location server and the device. These two components are composed of several other software components, which are described in the implementation chapter (chapter 4).

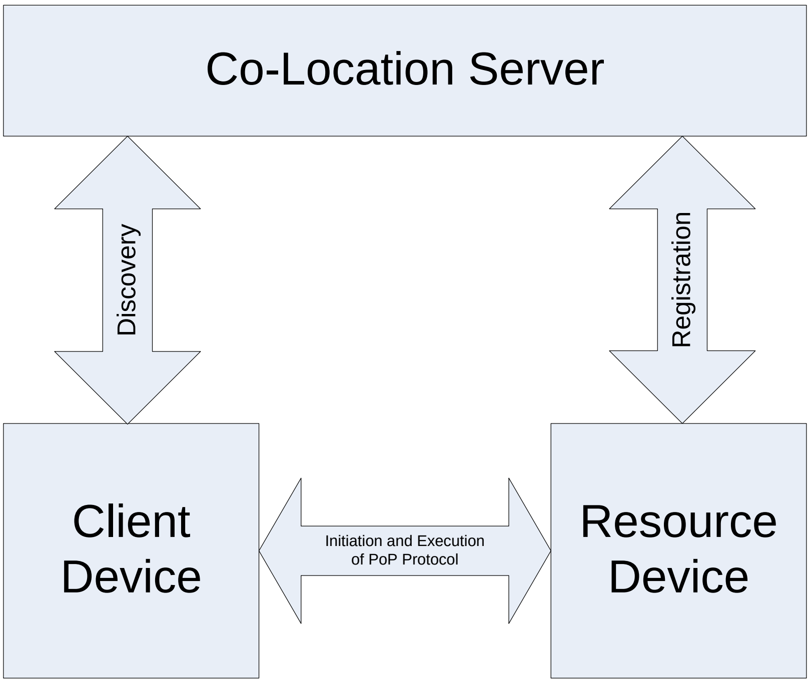

Figure 3.1: High-level architecture of the proposed system

We advocate that device or service registration and discovery mechanisms play an essential role in modern communication systems and there is an immense literature on service discovery to date. However, as discussed in chapter two, most of the prior work on device pairing considered a demonstrative approach to identification and discovery of the intended co-located device and it is assumed that the discovery process would be done by the user. Additionally, we also argued that most of the time, it is difficult for an ordinary user to identify the correct discovery option for a given scenario and it should not be the user's responsibility to figure out how and which method to use for device discovery each time, instead an automatic device discovery should take place. In pursuit of this argument, we have designed and integrated a device registration and discovery mechanism (i.e. first two phases) in the proposed system, which is described in section 3.4.1.4 preceded by a brief summary of some of the relevant discovery schemes.

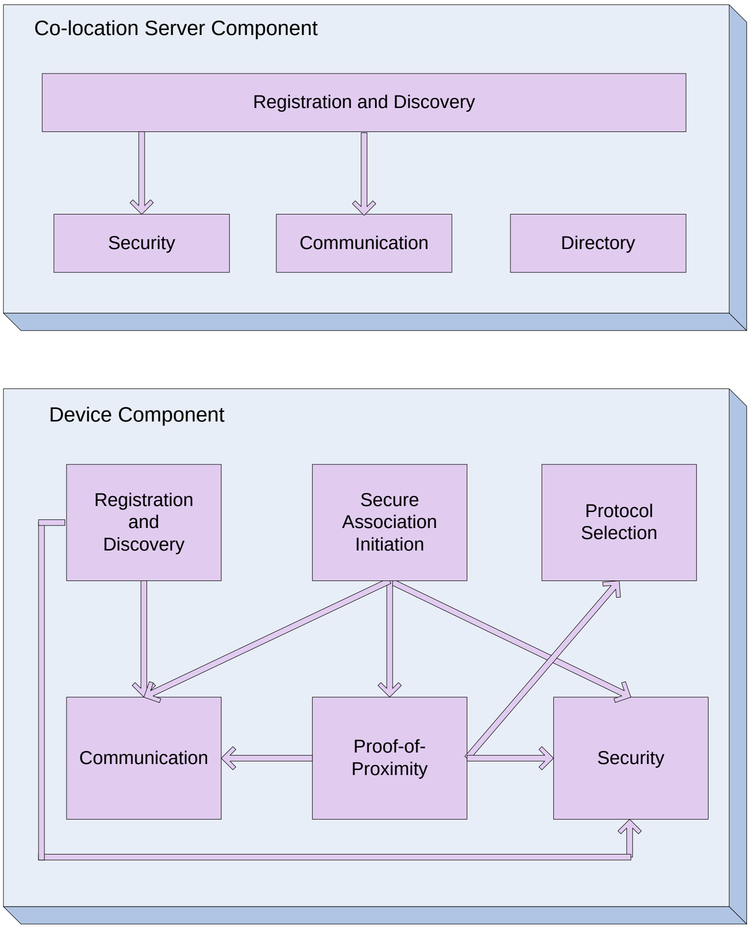

Figure 3.2: Two major components of the proposed system

In the third phase, the goal is to prove that both of the devices (i.e. client and resource device) are in close physical proximity through the use of one of several available PoP protocols. The registration, discovery and proof of physical proximity are integrated into Co-Location (CoLoc) protocol, which is core part of the system and one of our main contributions as well. The CoLoc protocol is described in section 3.5 in detail, however we are presenting the overview of the overall system as below:

1. First of all resource device(s) register their capabilities with an easily found database stored on the co-location server. New devices can be added while the system is running.

2. When two devices need to associate, the client queries the co-location server to acquire the required information of suitable resource device(s).
3. The co-location server prepares a device list containing necessary information for selecting and contacting the resource device in order to initiate the proof-of-proximity phase.
4. Based on the information from the co-location server and user preferences, the client first goes through the PoP protocol selection process and then initiates the secure association initiation process with the selected resource device. Different interactions to demonstrate physical proximity are possible and the selection requires a selection criterion along with device capabilities, constraints on pairing schemes and/or user preferences.
5. Both of the devices (i.e. client and resource) execute the commonly agreed PoP protocol for the purpose of demonstrating their physical proximity in order to establish the secure session. Note that secure pairing is achieved only when physical proximity between both of the devices is proved.

3.4.1 DEVICE(S) REGISTRATION AND DISCOVERY MECHANISM

As described earlier there is a huge amount of literature on service discovery in general; however during the last ten years many discovery protocols have also been proposed to facilitate dynamic discovery of services/devices in ubiquitous computing environments. Some well known discovery protocols include Service Location Protocol (SLP) [62], Bluetooth Service Discovery Protocol (SDP) [63, 64], Microsoft's Universal Plug and Play (UPnP) [65] and Jini [52, 66]. Each has its own design considerations. For example, SDP supports only Bluetooth device/service discovery; while Jini is restricted to Java applications, SLP and UPnP are designed for TCP/IP networks; however UPnP is targeted to small or home based computing environments, while SLP is targeted to both from small to large-scale enterprise networks. Detailed comparisons of discovery protocols can be found in [67-69]. However for the sake of completeness of this thesis we are presenting some of the relevant discovery protocols followed by the proposed device registration and discovery mechanism.

3.4.1.1 SERVICE LOCATION PROTOCOL (SLP)

Service Location Protocol (SLP) [62, 68, 70] is a discovery protocol developed by the IETF (Internet Engineering Task Force) working group for service registration and discovery within a particular location or scope. It is designed for small to large scale enterprise networks. There are three major components of SLP, which are known as agents: Directory Agent (DA), User Agent (UA), and Service Agent (SA). DA is responsible for providing the directory services. SA advertises the location information along with the service-attributes on behalf of a service through registration process, and UA on behalf of the client application sends service discovery requests to a DA.

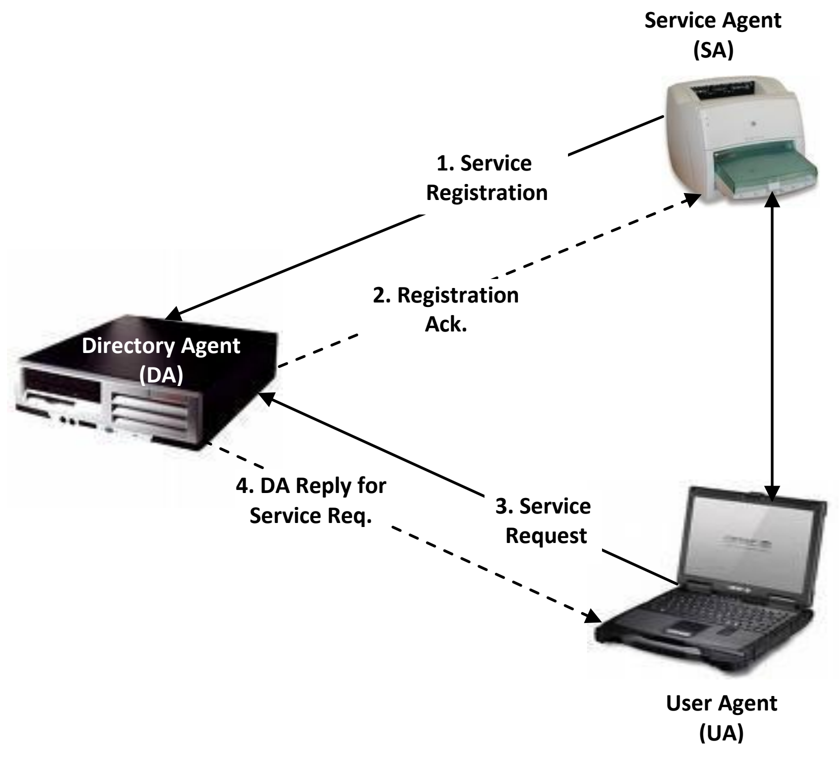

Figure 3.3: An illustration of service registration and discovery mechanism in SLP

Figure 3.3 illustrates the interaction mechanism between the three components of SLP in a small or Local Area Network. First of all DA announces its presence through periodic scoped-multicasting on a well known channel. A UA or SA discovers the address of the DA through some mechanism, such as listening to a DA advertisement message passively or actively multicasting discovery message to the SLP multicast network address (i.e. 239.255.255.253). It is also possible to configure the DA address statically through Dynamic Host Configuration Protocol (DHCP**)** [71].

Once an SA discovers a DA, it registers with it by sending a service registration message. Service advertisements are made through the use of a service URL and service template. The registration message contains the URL for the advertised service including its lifetime. An SLP service is described in the form of set of service attribute-value pair. A sample SLP service template for a print service is given below:

```
service:printer://lj2420dn.FONT.susx.ac.uk:1024/
scopes = FONT, adminstrator
printer-name=lj2420dn
printer-network-name = Inf-pev-5c4-bw
printer-location = Pevensey II, Room 5C4
color-supported = true
...
...
...
```

As stated, when a UA requests a service, it contacts a known DA by sending a service request/query to obtain the service URL. Once the UA receives the service URL, it can access the service pointed to by the returned URL. DA is an optional component in SLP; therefore, in the scenarios where there is no DA available, the UA and SA discover each other directly through a multicast mechanism.

3.4.1.2 JINI TECHNOLOGY

Jini [66] is a java-based service registration and discovery technology developed by Sun Microsystems. Jini provides service/device registration, discovery and communication mechanisms for ad hoc networks. The core part of Jini technology is a set of protocols known as discovery-join-lookup.

Figure 3.4 illustrates the functionality of these protocols. On bootstrapping, services look for a lookup service and register themselves with it. This process is known as the Discovery and Join process. During the registration process Jini services upload their service-object along with service attributes in a Lookup Table of the lookup service. Then, when a client needs a service, it also looks for a lookup service to find out the required services and to download the service-object. Once the client downloads the service-object from the lookup service, it directly contacts the service for further communication. This is known as the Lookup process. As in SLP, Jini lookup servers containing Lookup Tables serve the purpose of a directory. However, unlike SLP, Jini does not support directory-less mode and it always needs at least one lookup service. Further, in contrast to SLP, Jini services are described in Java.

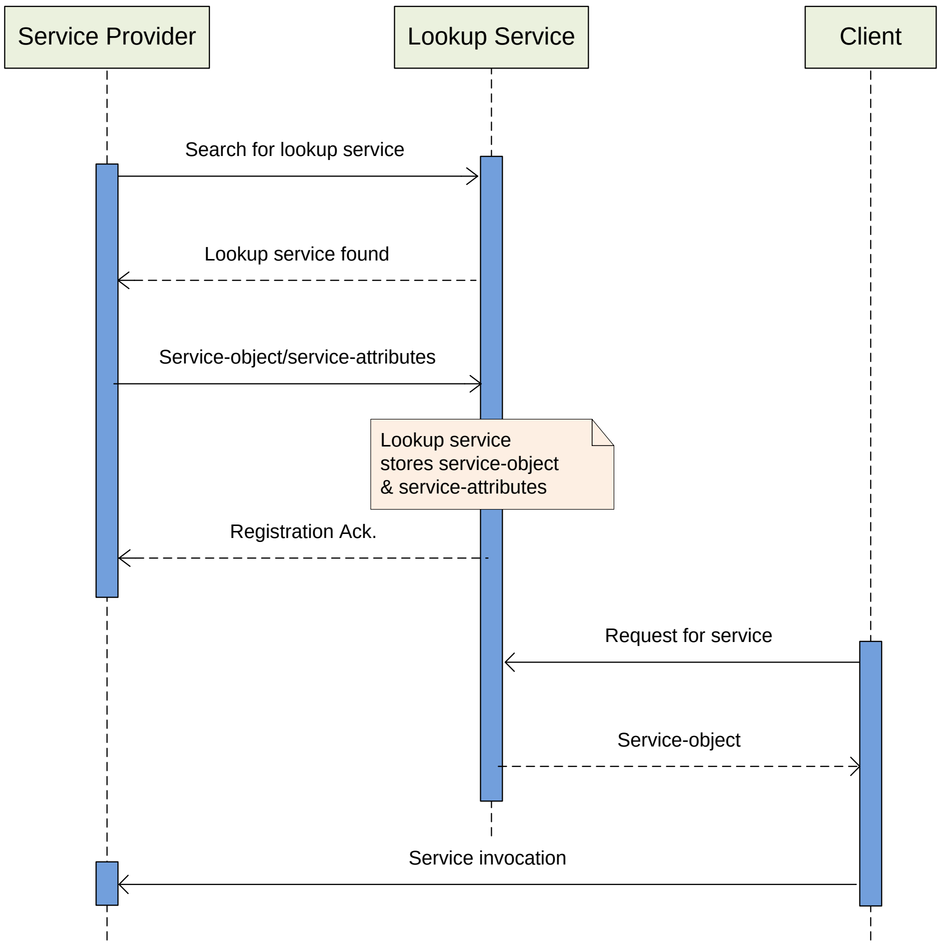

Figure 3.4: Message sequence diagram illustrating Jini's discovery-join-lookup process

3.4.1.3 Universal Plug and Play (UPnP)

Universal Plug and Play (UPnP) is a device-centric peer-to-peer technology developed by the UPnP Forum [65]. Microsoft Corporation played an important role in UPnP’s development and it is considered as an extension to the Microsoft’s Plug and Play technology; however it is more than just an extension. The major objective of UPnP is to enable discovery, auto-configuration, management and control of devices in unmanaged and small computing environments, such as small office or home environments. UPnP achieves its goal through utilizing existing standards, such as web and TCP/IP technologies. For service/device discovery, it uses Simple Service Discovery Protocol (SSDP) [72]. As in Jini’s discovery-join-lookup process, SSDP is used for both advertising the device’s (service) presences to the other devices in the proximity/scope as well as discovering other peer devices. However, unlike SLP and Jini, UPnP does not require any central repository to store the service or device information and/or service-object. Further, in contrast to SLP and Jini, UPnP uses XML for all the communication and exchange of device’s information among the two entities of UPnP network (i.e. Control Point and Device). Devices’ profiles describing their capabilities and features are written in XML format. Interested readers can refer to [65] for details of the UPnP device architecture provided by UPnP forum.

3.4.1.4 The Proposed Device Registration and Discovery Mechanism

When we analyze the previously described discovery schemes, it is noted that security has never been a major concern or major design goal of these technologies. For example, Jini uses non-encrypted Remote Method Invocation (Java RMI) for all the communication that makes it susceptible to eavesdropping. Additionally, when a client wants to create an association with the resource/service, as a part of this process the service-object is downloaded from the Jini lookup service and this introduces the overhead in the sense that small devices have scare resources, and also there is a security risk in that an adversary can register a bogus service containing malicious code as its service-object.

Further, it is also noted that at very basic level, the architecture of these discovery protocols is similar; however each has some of its own assumptions and features that make it feasible for implementing in particular scenarios or environments. For example, SLP and UPnP are targeted to IP based device environments, while Jini is not restricted to IP-based environments; however it requires JVM (Java Virtual Machine). UPnP utilizes XML technology for device/service registration and discovery, while it is not the case in SLP and Jini. In Jini, the client requires more processing capability as compared to UPnP and SLP due to the installed JVM and downloaded service-object. It is not a big deal for large devices, such as desktop computers or laptops; however it is still challenging for small resource-constrained devices. Another reason that limits the widespread use of Jini is the lack of support for J2ME-based devices, because when Jini was introduced there was no support for RMI in J2ME; however it is supported now.

In summary, to simplify the analysis, design and prototype implementation of the proposed framework to test our hypothesis, we decided to design our own registration and discovery mechanism. Our proposed discovery and registration system incorporates several similar features to the device discovery technologies discussed above along with some of its own unique features to make the registration and discovery process simple, easy to implement, independent of existing technologies, and confidentiality and integrity protected. For example, like UPnP we have used XML to describe the registration and discovery messages mechanism for the proposed system, however in a much simpler way than UPnP. The reason we use XML is that it is an advantage for any modern communication system due to its flexibility, programming language independence and portability characteristics. It is flexible enough that one can incorporate additional features in the system later on, if necessary, and it also significantly increases interoperability between systems.

Further, unlike Jini, where devices look for a lookup service through multicasting a search request on the network, in the proposed system the co-location server advertises itself through multicasting. We choose this mechanism as transmitting data consumes more battery power as compared to receiving data, and the devices in ubiquitous computing environments are more battery-constrained as compared to the server or base station. In our proposed system, the registration process could be considered equivalent to Jini’s discovery and join process, and the discovery process could be considered equivalent to Jini's lookup process. Also note that in our proposed mechanism, all the communication during registration and discovery process between several entities of the system is encrypted, which is described in section 3.5.

```
<DeviceProfile>
        <DeviceID></DeviceID>
        <LDuration></LDuration>
        <Keyword></Keyword>
        <DeviceLocation></DeviceLocation>
        <CommProtocol>
                <ChannelName></ChannelName>
                <Address></Address>
        </CommProtocol>
        <DeviceCap></DeviceCap>
        <UserInput></UserInput>
</DeviceProfile>
```

Figure 3.5: XML-based device description template

```
<!ELEMENT DeviceProfile
 ( DeviceID, LDuration, Keyword, DeviceLocation?, CommProtocol+,
DeviceCap, UserInput? ) >
<!ELEMENT DeviceID ( #PCDATA ) >
<!ELEMENT LDuration ( #PCDATA ) >
<!ELEMENT Keyword ( #PCDATA ) >
<!ELEMENTDeviceLocation ( #PCDATA ) >
<!ELEMENT CommProtocol ( ChannelName, Address ) >
<!ELEMENT ChannelName ( #PCDATA ) >
<!ELEMENT Address ( #PCDATA ) >
<!ELEMENT DeviceCap ( #PCDATA ) >
<!ELEMENT UserInput ( #PCDATA ) >
```

Figure 3.6: DTD file for device description/profile

There are several ways to write device descriptions, e.g. Composite Capability / Preference Profiles (CC/PP) [107], However, for the purpose of simplicity, we preferred to describe our own device template. Figures 3.5 and 3.6

show the XML based device description template and its corresponding Document Type Definition (DTD) respectively, which in contrast to UPnP device template are simple to understand and implement. DTDs contain information about an XML document's structure. For example, it holds information about what elements might be included in an XML document, what attributes these elements might have, and what might be the ordering of these elements, etc. It is not compulsory for every XML document to have its corresponding DTD; however it is good practice to use DTDs in order to ensure the conformity or validity of an XML document.

Another way to define the structure of an XML document is XML Schemas [73], which are more effective than DTDs. For example, DTDs provide support for only text data type; while XML Schemas support a wide range of data types including custom data types, and are useful when dealing with XML documents containing letters and numbers or having some restrictions on the acceptable data for its elements/attributes. We are fully aware of the fact that XML Schemas are more powerful than DTDs; however for the sake of simplicity we have used DTDs. Additionally, if required, it is now easy to convert DTDs into XML Schemas automatically using one of the several available DTD to XML Schema converters/utilities, such as [74].

3.5 CO-LOCATION (COLOC) PROTOCOL

The co-location (CoLoc) protocol is a core part of our system and one of our main contributions. It is designed to achieve our generality, usability and security goals. It provides the functionality of registration, discovery, and security association initiation and execution of the selected PoP protocol. For the sake of simplicity and clarity, we have divided the overall protocol into three parts: registration, discovery of intended pairable device, and the selection and execution of an appropriate protocol to demonstrate/authenticate the physical proximity. The selection process involves device capabilities, constraints on pairing schemes and/or user preferences. The detailed description of each of the parts can be found in subsequent sections preceded by the description of several notations used in describing the CoLoc protocol.

3.5.1 NOTATIONS

CS: Co-location server

A: Resource device

B: Client device

Process_i: Actions/processes performed at device *i* before sending or receiving a message.

X → Y: Msg : A message **Msg** sent from **X** to **Y** over a communication channel.

$PK_{i:}$ Public key of *i*.

K_i: Private or secret key of *i*.

SK_i: Session key internally generated by *i*.

PSK: Pairing session key.

PSK_{ij}: Shared pairing session key for the parties *i* and *j*.

CP: Credential password, shared among all the registered devices and co-location server.

Enc(): Encryption function.

Dec(): Decryption function.

Enc(x)y: An encryption function that encrypts plaintext x using key y, which could be a public/private key or shared secret key.

Dec(x)y: A decryption function that decrypts ciphertext x using key y, which could be a public/private key or shared secret key.

MAC(x)y: A keyed message authentication function that is applied to x using key y.

||: Concatenation operator

3.5.2 BOOTSTRAPPING

Bootstrapping in our system refers to the initialization and advertisement of the co-location server. During bootstrapping, the co-location server generates its public/private key pair (i.e. PK_{Coloc} and K_{Coloc}) and broadcasts its connectivity information along with its public key. Then, devices discover the co-location server for registration and/or discovery tasks by listening to the broadcast messages. Alternatively, in certain scenarios, where this mechanism is not available or difficult to implement, a one-time demonstrative discovery of the co-location server can be performed, which has now become more common in the literature of device pairing. As described in chapter 2, in this approach the user is involved in identifying and obtaining the connectivity information of the resource or intended communicating partner through some manual effort.

3.5.3 REGISTRATION AND DISCOVERY PART OF THE COLOC PROTOCOL

Figure 3.7 shows the registration part of the CoLoc protocol. Once the system is bootstrapped and device A receives the public key PK_{Coloc} of the co-location server, it encrypts the device profile with an internally generated temporary session key SK_A. Then it sends an encrypted message along with a message authentication code (MAC), and SK_A encrypted with the co-location server's public key PK_{Coloc} to the server. The device profile contains the id of device A along with connectivity information and some keywords (user friendly names) to identify the device in the networked environment, capability information (such as camera, display, keypad, etc), lease duration and optionally device location information (such as Pevensey II, Room 5c11, etc). Additionally, any constraints or user input/preferences are also injected in the DeviceProfile. The complete DTD, which is used for validating the protocol's message including device profile, query and co-location server's response for client's discovery request is given in Implementation chapter. However, a sample DeviceProfile is given in figure 3.8.

Process_CS:

Generate public/private key pair: PK_{Coloc}, K_{Coloc}

CS → A: ServerID,PK_{Coloc}

Process_A:

RegistryMsg:=Enc($DeviceProfile_A$)SK_A|| Enc(SK_A)PK_{Coloc}, MAC(RegistryMsg)SK_A

A → CS: RegistryMsg

Process_CS:

splits and Decrypt *RegistryMsg* to obtain SK_A first, and then *DeviceProfile*.
If(integrityChecked()), then
- (a) Generate: PSK_A
- (b) Register Device
- (c) ResponseMsg:= Success || PSK_A || CP

Else
- (a) ResponseMsg:= Abort/Fail

End If
RegAck:= Enc(ResponseMsg)SK_A
Destroy session key: SK_A

CS → A: RegAck

Process_A:

Dec(RegAck)SK_A to obtain PSK_A

Figure 3.7: Registration part of the CoLoc protocol

```xml
<DeviceProfile>
    <DeviceID>wiston</DeviceID>
    <LDuration>4</LDuration>
    <Keyword>Desktop PC;Computer</Keyword>
    <DeviceLocation>Netlab;Room 5C11</DeviceLocation>
    <CommProtocol>
        <ChannelName>Bluetooth</ChannelName>
        <Address>000A3A7E4CA2</Address>
    </CommProtocol>
    <CommProtocol>
        <ChannelName>TCP-IP</ChannelName>
        <Address>192.168.0.2:8009</Address>
    </CommProtocol>
    <DeviceCap>Display;Keypad;Button;Speaker;LED</DeviceCap>
</DeviceProfile>
```

Figure 3.8: A sample device profile

The co-location server splits and decrypts the registration message in order to obtain the SK_A first, which is then used to obtain the device profile. The co-location server also performs integrity check before registering the device. In response to a registration request, the co-location server sends an acknowledgement message to the device A, containing a one-time pairing session key PSK_A and credential password CP encrypted with temporary session key SK_A. Credential password CP is used in revocation mechanism, which is described in detail in section 3.7.2. The registration process applies to every device intended to become part of the deployed ubiquitous system. Once registration is done, device A will be visible to the other devices (i.e. clients) through the querying co-location server.

Process_B:

DiscoveryReq:=Enc(Query)SK_B || Enc(SK_B)PK_{Coloc},
MAC(DiscoveryReq)SK_B

B → CS: DiscoveryReq

Process_CS:

Split and Decrypt *DiscoveryReq* to obtain SK_B first, and then *Query*.
If(integrityChecked()), then
(a) Generate *ResultSet* containing matching devices' profiles and their corresponding pairing session keys along with expiry time.
(b) DiscoveryRespMsg:= Enc(ResultSet)SK_B
Else
(a) DiscoveryRespMsg:=Abort/Fail
End If
Destroy session key: SK_B

CS → B: DiscoveryRespMsg

Process_B:

Dec(DiscoveryRespMsg)SK_B
to obtain devices' information along with their pairing session key(s) and their expiry time.

Figure 3.9: Discovery part of the CoLoc protocol

In the discovery process (figure 3.9), device B (client) encrypts a query with temporary session key SK_B. Then, it encrypts SK_B with PK_{Coloc} and sends it to the co-location server along with the encrypted query and MAC of the overall message. The co-location server decrypts the client message and also performs an integrity check before going through the match-making process based on the criteria given in the query. Query contains the user-friendly name (if known) or the type of device, any user preferences for pairing process and optionally the locations in which devices should be searched (if server domain is too broad). A sample query is shown in figure 3.10.

```xml
<?xml version="1.0" encoding="ISO-8859-1" ?>
<ProtocolMsg>
    <Command>Discovery</Command>
    <XMLData>
        <Query>Keyword|Computer;Location|Room 5C11</Query>
    </XMLData>
</ProtocolMsg>
```

Figure 3.10: A sample query for device discovery

As a consequence, the co-location server sends information on matching devices (referred to as a ***ResultSet*** hereafter in this thesis) to the client (i.e. device B) encrypted with SK_B. *ResultSet* contains the profiles of found devices based on the criteria given in query along with their one-time pairing session keys (i.e. PSK_i) and their expiry time.

3.5.3.1 Registration Renewal, Update and Device De-Registration

In the proposed system, the registered devices are capable of renewing or updating their registration. Renewal and update requests can be for updating/modifying the device’s profile or device status (i.e. busy or available), and extension/renewal of the lease time and/or pairing session key. Explicit de-registration can be performed on the demand of the registered device by sending a de-registration request to the co-location server. The co-location server also performs implicit de-registration when the device lease time expires to keep the registered devices’ information up-to-date, and to maintain the device’s directory. During the implicit de-registration process, any device whose lease time expires is automatically de-registered by the co-location server by deleting their entry from the directory.

3.5.4 SELECTION AND EXECUTION OF MUTUALLY AGREED SCHEME

As shown in figure 3.11, during this phase the client sends a message, containing the name of the selected PoP protocol, to the resource to initiate the pairing process. Once the resource device receives that message, it starts generating PoP data that will be used to verify the physical proximity of the devices. PoP data could be generated in numerous ways based on the nature of agreed protocol. For example, many modern devices carry sensors for other purposes, which could be used to obtain the PoP data. Where sensors are not available or it is hard to obtain PoP data directly from sensors, then user could be involved to get the PoP data. Considering the nature and ways of demonstrating the physical proximity of devices, PoP protocols are classified into four categories. The first category belongs to those protocols, which require user involvement in only generating PoP data (such as Button-to-Button and Blink-to-Button). In that case, verification of PoP data is done internally by the system. The second category belongs to those schemes which require user involvement only in verification of PoP data (such as Display-Display and Blink-Blink). In that case PoP data is generated either internally by the system or from attached sensors with the devices. The third category belongs to those schemes, which require the user to be involved in generating PoP data as well as in verifying that data (such as Capture and Show, which is described in chapter 4). The fourth category belongs to those schemes, which do not involve the user in the proof-of-proximity process at all, so we call them automatic pairing schemes. We have further described each of the implemented schemes in chapter 4.

At the end of execution of this phase, if the physical proximity has been proved, the established session between both of the devices is considered to be secure. Then, it is possible to establish the long-term connections by using other well known cryptographic protocols/schemes [34].

Process_B:

Execute an algorithm that seeks for the best possible available mutually supported schemes that could be used to authenticate/demonstrate physical proximity between the devices.
Inv_PairingMsg:=Enc(SelectedProtocolName)PSK_{AB}

B → A: Inv_PairingMsg

Process_B:

Generate PoP data to verify the physical proximity based on the nature of agreed PoP protocol.

Process_A:

Dec(Inv_PairingMsg)PSK_{AB} to retrieve the name of pairing scheme.
Generate PoP data that will be used to verify the physical proximity between the devices.
Resp_Msg:= ENC(Resp_InvPairingMsg)PSK_{AB}

A → B: Resp_Msg

Process_B:

DEC(Resp_Msg)PSK_{AB}
ProximityVerMsg:= ENC(PoP data)PSK_{AB}

B → A: ProximityVerMsg

Process_A:

DEC(ProximityVerMsg)PSK_{AB}
Perform demonstration of physical proximity based on the previously agreed PoP protocol.
If(success)
    pairingA = 'Accepted'
Else
    pairingA = 'Rejected'
End If
ProximityVerResp:= ENC(pairingA)PSK_{AB}

Process_B:

Perform demonstration of physical proximity based on the previously agreed PoP protocol in parallel to device A.
If(success)
    pairingB = 'Accepted'
Else
    pairingB = 'Rejected'
End If

A → B: ProximityVerResp

Process_B:

DEC(ProximityVerResp)PSK_{AB} to obtain pairingA.
Pairing is achieved/granted only when
    (pairingA= 'Accepted' & pairingB = 'Accepted')
Otherwise B aborts the pairing process

Figure 3.11: Secure association initiation and execution of PoP protocol

3.6 SELECTION OF POP PROTOCOL(S)

As described earlier, once the device discovery operation completes, the subsequent phase is the selection and execution of the PoP protocol. To achieve the objective of selecting an appropriate PoP protocol, we have designed a protocol selection algorithm presented in figure 3.12. The input parameters of algorithm include client's own device profile, resource device profile and an XML-based PoP protocol specification and selection policy.

Input:

- Client's own profile,
- Resource's profile,
- Protocol specification and selection policy

Output:

RecommendedProtocol(s) based on the given input parameters

Step 1: Cl_Profile:= Read client's own profile

Step 2: Res_Profile:= Read resource's profile (i.e. received from Co-location server)

Step 3: Cl_SupportedProtocols:=getProtocolList(Cl_Profile)

Step 4: Res_SupportedProtocols:=getProtocolList(Res_Profile)

Step 5: Comm_SupportedProtocols:=mutualProtocolList(Cl_SupportedProtocols, Res_SupportedProtocols)

Step 6: RecommendedProtocols:=getBestProtocols(Comm_SupportedProtocols, Cl_Constraints/Preference, Res_Constraints/Preferences)

Step 7: Return: RecommendedProtocols

Figure 3.12: An algorithm to find out the best possible PoP protocol(s) based on given input parameters

A sample protocol specification and selection policy is shown in figure 3.13; however it's associated DTD is given appendix C.

```xml
<?xml version="1.0" encoding="ISO-8859-1" ?>

<PSPolicy>
    <Protocol>
        <Name>Button_to_Button</Name>
        <Type>1</Type>
        <ClCapabilities>Button</ClCapabilities>
        <ResCapabilities>Button</ResCapabilities>
        <ProximityLimit>100</ProximityLimit>
        <UILevel>1</UILevel>
    </Protocol>
    <Protocol>
        <Name>Capture_And_Show</Name>
        <Type>3</Type>
        <ClCapabilities>Camera;Display</ClCapabilities>
        <ResCapabilities>Display</ResCapabilities>
        <ProximityLimit>200</ProximityLimit>
        <UILevel>3</UILevel>
    </Protocol>
    <Protocol>
        <Name>Display_Display</Name>
        <Type>2</Type>
        <ClCapabilities>Display</ClCapabilities>
        <ResCapabilities>Display</ResCapabilities>
        <ProximityLimit>100</ProximityLimit>
        <UILevel>1</UILevel>
    </Protocol>
    <Protocol>
        <Name>Display_Speaker</Name>
        <Type>2</Type>
        <ClCapabilities>Display</ClCapabilities>
        <ResCapabilities>Speaker</ResCapabilities>
        <ProximityLimit>100</ProximityLimit>
        <UILevel>1</UILevel>
    </Protocol>
</PSPolicy>
```


Figure 3.13: A sample protocol specification and selection policy

<Name> tag contains the name of the PoP protocol for which other tags describe the selection criteria. The value of <Type> tag represents one of the categories of PoP protocols, which are briefly described in previous section and other details of these categories is given in chapter 4. The values of <ClCapabilities> and <ResCapabilities> tags describe the required capabilities of client and resource devices for the execution of the protocol. The value of <ProximityLimit> tag represents the maximum distance between the pairing partners up to which the protocol can work or can achieve good results. The value of <ProximityLimit> is

given in centimeters. The value of <UILevel> represents the level of required user interaction. ‘1’ represents the low or minimum level of user interaction and ‘3’ represents the high or maximum level of user interaction. These values are obtained based on the classification of PoP protocols presented in Chapter 4.

3.6.1 INTERNAL WORKING OF PROTOCOL SELECTION ALGORITHM

The protocol selection algorithm is consisting of several rounds. Each round facilitates with the filtration process of PoP protocols. During each round those PoP protocols are discarded which does not meet the requirements of some particular constraint of that round. Ultimately in round-5, we obtain those PoP protocols, which fully satisfy the user preferences and other requirements of the scenario/situation in which pairing process is going to be occurred. We describe each round of the execution of PoP protocol selection algorithm as below:

Round-1: (Input: client-device profile, resource-device profile, PoP protocols specification and selection policy)

Filter/select PoP protocols based on required capabilities of client and resource devices (refer to figure 3.13, ClCapabilities and ResCapabilities).

Round-2: (Input: selected PoP protocols from Round-1 and the PoP protocols specification and selection policy).

Select the PoP Protocols that are appropriate for working within the given distance. It is achieved through comparing and performing selection based on the value of ProximityLimit tag (figure 3.13) with the distance given/input by the user (figure 4.8).

Round-3: (Input: selected PoP protocols from Round-2 and the PoP protocols specification and selection policy).

Select PoP protocols based on the level of required user interaction during pairing process. It is achieved through comparing the value of UILevel tag (figure 3.13) with the user-interaction option as selected by the user (figure 4.8).

Round-4: (Input: selected PoP protocols from Round-3 and PoP protocols specification and selection policy).

Select PoP protocols based on the constraints/limitations of PoP protocols and user preferences. It is achieved through comparing the value of Constraints tag (refer to appendix C) with the given user preferences (figure 4.8).

Round-5: (Input: selected PoP protocols from Round-4 and PoP protocols specification and selection policy).

In this final round, the priority level/recommended order is assigned to each of the PoP protocols obtained from Round-5. The high-level description of the calculation process for priority-level is described below. Note that the scores/points used in these calculations are only for demonstration and proof-of-concept purposes.

PoP protocol points calculation process from security point of view:

```
If(fatal errors are not applicable to PoP protocol)
        FatalErrorPoints = 4;
Else
        FatalErrorPoints = 2;
EndIf
If(safe errors are not applicable to PoP protocol)
        SafeErrorPoints = 2;
Else
        SafeErrorPoints = 1;
EndIf
```

Note that the points for fatal errors and safe errors differ from each other due to the fact that fatal errors are more dangerous and serious than safe errors.

PoP protocol points calculation process from execution-time point of view:

```
If (ProtocolExeuctionTime <= 15 seconds)
        ExecutionTimePoints = 5;
ElseIf (ProtocolExeuctionTime > 15 seconds and <= 30 seconds)
        ExecutionTimePoints = 4;
ElseIf (ProtocolExeuctionTime > 30 seconds and <= 45 seconds)
        ExecutionTimePoints = 3;
ElseIf (ProtocolExeuctionTime > 45 seconds and <= 60 seconds)
        ExecutionTimePoints = 2;
ElseIf (ProtocolExeuctionTime > 60 seconds and <= 75 seconds)
        ExecutionTimePoints = 1;
Else
        ExecutionTimePoints = 0;
EndIf
```

Based on the above mentioned points calculation process, the level of priority is calculated, which eventually sets the recommended order of the PoP protocols. Rule is that the PoP protocol that has highest score/points will the best protocol for a given scenario/situation.

3.7 MESSAGE SEQUENCE DIAGRAM

In order to summarize the work presented so far, in figure 3.14, we have presented the message sequence diagram of the overall system.

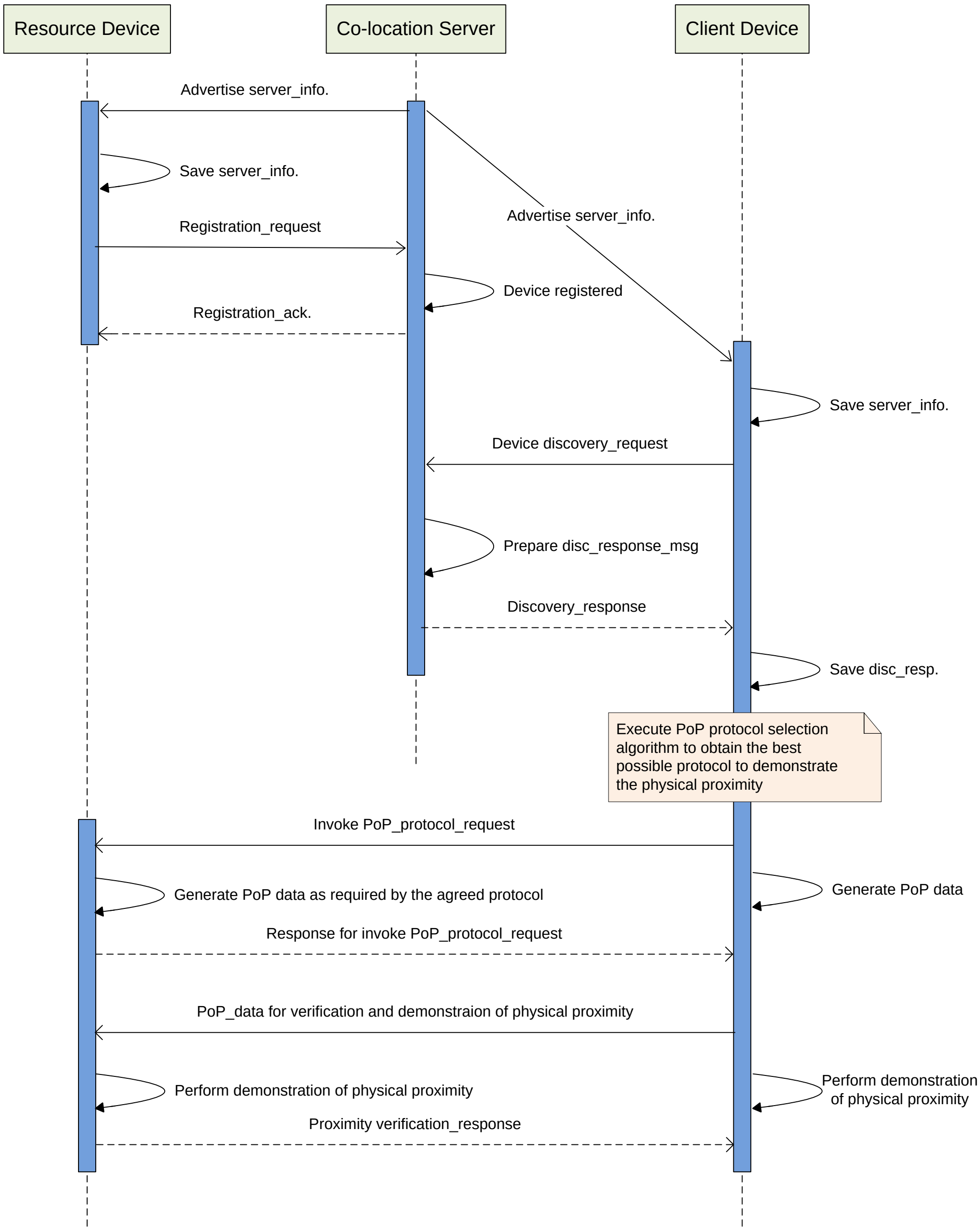

Figure 3.14: Message sequence diagram of the system

3.8 Extended Features of the System

Initially, we implemented the core system as described in previous sections and shown in figure 3.14. Once the core system was complete, we developed other important features as part of it. Since these advanced features are added as an incremental step, we refer them as the extended features of the proposed system. The inclusion of these features in incremental fashion demonstrates that the proposed system has the potential to be extended without changing its core design.

3.8.1 LONG-TERM DEVICE PAIRING

As stated earlier, once a secure session is established between the intended communicating partners, a long term pairing can easily be established through the use of some standard cryptographic protocol [34]; however there is a need to manage and maintain these pairings, and to facilitate the users in securely reconnecting previously paired devices without going through the discovery and the proof-of-proximity phases again. The issue of managing long-term pairings – including un-pairing and revocation mechanisms – is an important aspect of secure device pairing; however, it is not given much attention in most of the recently proposed solutions to device pairing. We realized the importance of this aspect and extended the device component of the core framework through the inclusion of the device pairing repository (DPR) software component. DPR works in association with the secure association initiation and proof of proximity components. As the directory component in the co-location server stores the profile information of the devices, the DPR component stores the information about successfully created pairings for future use. The core responsibility of the DPR component is to facilitate with storing, retrieving, modifying, updating and deleting the information relating to successfully established pairings.

The DTD for a DPR entry is shown in figure 3.15. Each DPR entry consists of several pieces of information, which include a user-friendly name for the successfully created pairing, its life time, device id, connectivity information for the paired device, and the information required for setting up a secure connection in the future (i.e. the type of algorithm, key-length, the actual key, and the credential password).

```
<!ELEMENT DevicePairing ( PairingName, PairingExpiry, DeviceID, CommProtocol, SecurityData ) >
<!ELEMENT PairingName ( #PCDATA ) >
<!ELEMENT PairingExpiry ( #PCDATA ) >
<!ELEMENT DeviceID ( #PCDATA ) >
<!ELEMENT CommProtocol ( ChannelName, Address ) >
<!ELEMENT ChannelName ( #PCDATA ) >
<!ELEMENT Address ( #PCDATA ) >
<!ELEMENT SecurityData ( Type, Key, CP ) >
<!ELEMENT Type ( #PCDATA ) >
<!ELEMENT Key ( #PCDATA ) >
<!ELEMENT CP ( #PCDATA ) >
```

Figure 3.15: The DTD for a device pairing repository (DPR) entry

3.8.2 Device Un-Pairing and Revocation Mechanism

We have classified the un-pairing as either explicit self un-pairing or selective on demand un-pairing. In the case of explicit self un-pairing, the DPR component provides the interface to the user/device owner for deleting either a particular or all of the previously created and stored pairing(s). The deletion process removes the pairing entry from the device pairing repository and thus causes the device to forget all the security relevant information for any previously established pairing(s). The advantage of this self un-pairing mechanism is twofold: firstly, if the device owner wants to sell the device, he/she can explicitly deletes all the required existing pairings before handing it to the new device owner. Secondly, if the client device is stolen or compromised, this mechanism allows the resource device to un-pair itself from client, thus avoiding any possible threats, such as keeping the resource busy in order to launch a DoS attack. Similarly, if the resource device is compromised, this mechanism allows the client device to un-pair itself from the resource (compromised) device.

In the case of selective on demand un-pairing, the client device connects to the resource device in a secure mode and sends it an un-pairing request. As a consequence, the resource deletes all the stored pairing information for the current pairing and sends back an acknowledgement message to the client device. Due to the nature of this scheme, it can also be described as mutual un-pairing. Additionally, the DPR component also performs the implicit un-pairing whereby a pairing is deleted automatically when it expires.

The revocation is given less attention in the literature of device pairing, however we believe that it is an important issue that needs to be addressed when proposing or designing a system that provide secure device pairing. For example, in the case of a paired device theft, loss or compromise, it is necessary that all the other registered devices must be informed in a timely fashion in order to revoke the credentials assigned to that device. The straight forward solution to this problem is the Certificate Revocation List (CRL) scheme [32]. However, this solution is practically infeasible for ubiquitous computing environments due to the fact that ubiquitous computing systems are ad hoc and dynamic in nature, most of the ubiquitous devices are small and resource constrained, and users of these systems are often non-technical. Considering these facts, the proposed system incorporates a simple revocation mechanism, which is based on a credential password (CP) generated by the co-location server and shared among all the registered devices including the co-location server. Once a pairing is established, when two devices need to connect with each other, they must demonstrate possession of the CP first. When any registered device is found to be compromised or stolen, the user provides the details of that device to co-location server through its interface in revoke mode. Then, the co-location server re-generates the CP and broadcasts it along with the device id or friendly name of the compromised device to all registered devices in encrypted mode using their pairing session keys (PSKs), excluding the compromised or stolen device. This way, the devices, which are recipients of the co-location server's revocation message, revoke the credentials assigned to the compromised device and also delete any already established pairing with it. Consequently, when the compromised or stolen device tries to connect and benefit from any of the legitimate devices, its connection is refused due to the lack of credentials as well as its inability to show possession of the updated/modified CP.

3.8.3 SECURE GROUP PAIRING

In addition to long-term device pairing, the proposed system is also capable of establishing secure group communication. Figure 3.16 illustrates the group pairing protocol of the proposed system. In order to describe this protocol, we are following the same notations as outlined in section 3.5.1. Two additional notations used in this protocol are GC (group controller) and M (group member). The group pairing process starts with the registration process, where all the intended members of the group are required to register with the co-location server. Then, a group controller (GC) queries the co-location server to obtain the list of intended group member devices along with their capabilities and connectivity information. The GC establishes a secure session with each of the group members through the execution of one of the PoP protocols followed by either sharing the group key K_{group} or group key material. Once the group key K_{group} or group key material has been shared between the entire group they can establish and maintain secure group communication either through our simplest scheme, standard cryptographic schemes[33, 34], or using more advanced schemes and their variations available in the literature, such as [75-79].

Process_GC:
DiscoveryReq:= Enc(Query)SK_{GC} || Enc(SK_{GC})PK_{Coloc},
MAC(DiscoveryReq)SK_{GC}
GC→CS: DiscoveryReq
Process_CS:
Detach and Decrypt *DiscoveryReq* to obtain SK_{GC} first,
and then *Query*.
If(integrityChecked()), then
(a) Generate *ResultSet* containing matching
devices' profiles and their corresponding
pairing session keys along with expiry time.
(b) DiscoveryRespMsg:= Enc(ResultSet)SK_{GC}
Else
(a) DiscoveryRespMsg:=Abort/Fail
End If
Destroy session key: SK_{GC}
CS→GC: DiscoveryRespMsg
Process_GC:
Dec(DiscoveryRespMsg)SK_{GC}
to obtain devices' information of the intended group
members along with their pairing session key(s)
and their expiry time.

$$\mathbf{GC} \xleftrightarrow[\text{secure session with each of the group member and to share the group key } K_{group} \text{ or group key material}]{\text{Perform PoP phase with each of the intended group member as described in CoLoc protocol in order to create}} \mathbf{M_1, M_2, \ldots, M_n}$$

Figure 3.16: Secure group pairing protocol

In our proposed scheme, when a new group member joins the group, it has to pair with GC first in order to obtain the shared group key K_{group}. The shared group key K_{group} is refreshed and re-distributed by the GC whenever an existing group member leaves the group, a new member joins the group, or a device compromise is detected to maintain the security of the group communication. This mechanism prohibits a former group member from accessing any on-going communication between the current group members and also prevents a new member from accessing and understanding any previously happened communication among the group members.

3.9 SUMMARY

The main goal of this research is to design a generic system that facilitates association of two co-located devices by demonstration of physical proximity in ubiquitous computing environments. To achieve this goal, we propose to integrate a discovery mechanism and a number of pairing scheme (called proof-of-proximity schemes in this dissertation) into a single system. The focus in this chapter was on the architectural view of the proposed system along with the design goals (i.e. generality, security, and usability), requirements and assumptions. A core part of the proposed framework is a novel protocol (CoLoc protocol), which provides the functionality of registration, discovery, and security association initiation and execution of the selected PoP protocol. We described the details of the CoLoc protocol along with the PoP protocol selection mechanism. Finally, we also showed that the proposed system has potential to be extended without changing its core design through the addition of extra features without substantial effort. In next chapter, we shall discuss the prototype implementation of the proposed system.

CHAPTER 4

IMPLEMENTATION

The focus of this chapter is the prototype implementation of the proposed system. In this chapter, we describe the software components of the proposed system, the structure of the CoLoc protocol messages, classification and description of the integrated PoP protocols followed by a demonstration of the prototype implementation.

4.1 IMPLEMENTATION

To evaluate our hypothesis and the proposed system, we built a prototype implementation of the system and conducted a usability study. The focus of this section is the implementation of the CoLoc protocol along with several integrated PoP protocols preceded by describing the software components of the proposed system. The details of usability study are described in chapter 5.

4.1.1 SOFTWARE COMPONENTS OF THE PROPOSED SYSTEM

We have implemented the prototype of the proposed system using Java (version 1.6) and Windows XP operating system. In the coding and implementation process, we have used Eclipse Galileo (version 3.5) as a Java IDE. Additionally, we have also used two PhidgetInterfaceKits [80] and a camera, which are requirement for some of the PoP protocols. The software components of the proposed system are implemented as Java packages, which are described in subsequent sections. Figure 4.1 illustrates the relationship between the software components for the server application, while the relationship between these components on the device application is shown in figure 4.2.

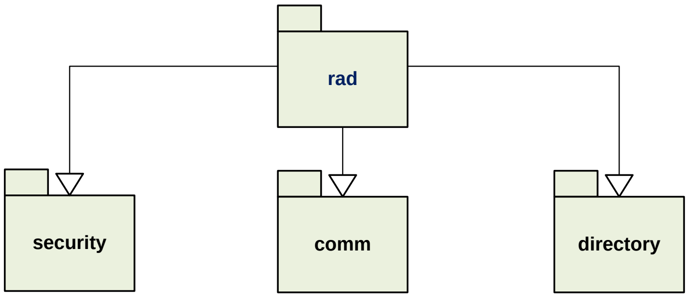

Figure 4.1: Illustrating the relationship between the software components for the server application

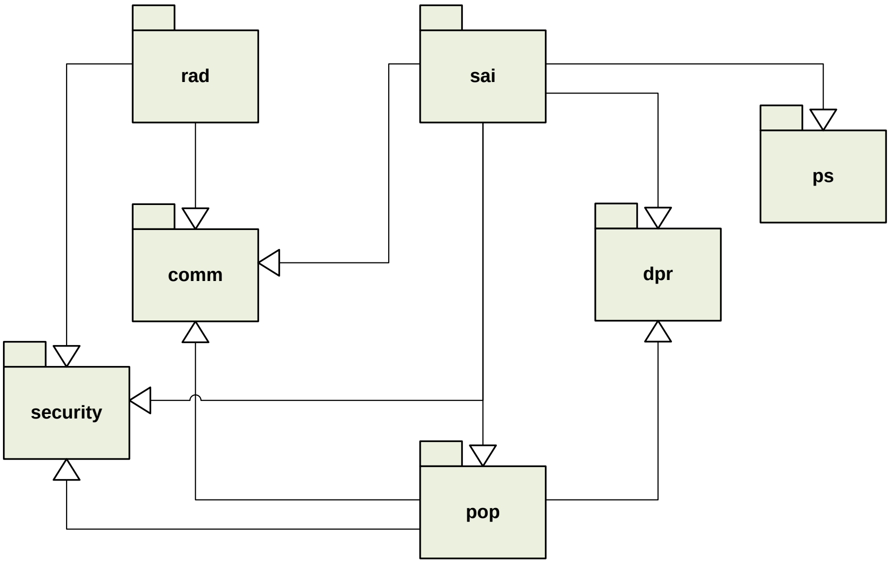

Figure 4.2: Illustrating the relationship between the software components for the device application

4.1.1.1 COMMUNICATION (COMM)

The communication component provides several wrapper APIs to facilitate the establishment of a communication channel between any two entities of the system. It also provides APIs for the broadcast or multicast of co-location server adverts and also provides communication APIs for the devices to discover the co-location server in order to perform subsequent registration and/or discovery operations.

4.1.1.2 REGISTRATION AND DISCOVERY (RAD)

The Registration and discovery component provides a set of APIs for the purpose of registration and discovery of resource devices. Some of the APIs provided by this component also utilize the APIs of communication component in order to exchange the registration and discovery messages between the two entities of the system.

4.1.1.3 DIRECTORY

The directory component of the system provides the functionality of registry. It keeps a record of all the registered devices. It provides APIs to access, read, write and update the directory information. In order to select a database package that suits our needs and can easily be integrated in the prototype implementation of the system, we surveyed several light-weight database packages, such as SQLLite [81], HSQLDB [82], Perst/PerstLite [83] and Berkeley DBXML [84], and finally decided to use Berkeley DBXML [84, 85] to maintain and keep record of the devices' profiles. The brief description of each of the mentioned database packages is given below followed by describing the reasons to choose Berkeley DBXML.

- **SQLITE:** SQLite is an embedded light-weight database management system suitable for implementing in server-less scenarios. It is developed in the C programming language and source code is in the public domain [81]. It implements a database engine, which is self-contained, does not require any configurations and supports many features of SQL (structured query language). Since it stores complete database including all of its tables into a single file, it is suitable for implementing in systems that require zero configuration. Some of the limitations of SQLite include lack of support for writing to views, partial support for triggers and limited support for complex queries, for instance, it does not allow changing or removing columns using Alter Table statement [81].

- **HSQLDB:** HSQLDB stands for Hyper Structured Query Language Database. It is an open source RDBMS (relational database management system), which is developed in Java and available under a BSD (Berkeley Software Distribution) license [82]. Like SQLite, it also offers a light-weight database engine, which facilitates both disk-based and in-memory tables. Additionally, it supports both server-less as well as server-based modes of operation.

- **PERST/PERSTLITE:** Perst is an open source object-oriented database management system (ODBMS), officially released by McObject on February 6, 2006. It is available in two implementations: Java and C#. It is an embedded light-weight database system, which leverages the object oriented nature of Java and C#. It enables developers to store/retrieve objects directly and could be included inside an application that needs its own data storage. PerstLite is an implementation of Perst for J2ME [83].

- **BERKELEY DBXML:** Originally Sleepycat Software developed Berkeley DBXML, which was later on acquired by Oracle and now known as Oracle Berkeley DBXML. It is an embeddable open source XML database package specifically designed for storing and retrieving XML documents. Berkeley DBXML is written in C++, APIs for Berkeley DBXML exist for Java, C/C++, Python, TCL and Perl [84].

  Berkeley DBXML is extended from and built on top of Berkeley DB (BDB) [86]. BDB is originated from University of California, Berkeley, and is known as a "key-value" database package. It facilitates data storage in the form of arbitrary key-value pairs as byte arrays with the support of multiple data items for a single key. Since Berkeley DBXML is extended from BDB, it incorporates all potential features of BDB, such as zero-configuration and zero-human administration. However, unlike BDB, and relational databases that store data in relational tables, Berkeley DBXML is designed to store arbitrary trees of XML data. It provides efficient and fast data retrieval, because it has an XQuery engine, an XML indexer and a parser on top of BDB [87]. In Berkeley DBXML, documents are stored in containers either as a complete document or as nodes. The containers that store complete XML document without any changes or alterations are known as Wholedoc containers, while those containers that store XML document as nodes are known as node containers. The default type is node container.

In summary, Berkeley DBXML provides such features and benefits that make it an ideal package to use in the implementation of the proposed system. For example, in our proposed system the overall communication between several entities of the system is in XML form, even devices use XML documents to register their capabilities. The other benefits of Berkeley DBXML includes fast XML data storage and retrieval, support for W3C standard XQuery and XPath, eliminating the need for a DBA, and the capability for unattended and continuous operation (i.e. zero administration). Also from development and programming point of views, Berkely DBXML is flexible and easy to deploy, eliminate the need to convert XML into other data structures and supports a wide range of programming languages and operating system platforms [84].

4.1.1.4 PROTOCOL SELECTION (PS)

The protocol selection component provides APIs to find the best possible PoP protocol(s) based on the devices' profiles, user-preferences and a selection criterion defined in an XML-based PoP protocol specification and selection policy file, which is already described in chapter 3 (section 3.6).

4.1.1.5 SECURE ASSOCIATION INITIATION (SAI)

The Secure association initiation component utilizes the APIs from protocol selection, communication, security and proof-of-proximity components in order to find, negotiate and initiate the execution of the selected PoP protocol.

4.1.1.6 PROOF OF PROXIMITY (POP)

The Proof of proximity component is responsible for providing APIs for the execution of the selected PoP protocol in order to prove the physical proximity between devices.

4.1.1.7 SECURITY

Since the system needs to perform several cryptographic operations, such as encryption/decryption and calculating hashes, the security component facilitates with these cryptographic functions.

4.1.1.8 DEVICE PAIRING REPOSITORY (DPR)

DPR works in association with the secure association initiation and proof of proximity components. As the directory component in the co-location server facilitates with the device registry, the DPR component stores the information about successfully created pairings for future use. The core responsibility of the DPR component is to facilitate with storing, retrieving, modifying, updating and deleting the information relating to successfully established pairings.

4.1.2 STRUCTURE OF THE COLOC PROTOCOL MESSAGES

The DTD to validate the CoLoc protocol messages is shown in figure 4.3. This DTD describes the structure of CoLoc protocol messages and is used to validate all the XML-based communication in the system during registration, discovery and proof-of-proximity phases, such as device profiles, discovery queries, and co-location server's reply to client in response of discovery request.

Every Coloc protocol message contains a 'Command' tag and at most one 'XMLData' tag. The command tag defines the type of message or protocol instruction, while the XMLData tag defines several other sub-tags, but only one can be used in any given message. For example, XMLData tag contains a 'DeviceProfile' sub-tag to define the device profile during registration, and a 'Query' sub-tag to define the client's query during the discovery phase (refer to figure 3.10). 'DeviceList' tag defines the list of found devices as a result of client's query. The DeviceID tag is used when a resource device performs explicit de-registration with the co-location server or request for a renewal of registration or pairing session key. The PSK tag defines the pairing session key and is used during the registration or the renewal of registration of the resource device, while CP tag defines the credential password that is used in providing credential revocation mechanism (see section 3.7.2). PoPProtocol and PoPData are used during the proof of proximity phase, which define PoP protocol name and PoP data respectively. A CoLoc message that illustrates the device's explicit deregistration request is given in figure 4.4.

```
<!-- This DTD validates protocol messages/commands -->
<!ELEMENTProtocolMsg ( Command, XMLData? ) >
<!ELEMENT Command ( #PCDATA ) >
<!ELEMENT XMLData ( DeviceProfile | Query | DeviceList | DeviceID | PoPProtocol | PoPData| PSK| CP) >
<!ELEMENTDeviceProfile ( DeviceID, LDuration, Keyword, DeviceLocation?, CommProtocol+, DeviceCap, UserInput? ) >
<!ELEMENT Query ( #PCDATA ) >
<!ELEMENT DeviceList ( DeviceInfo+ ) >
<!ELEMENTDeviceInfo ( PSK, ExpiryTime, DeviceID, DeviceLocation?, CommProtocol+, DeviceCap, UserInput? ) >
<!ELEMENT ExpiryTime ( #PCDATA ) >
<!ELEMENT PSK ( #PCDATA ) >
<!ELEMENT DeviceID ( #PCDATA ) >
<!ELEMENT LDuration ( #PCDATA ) >
<!ELEMENT Keyword ( #PCDATA ) >
<!ELEMENT DeviceLocation ( #PCDATA ) >
<!ELEMENT CommProtocol ( ChannelName, Address ) >
<!ELEMENT ChannelName ( #PCDATA ) >
<!ELEMENT Address ( #PCDATA ) >
<!ELEMENT DeviceCap ( #PCDATA ) >
<!ELEMENT UserInput ( #PCDATA ) >
<!ELEMENT PoPProtocol ( #PCDATA ) >
<!ELEMENT PoPData ( #PCDATA ) >
<!ELEMENT CP ( #PCDATA ) >
```

Figure 4.3: DTD to validate the messages of CoLoc protocol

```xml
<?xml version="1.0" encoding="ISO-8859-1" ?>
<ProtocolMsg>
    <Command>DeRegister</Command>
    <XMLData>
        <DeviceID>Wiston</DeviceID>
    </XMLData>
</ProtocolMsg>
```

Figure 4.4: A Coloc protocol message illustrating the device's explicit deregistration request

4.2 Classification of the Implemented PoP Protocols

In this section, we present the details of the PoP protocols, which are integrated in the prototype implementation of the proposed system in order to demonstrate the physical proximity of the devices. The fourteen integrated PoP protocols are classified into three categories (referred as category-1, category-2 and category-3 protocols hereafter in this dissertation) distinguishing user action(s) in the generation of PoP data and user involvement in verifying/matching PoP data. Those PoP protocols, which require some user action(s) in the generation of PoP data belongs to the category-1, the category-2 is composed of those PoP protocols, which involve user in the verification/matching of PoP data, and the category-3 contains those PoP protocols, which require user to be involved in both the generation and the verification of PoP data. The category-wise brief description of each of the implemented PoP protocol is presented below.

4.2.1 Category-1 PoP Protocols

4.2.1.1 Button-to-Button (B-to-B)

This protocol is originally introduced by Soriente et al. in BEDA [12]. This scheme requires that both of the devices must have at least a single button. The main idea is that the user simultaneously presses a button on both of the devices. The time-interval between button presses is then utilized to generate 15-bits PoP data on each of the device. In our system implementation, the ENTER key of the keyboard is programmed to serve the purpose of a single button.

4.2.1.2 BLINK-TO-BUTTON (BLINK-TO-B)

This scheme is a variant of Display-to-Button (D-to-B), originally introduced in BEDA [12]. This scheme requires that at least one of the devices have a single button and the other an LED (light emitting diode). The main idea is that the user presses a button on the first device in synchronization with blinking pattern of an LED on the other device. Actually, the first device selects a short secret and encodes it into several time-intervals and transmits it through the LED-blinks by inserting these time-intervals between the blinking patterns. Then, on the other device, time-intervals between button presses is utilized to generate the same secret key.

4.2.1.3 BEEP-TO-BUTTON (BEEP-TO-B)

This scheme is a variant of the Blink-to-Button scheme. The only difference is that instead of an LED, one of the devices requires a speaker or beeper and the other a single button. In this scheme, the user presses a button on the first device in synchronization with beeps generated by the other device.

4.2.1.4 SEEING IS BELIEVING (SiB)

This scheme is originally introduced by McCune et al. [29] and the detailed description of this scheme has already been given in chapter 2. As in [88], we have not implemented this scheme in its entirety. Instead, for the sake of testing usability, we have followed the simplified method mentioned in [88]. The idea described in [88] is that the user takes photo of a bar-code that will be stored in the client device and later on manually processed by the test administrator.

4.2.2 CATEGORY-2 POP PROTOCOLS

As described earlier, in these schemes, the system generates PoP data internally through the use of cryptographic functions and then the user is responsible for verifying that both of the devices possess the same PoP data. For those schemes that require synchronization, we again follow the method of [88]. According to that method, synchronization is achieved with the help of the user by pressing a single button (i.e. the ENTER key in our case) on both of the devices simultaneously.

4.2.2.1 BLINK-BLINK

The verification process starts when the user presses an ENTER key on both of the devices simultaneously. Then, both of the devices encode the PoP data into blinking patterns and transmit it. The user is then responsible for observing the blinking patterns emitting from both of the devices and accepting or rejecting pairing based on his/her observation.

4.2.2.2 BLINK-BEEP

Blink-Beep is same as Blink-Blink except that one device encodes and transmits PoP Data into blinking patterns; while the other uses speakers/beeper to transmit beeps in synchronization with LED blinks on the first device. As in Blink-Blink, the user is then responsible for observing the blinking patterns emitting from one device and beeps from the other device, and accepting or rejecting pairing based on his/her observation.

4.2.2.3 BEEP-BEEP

Again it is same as Blink-Blink. Here the only difference is that a speaker/beeper is required in both of the devices and PoP data is encoded and transmitted through the beep sounds. The user is again responsible for observing that the beep sounds are generated in synchronization from both the devices, and accepting or rejecting pairing based on his/her observation.

4.2.2.4 DISPLAY-DISPLAY

This scheme is originally introduced by Goodrich et al. in Loud and Clear [31]. We have used the same dictionary to generate the MadLib sentences (i.e. sentences that are non-sensical, but syntactically or grammatically correct) used in the variant we have implemented. The main idea of this scheme is that it encodes PoP data into a MadLib sentence and displays it on the screen of both of the devices. Then the user is responsible for comparing the two MadLib sentences shown on the screens of both of the devices, and accepting or rejecting the pairing based on his/her observation.

4.2.2.5 DISPLAY-SPEAKER

In this variant PoP data is encoded into a MadLib sentence. One of the devices shows it on its screen; while the other vocalizes it. Then the user is responsible for comparing the vocalized MadLib sentence with the one shown on the screens of other device, and accepting or rejecting the pairing based on his/her observation.

4.2.2.6 SPEAKER-SPEAKER

This is the same as the Display-Display, the only difference is that both of the devices require speakers. Thus, instead of showing Madlib sentences on screen, both of the devices vocalize them. Then the user is responsible for comparing the vocalized Madlib sentences from both of the devices, and accepting or rejecting pairing based on his/her observation.

4.2.2.7 DIGITS COMPARISON

It is very similar to Display-Display scheme, the only difference is that instead of comparing Madlib sentences, a number is compared, which is shown on the screens of both of the devices. The user is then responsible for accepting or rejecting pairing based on his/her observation. Unlike Display-Display, this scheme is more suitable for devices that have limited/small displays.

4.2.2.8 Hash Comparison

It is similar to Display-Display scheme. However instead of encoding PoP data into a MadLib sentence, it is encoded into a hash value that is displayed on the screens of both of the devices. Then the user is responsible for comparing the two hashes shown on the screens of both of the devices and accepting or rejecting the pairing based on his/her observation.

4.2.3 Category-3 PoP Protocols

4.2.3.1 Selective Image Comparison (SiC)

In this scheme, the user selects an image that is transmitted as PoP data and shown on the screen of both of the devices. The user is then responsible for comparing the image shown on the screens of both of the devices, and accepting or rejecting the pairing based on his/her observation.

4.2.3.2 Capture and Show (CaS)

In this method, the user is fully involved in the pairing process. This method requires that at least one of the devices has a Camera. The user takes a snap-shot of something, such as any object or scene, near to him. The system shows the captured image on the screens of both of the devices. Then the user is responsible for accepting or rejecting the pairing based on his/her observation.

4.2.4 Automatic PoP Protocols

Table 4.1 presents the summary of the features of the implemented PoP protocols. However, apart from the previously discussed three categories of PoP protocols, we also envision an additional category of automatic PoP protocols. The automatic PoP protocols might not involve the user in the pairing process at all. These schemes will rely heavily on sensor technologies. In the context of our proposed system, these schemes will generate the PoP data from automatic, accurate and reliable sensors, which will also verified automatically without involving users. Others have already started to explore these schemes. Although, they haven't developed fully automatic protocols yet, their proposed schemes require much less user interaction/involvement during the pairing process as compared to the previously discussed three categories of protocols. Examples of these schemes include [11, 14, 16, 28], however these schemes are costly, and require exotic hardware and/or common interface on both of the devices, which is impractical in most of the device pairing scenarios in ubiquitous computing environments.

The Implemented PoP Protocol	Device Capabilities		Human/User involvement	Out-of-band/Location-limited secondary channel
	Device A	Device B		
Category-1: Users are involved in generating PoP data.				
Button-to-Button	A single button on each of the device		Press button on both of the devices simultaneously with random time-intervals until response signal received	Tactile
Blink-to-Button	An LED	A single button	Press and release button on device B whenever the LED of device A flashes/blinks	Visual and Tactile
Beep-to-Button	Speaker/Beeper	A single button	Press and release button on device B whenever the LED of device A flashes/blinks	Audio and Tactile
Seeing-is-Believing	Display or Bar code sticker/label	Camera	Properly place camera of device B at the displayed bar code on device A with sufficient proximity and take the photograph	Visual
Category-2: Users are involved in verifying the PoP data.				
Blink-Blink	A single LED on each of the device		Compare the two synchronous LED blinking patterns	Visual
Blink-Beep	An LED	Speaker	Observe the synchronization of an LED blink with a beep generated from the other device	Audio and Visual
Beep-Beep	Speaker	Speaker	Compare the two vocalized MadLib sentences from both of the devices	Audio
Display-Display	Display	Display	Compare the two MadLib sentences displayed on both of the devices	Visual
Display-Speaker	Display	Speaker	Compare the MadLib sentence displayed on the screen of device A with the vocalized MadLib sentence from device B	Audio and Visual
Speaker-Speaker	Speaker	Speaker	Compare the two vocalized MadLib sentences from both of the devices	Audio
Digits Comparison	Display	Display	Compare two numbers displayed on both of the devices	Visual
Hash Comparison	Display	Display	Compare two hashes displayed on both of the devices	Visual
Category-3: Users are involved in both generating and verifying the PoP data.				
Selective Image Comparison	Display	Display	Compare two images displayed on both of the devices	Visual
Capture and Show	Display	Display and Camera	Take a photo of a nearby object/scene and compare it with the image displayed on the screens of both of the devices	Visual

Table 4.1: Features summary of the PoP Protocols

4.3 DEMONSTRATION OF THE IMPLEMENTATION

Figure 4.5(a) and 4.5(b) shows the screen shot of the user interface of the client and resource applications. We have designed simple user interfaces. In this section, we demonstrate the execution of the proposed system through the help of several screen shots.

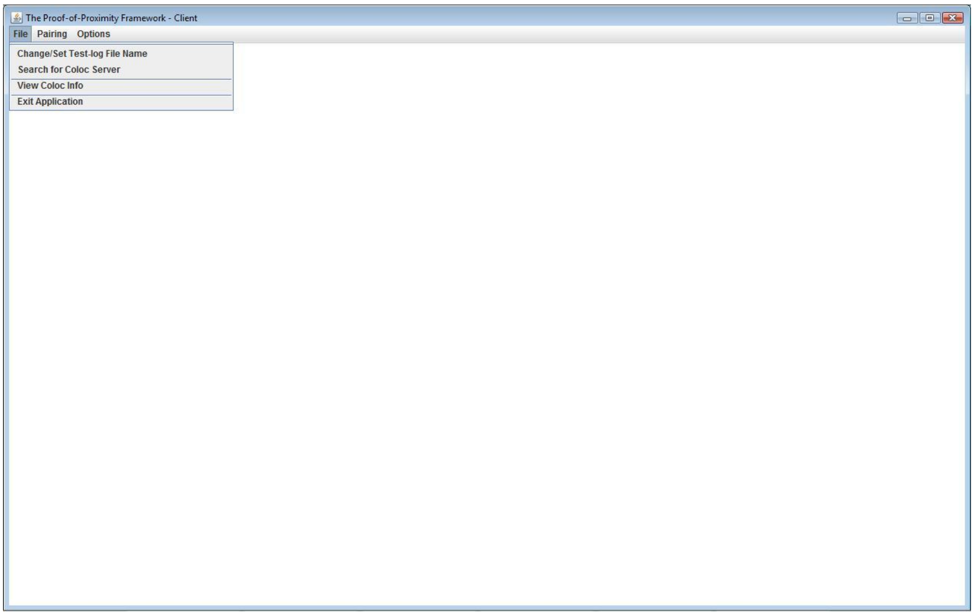

Figure 4.5(a): Client applications' GUI

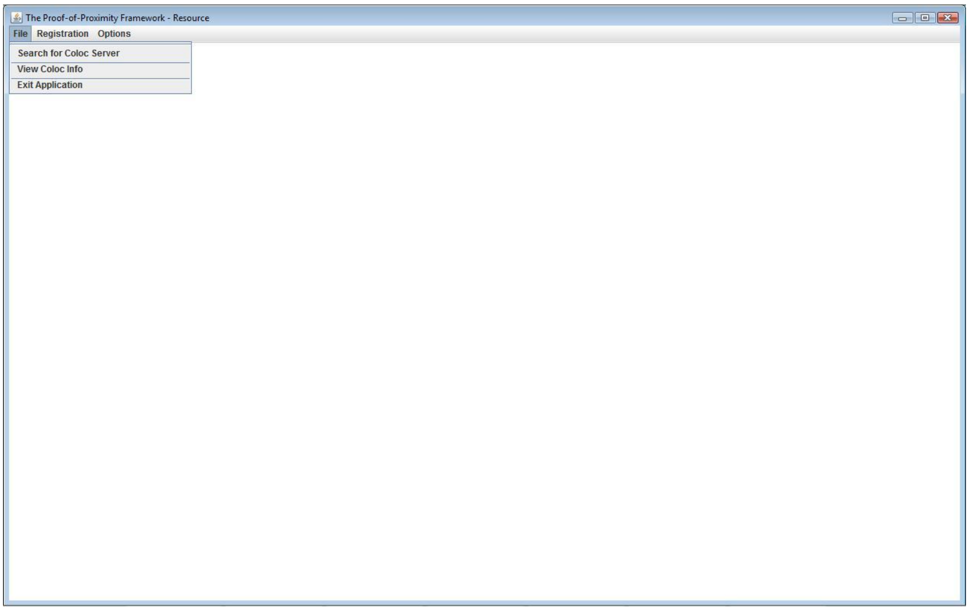

Figure 4.5(b): Resource applications' GUI

Once the co-location server bootstrap the system through broadcasting its public key and connectivity information, the user can find it by clicking the ‘Search for Coloc Server’ menu item from the client or resource application. In order to register several devices in the system, we have simulated different kinds of devices (i.e. printers, laptops, desktop computers, mobile phones) through creating their XML-based device profiles and stored them locally. Then each device is registered with the co-location server just by clicking the ‘Device Registration’ menu item (figure 4.6) and then providing its XML-based device profile.

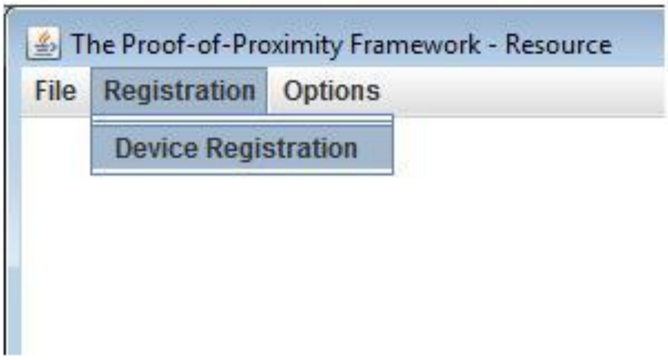

Figure 4.6: Screenshot of resource application illustrating the device registration step

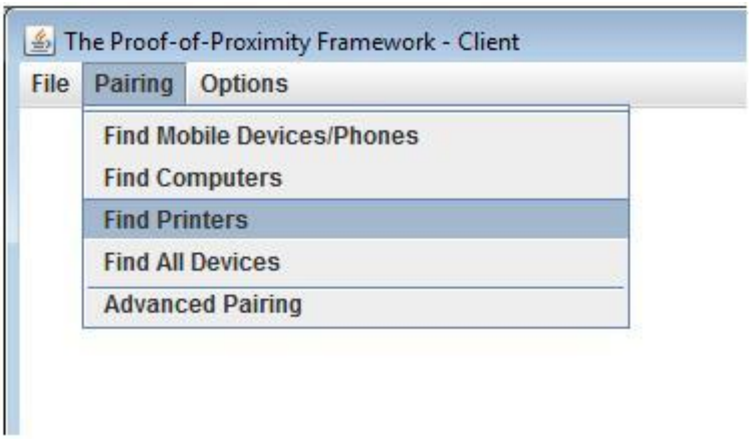

Figure 4.7: Screenshot of client application illustrating the device discovery step

Once devices are registered with the co-location server, these can be discovered by the client application. A user can use any one of the pre-set discovery options (figure 4.7) or can use the ‘Advanced Pairing’ menu item to perform an advanced device discovery and pairing (figure 4.8). In advanced pairing a user can establish a long-term pairing, optionally enter the user input/preference, location in which devices required to be searched and the approximate distance of the intended pair-able devices. The distance value entered by the user facilitates the PoP protocol selection process with filtration of those PoP protocols that does suit the required distance. During this process the user-entered distance will be compared with the distance as given in protocol specification and selection policy file for each of the PoP protocol. Also note that this is an approximate distance.

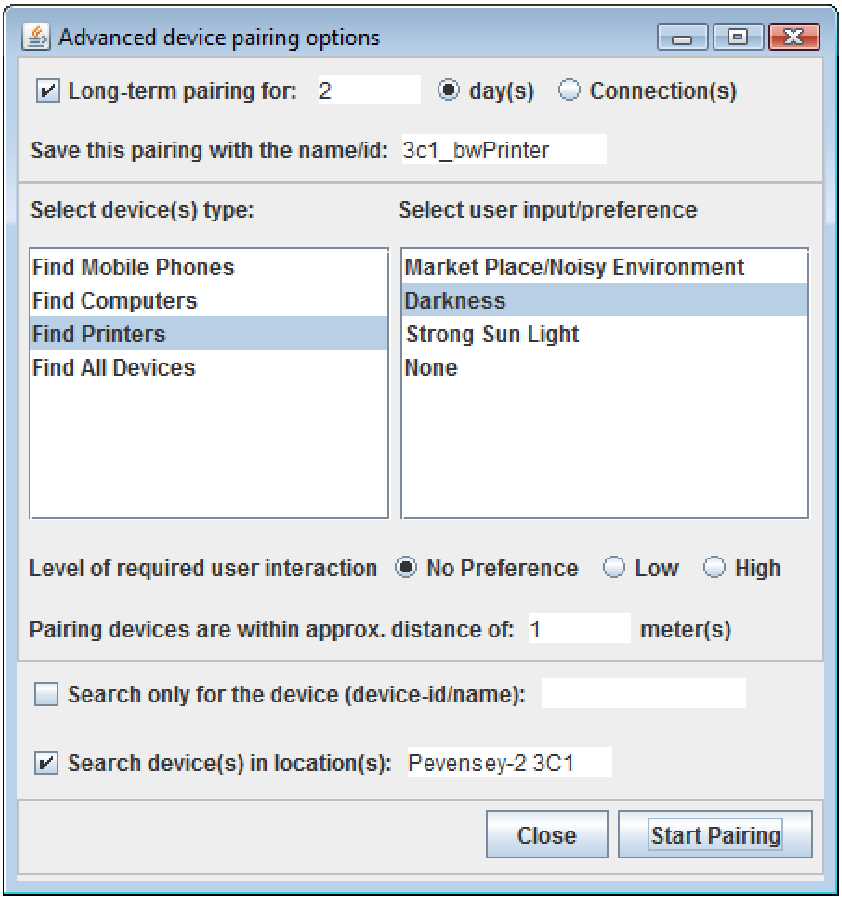

Figure 4.8: Advanced pairing options

Based on the user's selection of the device-type (in this case printer), the client receives a list of the matching devices as illustrated in figure 4.9. User selects the intended device and clicks the 'Next' button to proceed. When user clicks the next button, another list box appears on the screen containing the list of PoP protocols in recommended order (figure 4.10). Finally, the user selects the name of a PoP protocol and clicks the 'Do Pairing', which initiates the process of demonstrating the physical proximity of devices through the chosen PoP protocol.

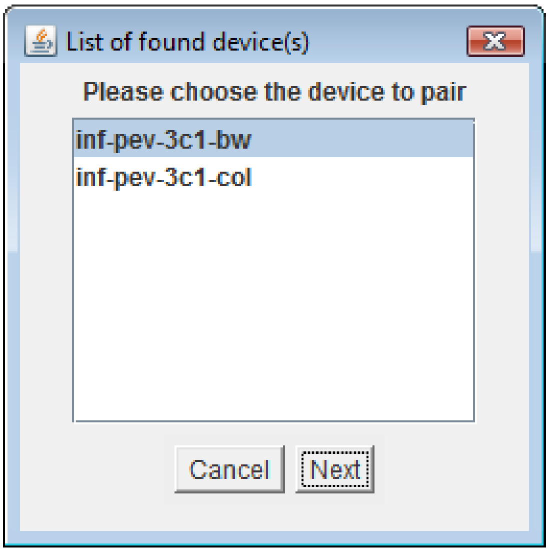

Figure 4.9: Screenshot showing list of found devices

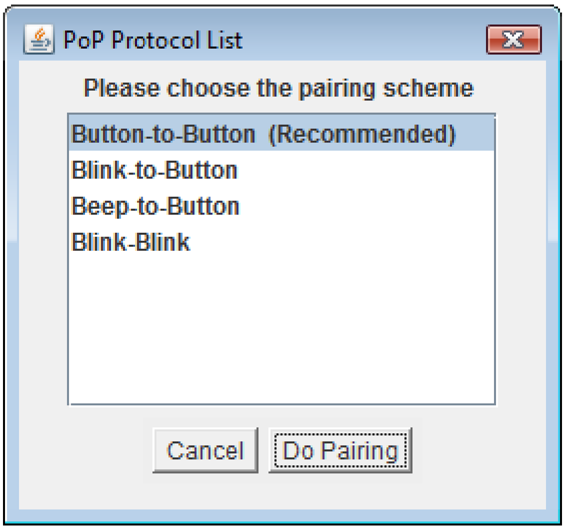

Figure 4.10: Screenshot showing list of PoP protocols

4.4 SUMMARY

The focus of this chapter was the prototype implementation of the framework. We described the software components of the proposed system and discussed the structure of the CoLoc protocol messages. We also presented details of the fourteen PoP protocols that are integrated in the prototype implementation of the proposed system. The implemented PoP protocols are classified into three categories distinguishing user actions in the generation of PoP data and user involvement in verifying/matching PoP data. Apart from the three categories of implemented PoP protocols, we also envisioned an additional category of automatic PoP protocols. The protocols belonging to the automatic category of PoP protocols might not involve the user in the pairing process, instead these might rely heavily on sensor technologies; however these schemes are costly, and require exotic hardware and/or common interface on both of the devices that make them impractical in most of the device pairing scenarios in ubiquitous computing environments. Finally, we presented the demonstration of the prototype implementation. In next chapter, we provide the details of a usability study that is carried out in pursuance of the hypothesis of the dissertation and to evaluate the proposed system.

CHAPTER 5

EVALUATION

The focus of this chapter is a usability study that is carried out in pursuance of the hypothesis of the dissertation and to evaluate the proposed system. The evaluation of the proposed system is presented through the analysis of the usability study results and its design features. At the end of this chapter, we also discuss the wider view of the usability study results in general and their impact on the PoP protocol selection criteria in particular followed by the chapter's summary.

5.1 USABILITY STUDY

In order to evaluate the proposed system and to support our main argument that the integration of discovery mechanism and several proof-of-proximity protocols into a single device pairing system is an effective approach for ordinary users, we conducted a usability study. This is a study of the eight pairing schemes as well as the proposed system, which integrates them. The results of the usability study are useful to test three hypothesis: 1) are the users good at identifying an appropriate (right) pairing scheme when they have to choose between large number of pairing schemes; 2) To what extent users like to be involved in the pairing process; and most importantly to evaluate the main hypothesis of the dissertation that 3) is the integration of discovery mechanism and the pairing schemes into a single system an effective solution for ubiquitous computing environments from the user's point of view, and are they perceiving it as usable. Additionally, the results of the usability study also highlight some of the general aspects of device pairing schemes that are worth consideration when designing future solutions for device pairing. In this section, firstly we discuss the prior work on usability of device pairing schemes, then we describe the test apparatus and the test cases that are selected as part of this study followed by describing the demographic information of test's participants, test procedure, and the results.

5.1.1 Prior Work on Usability of Device Pairing Schemes

More recently the usability issue of secure device pairing schemes has got significant attention from researchers and there exist some recent work on the usability of device pairing schemes in the literature. Below are described some of the notable work in this area.

In the literature, Uzun et al. [89] are considered to be the first who performed the usability analysis of secure device pairing methods followed by [90, 91]. Uzun et al. [89] presented a comparative usability analysis of some of the conventional paring schemes. In their study, the participants were asked to compare strings displayed on mobile devices, copy a PIN displayed on one device and enter it onto another, and select a PIN from among 4 numeric values that matched a string displayed on another device. Their findings were that participant perceived copying and entering as booth secure and professional while comparing was perceived as easy to use. They recommended using a PIN of not more than 7 digits and that the user interface should be designed in such a way that the default option is the most secure. More recently, Kumar et al. [91] presented an experimental evaluation of a large set of device pairing schemes. Their [91] results showed that some simple schemes, such as number comparison, were quite attractive overall in terms of speed, security and usability. Subsequently, in [92] authors argued that the participants of prior study [91] comprised of mostly young males (70%) and the test organizers were experts in security relevant research as well as developers of some of the tested pairing schemes. They argued that the results of the study [91] were valuable, however it required further experimentation (usability tests) with more diverse participants, and more diverse scenarios. Many of the tested pairing schemes in [92] overlapped with the already tested schemes in [91], however this study differed from [91] in that the focus of this study was on within subjects analysis. The results of the study [92] were helpful in indentifying the pairing schemes, which were not feasible for some specific groups of users with regard to age, gender and prior experience with device pairing. More recently, Kainda et al. [93] also performed a usability and security evaluation of the pairing schemes. The main focus of this [93] work was on comparison of the usability and the security of those pairing schemes, which used more recently proposed and identified out-of-band channels together with some of the conventional ones as presented in [89]. The four classes of pairing schemes that were covered in this study are: Comparing (compare and confirm), Selecting (compare and select), Entering (copy and enter), and Barcode (taking a picture of a barcode using a camera). This work differed from [89] in the sense that authors also took into account the scenarios where the compared strings were nearly similar (i.e. mismatched by only one or two digits, characters or words depending on the scheme). Our work is similar to [93] in terms of the methodology used to carry out the usability study, however the selected pairing schemes in our study are different from those tested in [93].

5.1.2 TEST APPARATUS

We have setup the implemented system into two 1.9GHz Dell Machines each with 1GB RAM running the WindowsXP operating system. We have also used two PhidgetInterfaceKits [80]. We have used these kits to implement the blinking operation (i.e. LED blinking patterns), which is a requirement for two of the PoP protocols. Additionally, two of the implemented PoP protocols require a camera to capture the photo of barcode or some object/scene in the proximity. Since, there is not a built-in camera with the PCs; we have used an external web-cam for the purpose of capturing photos.

5.1.3 Selection of Test Cases

We have selected eight PoP schemes for our experiment: three from the first category (section 4.2.1), three from the second category (section 4.2.2), and two from the third category (section 4.2.3).

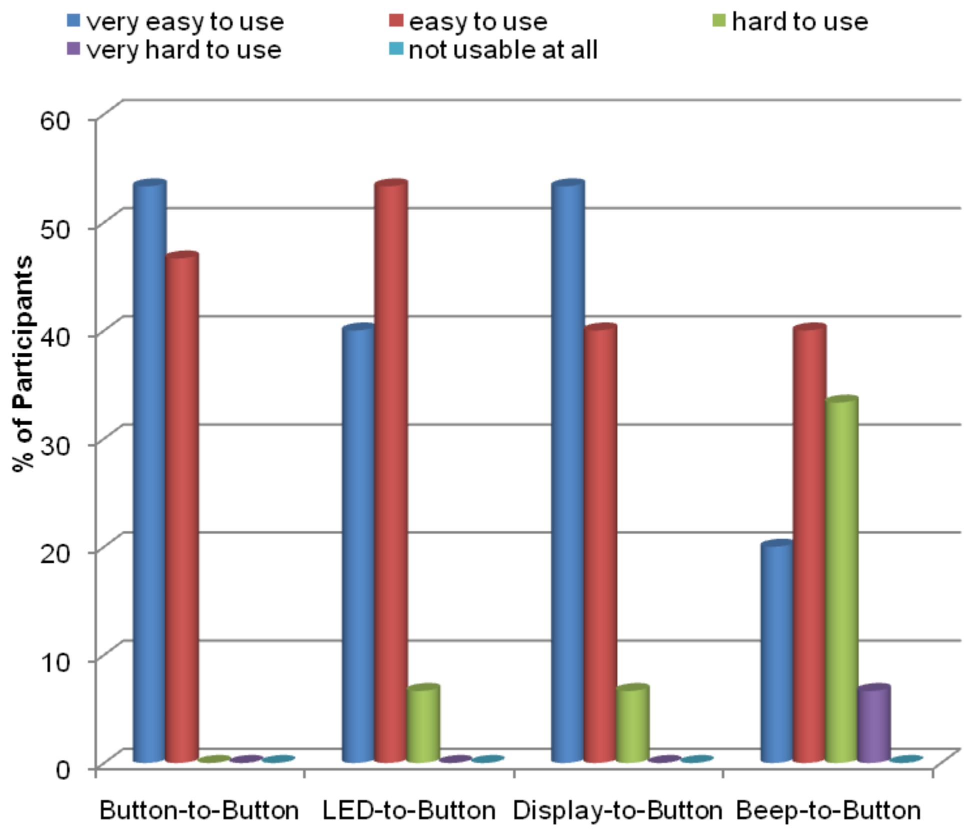

Figure 5.1: Participants response for the usability of 4-button based pairing schemes

The reason for conducting the user study with a reduced number of PoP schemes rather than all fourteen implemented schemes is to avoid user fatigue. With all PoP schemes, a single experiment takes around an hour, which causes for unrealistic/unproductive data, especially for a few of the last test cases/tasks.

Therefore after a careful analysis, 6 pairing schemes are eliminated from this usability study. During elimination process, we considered the results of some of our previous experiments (3, section 1.3) conducted with 15 users and 4 button-based schemes, and also referred the prior work [89-93] on usability of device pairing schemes. For example some of those schemes, which produce/require synchronized audio/visual signal did not perform well in prior evaluations due to high error rate and user-annoyance [9, 88], so we eliminated Beep-Beep and Speaker-Speaker and Blink-Beep. According to the results of our previous experiments (as shown in figure 5.1), the users perceived Beep-to-Button scheme as harder compared to the other three button-based schemes, so we eliminated the Beep-to-Button scheme as well. Digits comparison is too simple approach and hash comparison is not such a user-friendly approach [93], so we preferred Display-Display over these two schemes.

In summary, the following are the short-listed PoP protocols that we have selected for the usability study.

- **Category - 1**

  Button-to-Button

  Blink-to-Button

  Seeing-is-Believing (SiB)

- **Category - 2**

  Blink-Blink

  Display-Display

  Display-Speaker

- **Category - 3**

  Selective Image Comparison (SiC)

  Capture and Show (CaS)

5.1.4 PARTICIPANTS

It is widely accepted that any user study that is performed by 20 users captures over 98% of usability issues [94], so a total of 20 volunteers were recruited. The majority of the participants are students of the University of Sussex and most of them are proficient computer users. The background profile information of the participants is summarized in table 5.1.

Gender	
Male	55%
Female	45%
Age	
18 - 25	40%
26 - 40	40%
41 or above	20%
Education	
High School/College	15%
Bachelor	40%
Masters	35%
Doctorate (PhD)	10%
Pairing Experience	
Yes	90%
No	10%
Daily Computer Usage	
2 or less hours	15%
3 - 5 hours	50%
6 or above hours	35%

Table 5.1: Test participant's demographic information

5.1.5 TEST PROCEDURE

The tests were conducted in a lab-based environment. Before the start of each experiment, we explained briefly the goals of the experiment along with the description of each pairing method to the participant; however we had already provided a leaflet to each participant in either hardcopy or through email before the actual day of the experiment that contains all the details of the experiment. Each participant filled a pre-test questionnaire before starting the test cases. The pre-test questionnaire was used to collect the demographic information of the participants.

Each experiment consisted of two parts. In the first part, each participant performed the tasks of executing the eight PoP protocols, which are mentioned earlier in section 5.1.3. These eight protocols were programmed to work independently from the proposed system and do not include device registration and discovery phase. Every participant performed each of the tasks twice. The first execution of each of the tasks was without any attack, while the second execution was under an attack scenario, in which users had to identify the mismatches. From the data of the first execution of each user, safe errors (i.e. identifying a match as a mismatch) are identified, while second execution provides data for fatal errors (i.e. identifying a mismatch as a match). Note that for the pairing schemes in which user is involved in generating the PoP data, fatal errors are not applicable. Thus, in that case both of the executions were performed without the attack scenario. Timing information was also recorded and stored in the test log file along with the other data. At the end of first part, each participant was given an After Scenario Questionnaire-1 (ASQ-1) to record his/her satisfaction with the performed tasks.

In the second part of the experiment, each participant performed two executions of the proposed implemented system, which is described in chapter 4 (section 4.3). At the completion of this part of experiment, each participant is given an After Scenario Questionnaire-2 (ASQ-2) to record his/her satisfaction with the proposed implemented system, which is denoted as CoLoc in the results of the usability study.

Finally, at the end of overall experiment every participant also filled a post-test questionnaire that contains two scenario-based questions and one question regarding the ranking of each category of pairing schemes. The pre-test questionnaire, ASQ-1, ASQ-2, and post-test questionnaire are given in appendix B.1 – B.4.

5.1.6 RESULTS

The usability study results are obtained from the collected data by means of questionnaires (i.e. two ASQs and one Post-Test questionnaire) as well as by the log files generated during the experiment. Two separate log files were created for each participant during the experiment; one for first phase of the experiment and the other for second phase of the experiment. The first log file recorded 16 lines of data and each line contained 7 data items. These include test date and time, pairing scheme name, completion duration in seconds, expected completion result, actual completion result, error information, and information about the successful completion of task. There are 20 participants, so we got 2240 data items in total from first set of log files. The second log file recorded 2 lines of data and each line contained 8 data items, so we got 320 data items in total from the second set of log files. The seven data items are similar as in first log file and the eighth data item records information about the user input/preference. Further, we got 35 data items from the three questionnaires for each participant, thus we got total of 700 data items for 20 participants. Overall we got 3260 data items for analysis from questionnaires and log files. All of the data was transferred and recorded into Microsoft Excel workbooks for analysis and evaluation.

5.2 EVALUATION

In this section, we present the evaluation of the proposed system through the analysis of the system's design features and the results obtained from the usability study. In the view of our previously defined goals (chapter 3) and objectives of this research, we consider the three major metrics for evaluating the proposed system. These are usability, security, and generality. Usability evaluation will provide an assurance that the system is easy to use for the users and they are satisfied with the way system works. Security evaluation will make sure that the objective of securing communication between several entities of the system is achieved, along with providing confirmation of the physical proximity of the devices involved in the pairing process. Generality evaluation will ensure that the system is applicable in a large set of device pairing scenarios in ubiquitous computing environments, capable of incorporating existing pairing schemes, and can be extended without substantial effort. Apart from these main evaluations, we have also presented the performance analysis, combined metrics analysis of the eight PoP protocols and the proposed system, and the ranking analysis for the categories of PoP protocols.

5.2.1 USABILITY EVALUATION

The data obtained from both of the ASQs and post-test questionnaires revealed the participant's opinion of each of the test cases and their capability to perceive an appropriate pairing scheme for a given device pairing scenario. The participant's opinion is expressed in terms of rating scores on a scale of 1 to 7 in which 1 is representing the lowest score and 7 is representing the highest or the most satisfactory score. The selection of seven-step scale is based on the fact that it captures proper balance between reliability of scale and discriminative demand on the participants [95-97].

The graphs shown in figures 5.2 and 5.3 are drawn from the data obtained from ASQ-1 and ASQ-2. Every participant recorded their satisfaction opinion for each of the test case by giving a score (i.e. 1-7) to each of the three measures on the ASQs. The graph in figure 5.2 shows the participants' rating view for each of the three measures. However in order to calculate the single score and to present the overall satisfaction of the participants for each of the test case, these scores are averaged and presented in figure 5.3. The results show that Button-to-Button pairing scheme is on top with the users average satisfaction score of 6.216. Display-Display and SiB has an average score of 6.1 and 6.15 respectively followed by CoLoc and CaS with the average satisfaction score of 5.616 and 5.556 respectively. Display-Speaker has the lowest average satisfaction score of 4.85, while Blink-to-Button and Blink-Blink stands with an average satisfaction score of 5.416 and 5.106 respectively.

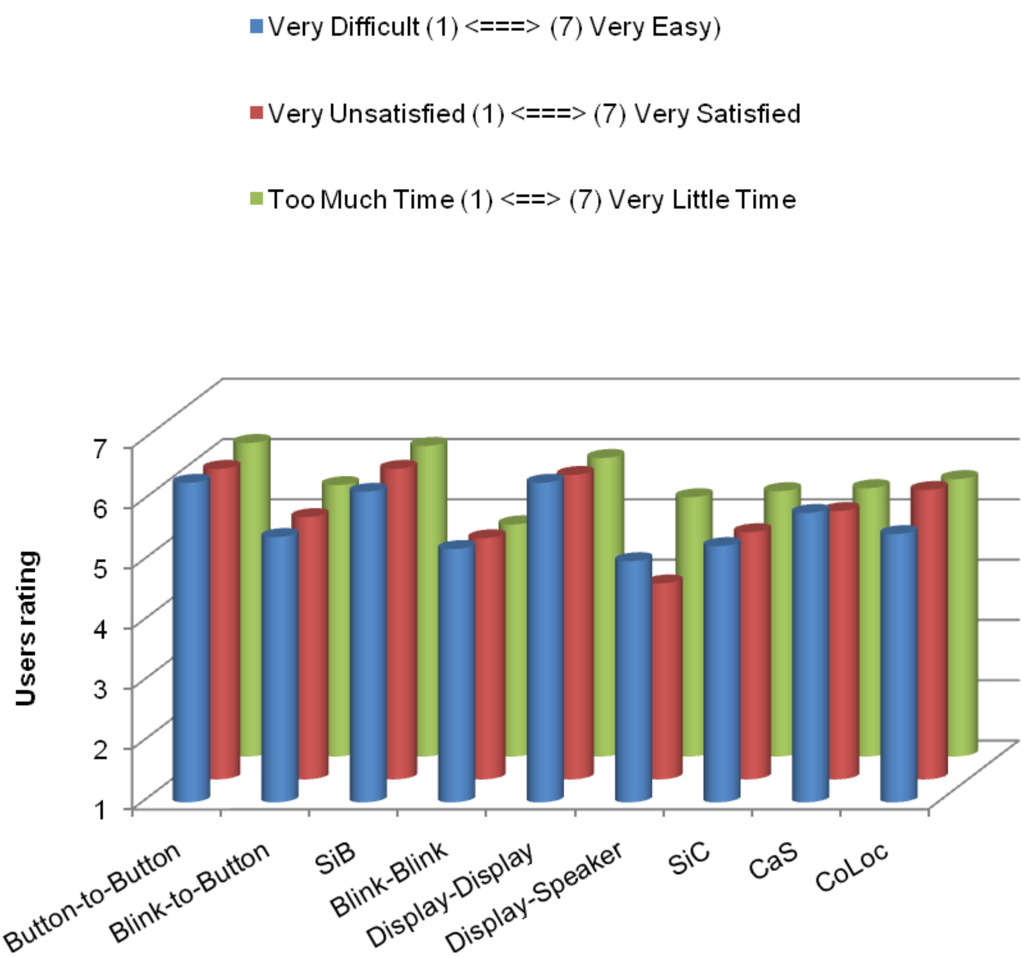

Figure 5.2: Users average rating score on a 7-step scale for the three measures of user's satisfaction

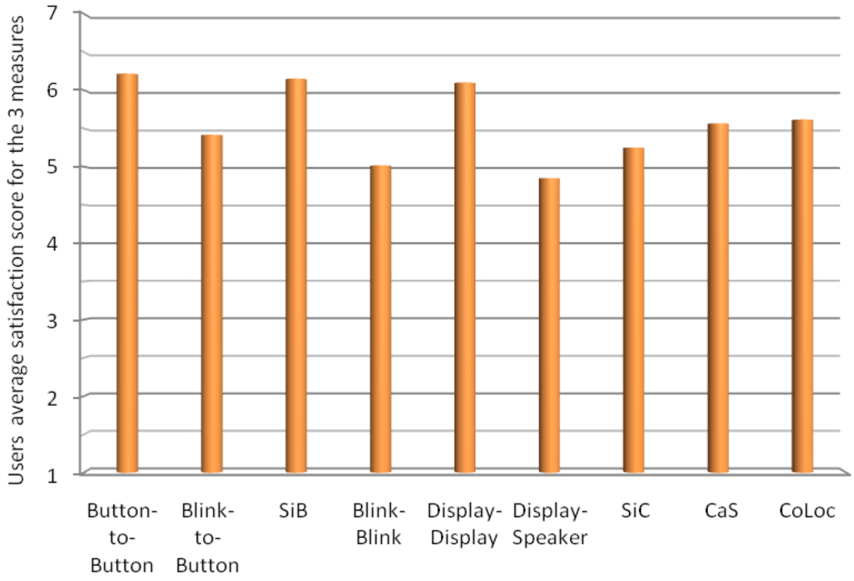

Figure 5.3: Users average satisfaction score on a 7-step scale for three measures

It is well known in the literature of usability evaluations that an average score of 5.6 on a 7-step scale is considered to be satisfactory and acceptable for a system or product, while an average score of 4 is the acceptable score on a 5-step scale [98]. CoLoc has an average satisfaction score of 5.616 for the three measures of usability, which indicates that the proposed system is usable and practically feasible for its users.

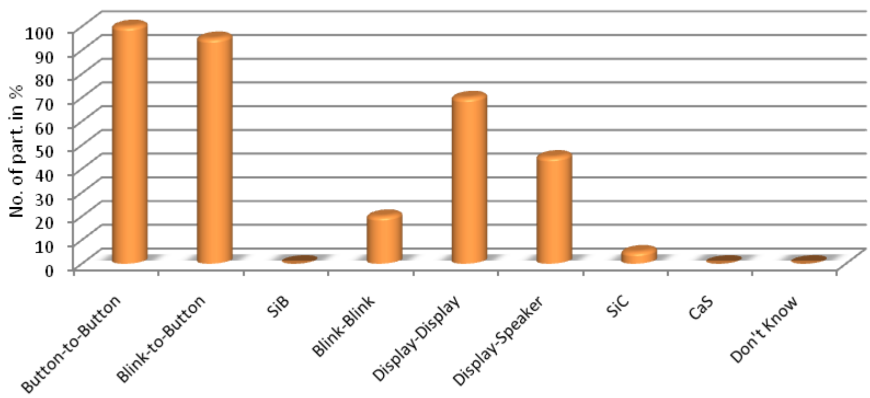

Figure 5.4: Participants response to question 2 (see appendix B.2) of the post-test questionnaire

The graph in figure 5.4 is drawn from the data collected as response to a scenario-based question. A scenario is presented to the participants on post-test questionnaire (see Q2 in appendix B.2) with a number of options and asked to select all of the possible pairing schemes. The correct response was Button-to-Button and Blink-to-Button. However, results in figure 5.4 show that many participants have selected the wrong pairing schemes as well along with the correct ones.

Graphs in figure 5.5 and figure 5.6 are drawn from the data collected as response to another scenario-based question on post test questionnaire (see Q3 in appendix B.2). The scenario is presented to the participants with a smaller number of options and asked to select one of the best possible pairing schemes. The correct response is Button-to-Button. Results show that all of the participants (100%) selected Button-to-Button scheme, however 5% selected the Display-Speaker and 10% selected the Blink-to-Button scheme along with the Button-to-Button scheme. Considering the fact that Button-to-Button is the correct choice, it can be concluded that 85% of the participants have selected the right choice, while 15% of the participants have selected nearly correct response, but none has selected a totally wrong choice.

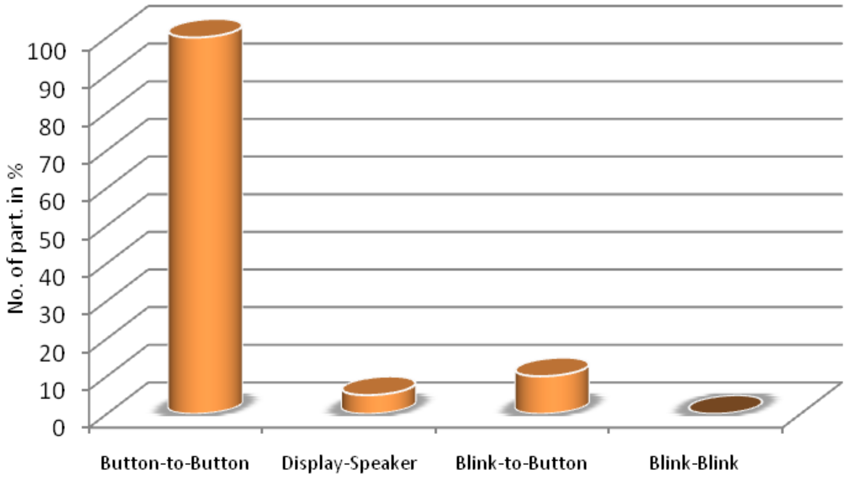

Figure 5.5: Participants response to question 3 of the post-test questionnaire

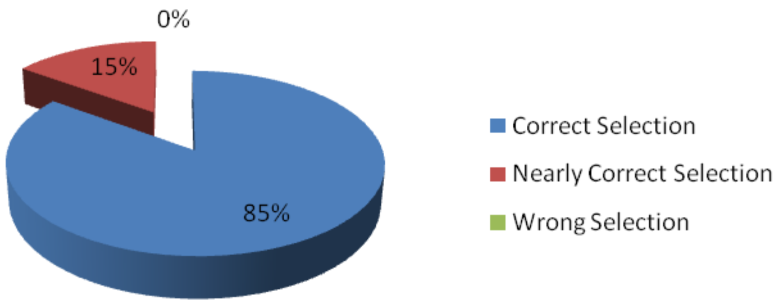

Figure 5.6: Interpreted results for response to question 3 of the post-test questionnaire

The results presented in figure 5.4 reveal the fact that users are not good at identifying which pairing schemes are applicable in which scenarios. However, when users are given short listed pairing schemes, they performed well at identifying the suitable pairing schemes (figures 5.5 and 5.6). These results support our argument related to usability that ordinary users are not good at identifying appropriate schemes in a situation when they have to choose between many different pairing schemes; however if the cognitive overhead in terms of deciding/thinking an appropriate pairing scheme could be reduced, they are capable of performing very well in the pairing process. This result clearly supports our hypothesis that assistance in choosing a pairing scheme has value.

5.2.2 Security Evaluation

As stated earlier, the objective of security evaluation is to ensure that the proposed system is integrating the PoP protocols well and also securing the overall communication between several entities of the system. The security of device pairing schemes, where users are involved in security-related interactions, is evaluated in terms of safe errors and fatal errors [99]. Safe error denotes the systems inability to pair two legitimate co-located devices due to system error or user error in case of use of out-of-band channels. User errors are due to either very complicated steps of pairing, unclear instructions for the user to what to follow to achieve successful pairing or user's own carelessness during the pairing process. Fatal error denotes the systems inability to prevent pairing of an adversary with a legitimate device of the system. Note that fatal errors are more dangerous and cause more serious consequences compared to safe errors. Fatal errors are not applicable in most of the schemes that involve users in only generating PoP data. In the case of our system, fatal errors are not applicable to button-based schemes and SiB. Since CoLoc incorporates these schemes and also it encrypts all the communication between the communicating partners, fatal errors are also not applicable to it.

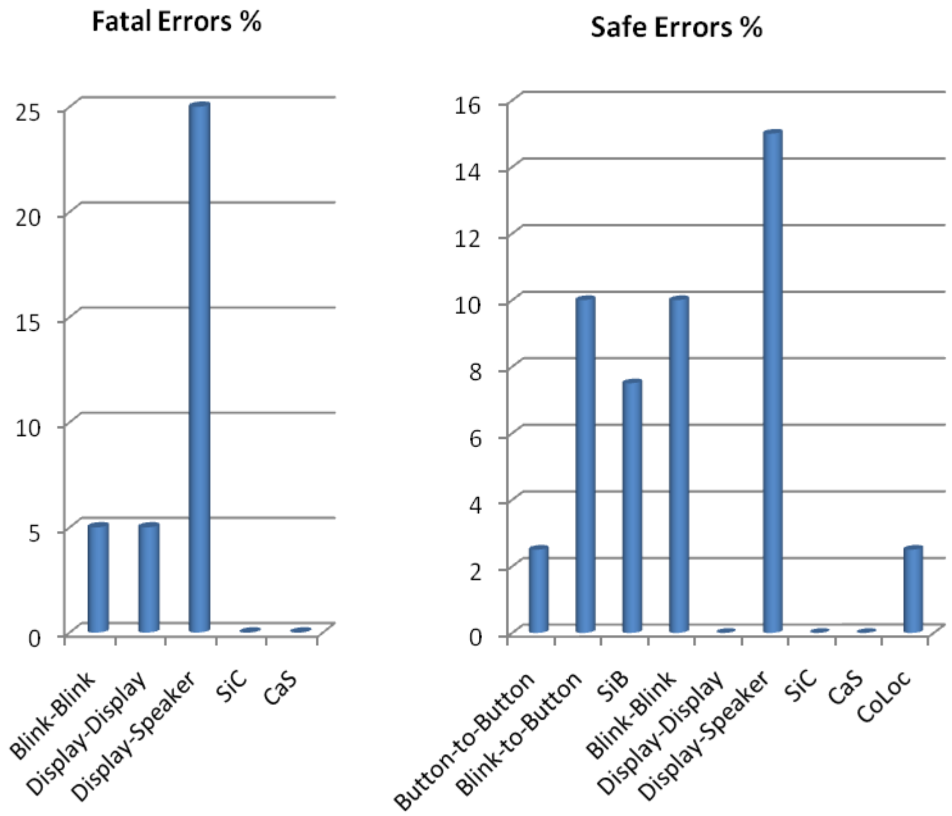

Figure 5.7: Safe and fatal errors for each of the test case

When looking at the safe errors in figure 5.7, Display-Speaker has the largest safe error rate, while Display-Display, Selective Image Comparison, and Capture and Show have not even a single safe error. Button-to-Button and CoLoc have lower error rates as compared to the other schemes. CoLoc has an average error rate of only 2.5%.

When we performed a more detailed analysis of these errors, it comes to our notice that these 2.5% safe errors occurred when the participant selected Blink-to-Button as the PoP protocol during the execution of the proof-of-proximity phase, and the safe error rate of Blink-to-Button is already high in comparison to the other schemes, excluding Display-Speaker. This indicates that the rate of safe errors for CoLoc is somehow dependent on the selection of PoP protocol. These results indicate that the proposed system achieves its first security goal (i.e. demonstrating physical proximity of devices) very well.

The second security goal is to make sure that the communication between several entities of the system is secure. We have achieved this goal through encrypting all the communication from resource registration until the end of the execution of the proof-of-proximity phase. The encrypted and integrity protected mode of communication used during the resource registration and discovery phase protect the pairing process from the bidding-down-attack. As stated earlier in chapter 2, in this kind of attack, the goal of an adversary is to fool (bid-down) the intended pair-able devices to use weaker security than is possible. For instance, when pairing two display and camera-equipped devices, an adversary could modify the capabilities of one of the devices into a display-less and/or camera-less device (bidding-down) to force a radio-based pairing protocol to be used, which is easier to intercept without being detected. Additionally, when the proposed system is implemented considering the assumptions provided in chapter 3, it is also secure against MiTM attack. These facts indicate that beside the usability, the proposed system also achieves its security goals.

5.2.3 GENERALITY EVALUATION

The purpose of generality evaluation is to make sure that the system is capable of incorporating existing pairing schemes, as well as being extendable without substantial modifications in the design, and being applicable in a large set of device pairing scenarios in ubiquitous computing environments. Towards this, we have already shown in the previous chapter (implementation) that CoLoc is capable of integrating several pairing schemes (known as PoP protocols in this dissertation) to authenticate the physical proximity of the devices. Further, in addition to establishing a secure session between two previously unassociated devices, the proposed system is also capable of establishing secure group communication, creating and managing long-term pairings, and also offers a mechanism for the selection of PoP protocol that gives some control to the user. Moreover, the system is designed in a way that it can be extended without substantial effort. We are defining extension from two different points of view: the developers/programmers' point of view and the deployment point of view.

From the developers' point of view, they can add a new PoP protocol to the system by performing following steps:

- Firstly, they are required to include specifications for the new PoP protocol in the XML-based policy file.
- Secondly, they are required to write the PoP protocol implementation code in Java, which needs to be included into proof-of-proximity software component.

From a deployment point of view, the proposed system is capable of being deployed to multiple servers; thus facilitating the secure association of a pair of devices, each of which belongs to a different co-location server.

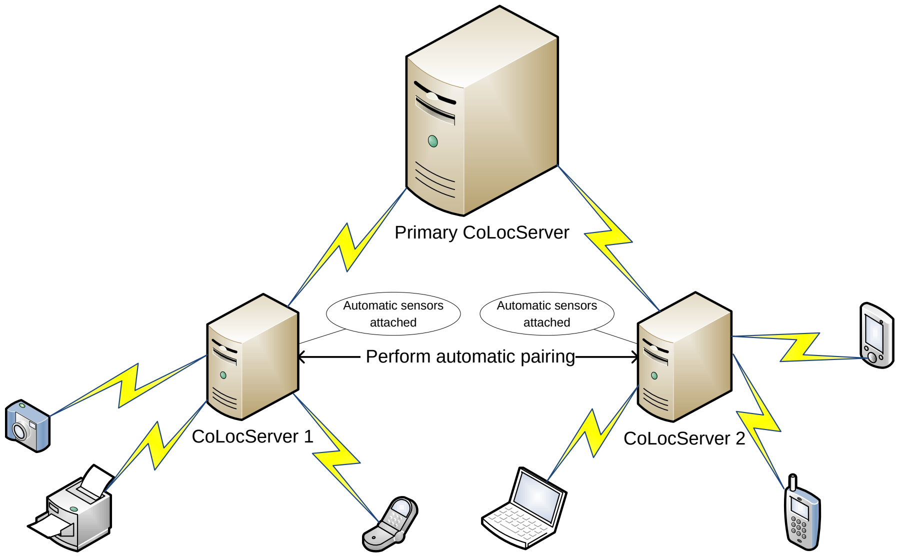

Figure 5.8: Scenario depicting the deployment of multiple co-location servers

Figure 5.8 shows the scenario of multiple co-location servers. In fact, it is similar to a single co-location server's scenario, where devices use the server as a mediator. Now the *Primary ColocServer* serves as a mediator for the other two servers (i.e. ColocServer1 and ColocServer2). These two servers can authenticate each other by either using an automatic pairing scheme, or with the help of a user/administrator using any other category of pairing schemes. Once these two servers are in a paired state, they can securely exchange the device's profiles to each other depending on the received queries from their clients.

In summary, the proposed system is designed in such a generic way that it is not restricted to any particular set of PoP protocols. It can be used with various types of PoP protocols or same PoP protocols, but with different selection criteria based on the scenario in which it is deployed. We have also shown that the proposed system is capable of getting user's preferences and considers them during the PoP protocols selection phase. The protocol selection mechanism (section 3.6) uses an XML-based policy as PoP protocols selection criteria, which is mainly defined in terms of required device capabilities and constraints over PoP protocols. Since the criterion for the selection of PoP protocols is described in an XML-based protocol specification and selection policy file; it can be changed / modified at run-time. Moreover, we also showed in chapter 3 (section 3.7) that the proposed system is extendable without changing the core design of the system and without substantial effort. All of these features indicate that the proposed is generic enough that it can cover a wide range of device pairing scenarios in ubiquitous computing environments in terms of both two device setting and group pairing.

5.2.4 PERFORMANCE ANALYSIS

The data obtained from both log files became the basis for performance analysis of the proposed system and the eight pairing schemes. Performance analysis is based on the average task completion times and the average task completion rates for each of the test cases. The graph presented in figure 5.9 shows the average task completion time along with the standard deviation. It shows that Button-to-Button and SiB schemes are faster than all of the other schemes. Their average task completion times are 19.161 and 20.209 seconds respectively. Blink-to-Button and SiC have similar approximate average task completion times. Similarly, Blink-Blink and CaS have similar approximate average task completion times. CoLoc has the largest average task completion time. Since CoLoc incorporates several PoP schemes as sub-protocols, it is the fact that the task completion time for CoLoc is dependent on the chosen PoP protocol.

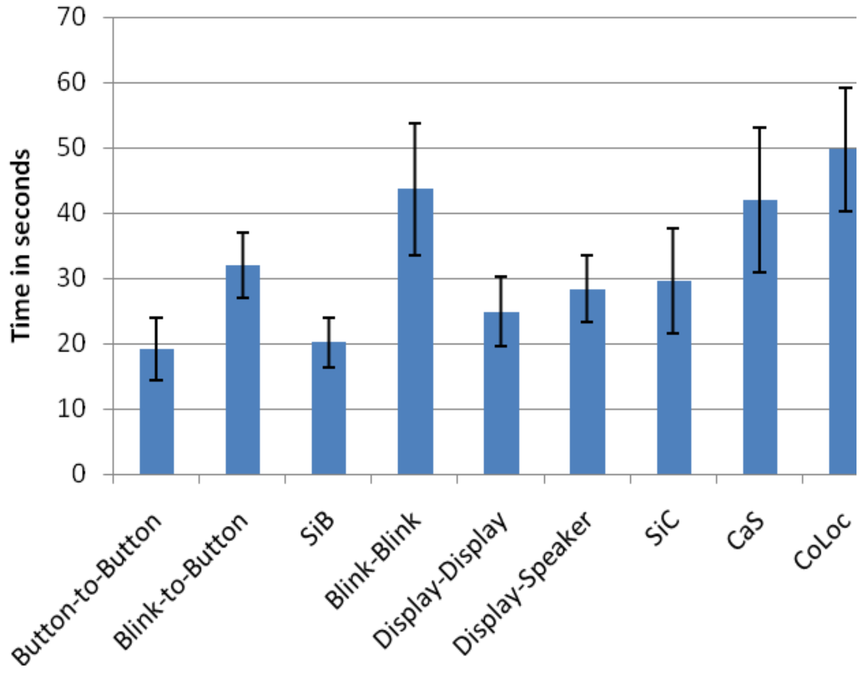

Figure 5.9: Average task completion time with standard deviation

Refer to the graph shown in figure 5.10 for task completion rates. The task completion rate for all of the schemes is good. The only schemes that could not achieve 100% completion rate are Blink-to-Button with 95% completion rate, Blink-Blink with 85% completion rate and Display-Speaker with 90% completion rate.

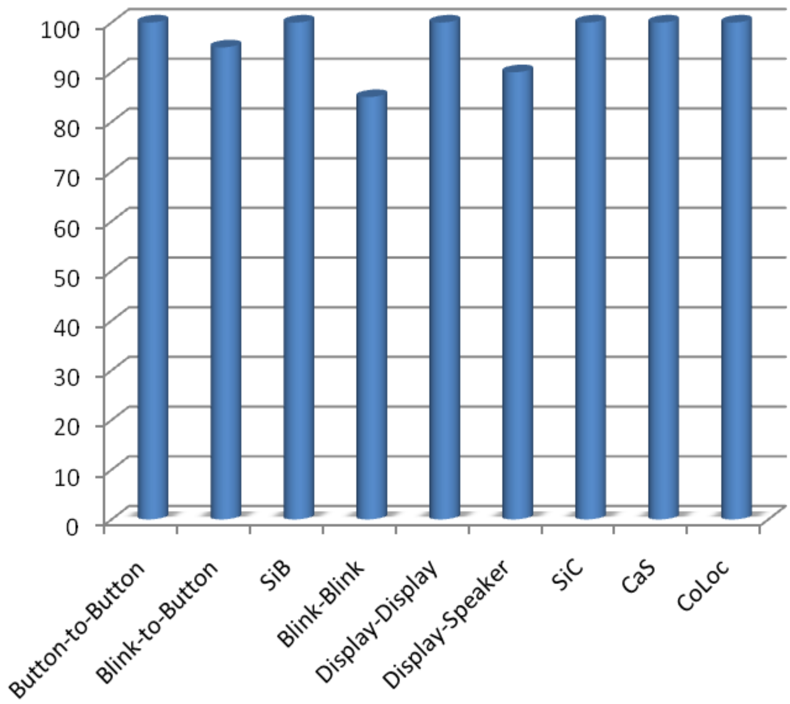

Figure 5.10: Task completion rate for each of the test case

Prior to performing the combined metrics analysis, we present the categorized summary of the usability study results in table 5.2.

PoP Schemes	Average Time (sec)	Std Dev.	Safe Errors (%)	Fatal Errors (%)	User Satisfaction on Scale of 1-7	Task Completion Rate (%)
Category One						
Button-to-Button	19.161	4.857	2.5	N/A	6.216	100
Blink-to-Button	31.982	5.036	10	N/A	5.416	95
Seeing is Believing	20.209	3.778	7.5	N/A	6.15	100
Category Two						
Blink-Blink	43.608	10.153	10	5	5.016	85
Display-Display	24.841	5.342	0	5	6.1	100
Display-Speaker	28.33	5.075	15	25	4.85	90
Category Three						
Selective Image Comparison	29.587	8.05	0	0	5.25	100
Capture and Show	42.033	11.046	0	15	5.566	100
Proposed System						
CoLoc Protocol	49.738	9.414	2.5	N/A	5.616	100

Table 5.2: The categorized summary of the overall results

5.2.5 COMBINED METRICS ANALYSIS

We calculated a single score from the collected raw data using the Single Usability Metric (SUM) model [100, 101] to rank each of the tested pairing schemes and the proposed system. SUM was originally introduced by Sauro and Kindlund [100] . It is a single, summated and standardized usability evaluation metric, which is based on the ANSI [102] and the ISO 9241 pt. 11 [103] defined dimension of usability. A SUM score is calculated from the four usability evaluation metrics, which are: post-task satisfaction, task completion rate, average completion time of task, and average number of errors. These four metrics are aggregated into a single measure (known as SUM score) through the standardization process outlined in [104].

Task Name	SUM Score			Task Completion Score	Task Satisfaction Score	Execution Time Score	Errors Score
	Low	Mean	High				
Seeing is Believing	76.8%	**83.8%**	97.9%	91.7%	97.6%	53.2%	92.8%
Display-Display	74.5%	**82.0%**	96.6%	91.7%	87.5%	53.6%	95.0%
Button-to-Button	72.8%	**80.5%**	96.3%	91.7%	83.3%	54.3%	92.8%
Coloc Protocol	61.8%	**72.2%**	89.4%	91.7%	51.1%	53.2%	92.8%
Capture and Show	60.4%	**71.3%**	87.9%	91.7%	47.2%	54.3%	91.9%
Selective Image Comparison	57.2%	**66.9%**	81.5%	91.7%	24.6%	54.6%	96.9%
Blink-to-Button	53.6%	**65.8%**	97.9%	87.7%	36.7%	52.6%	86.1%
Blink-Blink	48.1%	**59.8%**	74.9%	79.4%	14.1%	53.9%	91.9%
Display-Speaker	45.6%	**56.1%**	70.7%	83.6%	3.7%	53.0%	84.2%

Table 5.3: Overall ranking of schemes based on SUM scores

In order to avoid any errors in calculation, we have used an Ms Excel utility package to calculate the SUM score, which is designed by Sauro and available from [105]. This utility follows the original calculation process of the SUM score as outlined in [104] and contains the required pre-set functions, formulas and a set of sample data for illustration purposes. We entered all of the collected raw data into the downloaded spreadsheet package according to the specified rules and format, and finally got the results presented in table 5.3 using the confidence level of 95%. According to these results, SiB has the highest SUM score, while Display-Speaker has the lowest. Our proposed system is fourth with a SUM score of 72.2%, which interestingly out performs most of the schemes that are tested independent of the proposed system and do not include device registration and discovery phases.

5.3 WIDER VIEW OF USABILITY STUDY RESULTS

5.3.1 CATEGORIES RANKING

Graphs shown in figures 5.11 are drawn from the data obtained through post-test questionnaire (Q1 in appendix B.2). Results show the ranking for each of the category of PoP scheme as given by the participants. Note (as stated earlier) that category-1 is that in which user is involved in generating PoP data; category-2 is that in which user is responsible for verifying the PoP data; and category-3 is that in which user is involved in both generating the PoP data and verifying it. Category-1 consists of Button-to-Button, Blink-to-Button and Seeing is Believing schemes. Category-2 is composed of Blink-Blink, Display-Display and Display-Speaker schemes. Category-3 consists of Selective Image Comparison and Capture and Show schemes.

Results show that 85% of the participants ranked category-1 as the number one; it is ranked as number two by 15% participants, and none has ranked it as number three. Category-2 is ranked as number one by 15% participants; it is ranked as number two by 35% participants, and 50% of the participants ranked it as number three. In the case of category-3, none has ranked it as number one, however 50% of the participants ranked it as number two and remaining 50% of the participants ranked it as number three. These results revealed that most of the participants preferred the first category of PoP protocols in which they were involved in generating PoP data. None of the participants ranked the third category of PoP protocols as their favourite, while second category has a moderate ranking as few have ranked it first, some have ranked it second and the remaining ranked it third. Further, in category-1 the most preferred or satisfactory PoP protocol is Button-to-Button, in category-2 the most preferred PoP protocol is Display-Display, and in category-3 the most preferred PoP protocol is Capture and Show. These results suggest that users usually prefer to take part and willing to be involved in the pairing process. However, some of them prefer to be involved at the beginning of the proof-of-proximity phase in order to generate the PoP data, while others prefer to be involved during the last steps for verification of the PoP data; but majority of the users do not prefer to be involved in the process from the beginning to end of the pairing process. This advocates the fact that user involvement in the secure pairing process is unavoidable [106]; however users do not prefer excessive and unrestrained involvement or interactions during the pairing process.

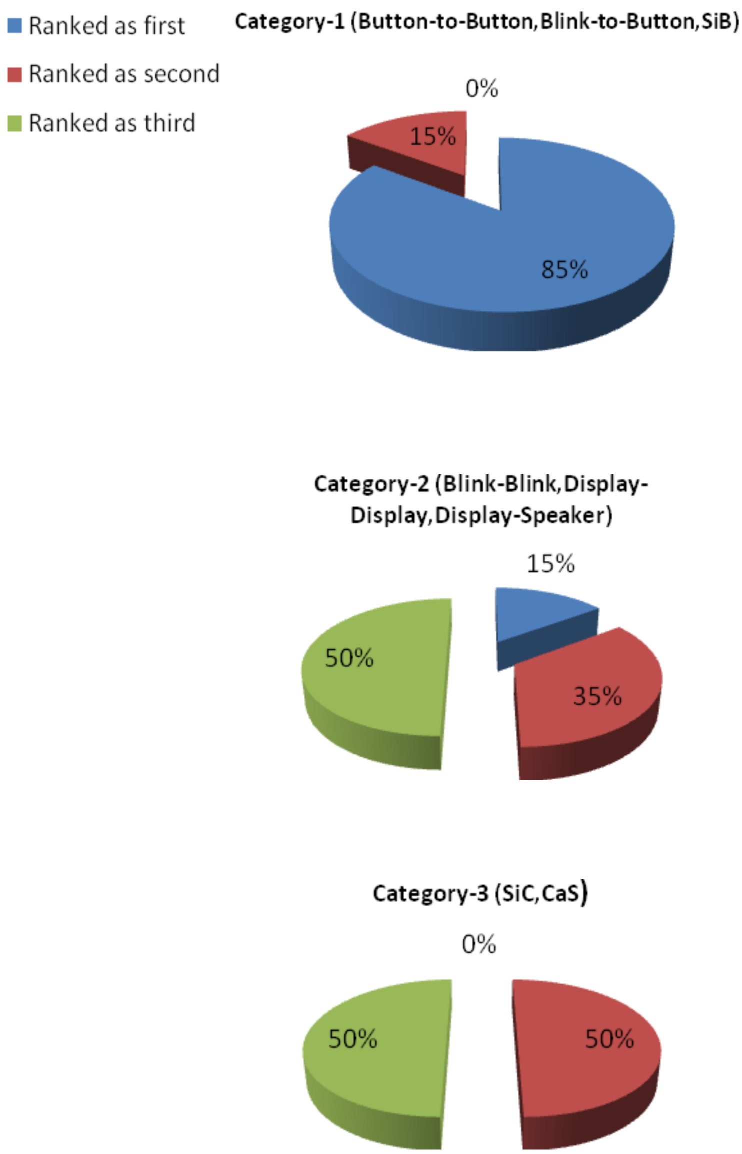

Figure 5.11: Category-wise ranking of pairing schemes

5.3.2 IMPACT OF GENDER

The usability study participants' recruitment process was performed without controlling any balance on the participant dependant variables, such as age and gender. Since there was little difference in the sample size for each gender (55% male and 45% female), we performed unpaired t-tests to investigate the effect of gender on average completion time and user satisfaction rating scores. Test results revealed that there is not a statistically significant effect of gender on participants' satisfaction ratings and task completion time.

5.3.3 IMPACT ON THE PROPOSED SYSTEM

The usability study of the eight pairing schemes has an impact on the proposed system. The results are useful in improving the PoP protocol selection mechanism. Particularly, the SUM score can be used to prioritize the PoP protocols. In this case, the decision mechanism used to set the recommended order of the PoP protocols would also consider the ranking as produced by the SUM score.

Further, categories ranking analysis presented in section 5.3.1 has also an impact on the protocol selection mechanism. Refer to section 3.6 in which we have described an XML-based PoP protocol specification and selection policy (figure 3.13). We have mentioned that the value of <UILevel> tag represents the level of required user interaction or involvement during the execution of the PoP protocol. '1' represents the low or minimum level of user interaction and '3' represents the high or maximum level of user interaction. The PoP protocols belonging to the category-1 and category-2 come under the categories of protocol that require low level of user interaction, while the category-3 comes under the category of protocols that require high level of user interaction. However, the ranking analysis proved to be helpful in drawing the scope for a set of protocols that require moderate level of user interaction. Based on the categories ranking analysis, we have put the PoP protocols belonging to category-2 under those protocols that require moderate level of user interaction (or more precisely moderate level of user attention).

5.4 SUMMARY

The detail of the usability study is presented along with the analysis of the results and evaluation of the proposed system. The analysis and evaluation supports the assertion that the integration of the discovery mechanism and several proof-of-proximity protocols into a single system is a more effective approach to device pairing as compared to proposing and developing a plethora of pairing protocols that work in a totally independent fashion. It is also highlighted that most of the recent work on device pairing has given less importance to certain aspects of device pairing, such as credential revocations and device un-pairing mechanisms. We not only realized the importance of such aspects of device pairing, but also incorporated them in the proposed system. In view of these facts, we believe that our work is an important and timely first step in academic research that highlights the need of a framework based approach to device pairing. Our work helps with answering several questions relevant to secure device pairing. These include: 1) are the users good at remembering several steps of dozens of pairing schemes for a number of device pairing scenarios and situations; 2) are they capable of performing well when cognitive overhead would be reduced; 3) are the users willing to be involved in the pairing process, and if yes, then to what extent; and most importantly 4) are the frame-work based approaches feasible for tackling the issue of device pairing in ubiquitous computing environments. The task of answering these questions was at least very difficult, if not impossible, before the work presented in this dissertation.

The additional results obtained from the usability study are in favour of a widely accepted view on the part of academic researchers, and the device manufacturers or industrial researchers that some form of human involvement in the secure pairing process is unavoidable [106]; however, the results also indicated that human users are interested in moderate involvement. They do not want to be overburdened with human-to-device interactions. The usability study results of 8 pairing schemes are also useful in improving the protocol selection criteria. Finally, we believe that the results and findings of this work motivates the research community to re-think the issue of secure device pairing and come up with a more standardized, common and universal solution.

CHAPTER 6

CONCLUSION

In this very last chapter, we present the summary of the overall work presented in this dissertation followed by the summary of the contributions, and future work. At the end of this chapter, we conclude this dissertation with closing remarks.

6.1 RECAPITULATION

We are moving towards a world in which computing is omnipresent and the security and privacy remain to be a major concern for this computing world – from traditional wired networks to modern ad hoc and ubiquitous computing systems. Ubiquitous computing systems differ from more traditional computing systems due to the ad-hoc and spontaneous nature of interactions among devices. Most of the time, these systems are composed of modern small, handheld or embedded devices, which support wireless communication in some form. Since ubiquitous computing systems use wireless communication, these are prone to security risks, such as eavesdropping, and require different techniques as compared to traditional security mechanisms.

Consequently, the problem of secure device pairing for ad hoc and ubiquitous comporting environments has had significant attention from many researchers during the last 10 years and a significant set of techniques and protocols have been proposed. More recently numerous standardization and industrial bodies, (such as Microsoft, WiFi Alliance, Bluetooth Special Interest Group, and the Universal Serial Bus (USB) Forum) have also recognized the significance of this problem, and are working on specifying more general, usable, and secure procedures for device pairing. However, as we have shown in our detailed analysis of the state-of-the-art in chapter 2, currently available schemes for secure device pairing vary in their security against different attacks, in the needed hardware capabilities and in the necessary level of user attention. Some of these techniques consider devices equipped with infrared, laser or ultrasound transceivers, whilst others require embedded accelerometers, cameras and/or LEDs, display, microphone and/or speakers. Some techniques exploit the knowledge of radio environment to securely pair the devices; others require the user's careful attention and significant manual intervention in pairing process.

We advocated that if we continue to multiply the pairing protocols each of which is feasible for certain scenarios only, users might be confused about the selection of appropriate pairing schemes as well as about the steps to follow them due to cognitive overhead of remembering several steps of dozens of pairing schemes. For instance, a user wanting to create an association of two mobile phones having a microphone, accelerometer, speaker, camera, display and infrared might be confused about the varied types of possibilities of device pairing protocols. We also advocate that in a world of modern heterogeneous devices and requirements, we need mechanisms to allow automated selection of the best protocols without requiring the user to have an in-depth knowledge of the minutiae of the underlying technologies. In view of that, at the end of chapter 2, we argued that it is appropriate to investigate ways of integrating different pairing protocols within a general architecture for providing secure and usable pairing mechanisms for a large set of scenarios in ubiquitous computing environments. As a consequence, we proposed a framework based approach to device pairing by demonstration of physical proximity. In chapter 3, we presented the three major goals of the system – usability, security and generality

– followed by the details of the proposed framework along with CoLoc protocol and PoP protocol selection mechanism. The key features of the PoP framework are:

Secure and Generic: The PoP framework integrates the discovery mechanism and a number of different pairing schemes mainly identified and discussed in chapter 2 of this dissertation. Since none of the surveyed discovery systems in their original form were found to be suitable in for integration and prototype implementation of the PoP framework in terms of complexity and the features offered by these systems, we designed our own confidentiality and integrity protected device registration and discovery mechanism through combining several features of the existing well known discovery systems. As a consequence, we showed that this way the proposed framework is able to provide support for various PoP protocols along with integrity and confidentiality protected device discovery, and thereby comprehensively provides support for a wide range of device pairing scenarios in ubiquitous computing environments.

Dynamic PoP protocol selection mechanism: The proposed framework follows a PoP protocols specification and selection policy when selecting a PoP protocol based on device capabilities and user preferences. The PoP protocol specification and selection mechanism is defined in an XML-based policy file. The XML-based policy allows modifying the protocol selection criterion at runtime. A runtime modification of the protocol selection criterion is useful for customizing the PoP protocol selection mechanism according to an individual user's or enterprise needs and preferences which were not foreseen at the time of deployment.

We showed the implementation of the proposed system and the CoLoc protocol along with the details of the software components and classification of the PoP protocols in chapter 4. The details of the usability study along with the evaluation of the proposed system were presented in chapter 5. The proposed system is evaluated through the analysis of the system's design features and the results obtained from the usability study. Evaluation demonstrated that the proposed system achieved its defined goals well.

6.2 SUMMARY OF CONTRIBUTIONS

We are summarizing the main contribution of this dissertation as the design and implementation of a Proof-of-Proximity framework for device pairing in ubiquitous computing environments along with the CoLoc protocol. The other contributions include a detailed critical and comparative analysis of the device pairing schemes, a simple device discovery mechanism, a protocol selection mechanism that is used to find the best possible scheme(s) to demonstrate the physical proximity of the devices according to the scenario/situation, and a usability study of eight pairing schemes along with the proposed system.

6.3 FUTURE WORK

The research work presented in this dissertation is complete in itself; however it has potential to be extended in order to further increase its efficiency and usage scenarios. Following is a non-exhausting list of possible future extensions to this work.

- The current prototype implementation of the proposed system utilizes a co-location server in order to store and manage the devices' profiles, however it is also possible that the proposed system could be implemented without the co-location server, in which case the devices are responsible to maintain the directory (as in SLP architecture there are two modes: directory based and directory-less). Alternatively, a directory-less implementation of the proposed system is also possible in which case instead of directory component, an out-of-band software component is required that should be responsible for secure exchange of the devices' capabilities information.
- We have performed usability study of some of the PoP protocols; however there is need for an exhaustive and more detailed usability study of the existing PoP protocols as well as new novel PoP protocols in more realistic scenarios and with more real-world and more diverse devices, such as mobile phones, PDAs, laptops. Further, we have also planned to extend the PoP protocol selection mechanism in such a way so that it takes into account the

results of the usability study. The impact of the usability study results on the proposed system is already described in section 5.3.3.

- The device registration and discovery process can be standardized using existing standards, such as CC/PP [107], for describing device profiles and discovery queries. Alternatively, the proposed system has also potential to be integrated with such a discovery mechanism that provide confidentiality and integrity during the registration and discovery process and also facilitates with the directory service in order to manage and maintain the device's profiles.
- Since we have implemented a simple protocol selection and specification policy, there is need to develop more efficient and usable policies, which in turn requires more exhaustive and real word usability testing of the proposed framework.

6.4 CLOSING REMARKS

It is noted that our focus in this work had been more on the ways of integrating the discovery mechanism and PoP protocols into one generic system for ubiquitous computing environments with minimal or non-substantial requirements, rather than on ways of providing new device pairing mechanisms each of which is feasible for an individual or particular scenario/situation of a completely ad hoc or infrastructure-less environment. Therefore, we believe that this work can be extended in two ways. Firstly, through performing a careful and more detailed analysis and/or usability of the existing pairing schemes in order to integrate more useful and realistic PoP protocols in the proposed framework as well as creating the new novel PoP protocols considering the design of the proposed framework. Secondly, as the proposed framework has potential to be used in peer-to-peer scenarios where there is no need of co-location server, so it can be extended in order to work in fully-infrastructure-less environments.

BIBLIOGRAPHY

1. Diffie, W. and M.E. Hellman, *New Directions in Cryptography*. IEEE Transactions on Information Theory, 1976. IT-22(6): p. 644--654.
2. Stajano, F., *The Resurrecting Duckling - What Next?*, in *Revised Papers from the 8th International Workshop on Security Protocols*. 2001, Springer-Verlag.
3. Stajano, F. and R. Anderson, *The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks*, in *Security Protocols*. 2000. p. 172-182.
4. Stajano, F. and R. Anderson, *The Resurrecting Duckling: security issues for ubiquitous computing.* Computer, 2002. 35(4): p. 22-26.
5. Naik, P., K. Ravichandran, and K.M. Sivalingam, *Cryptographic key exchange based on locationing information.* Pervasive and Mobile Computing, 2007. 3(1): p. 15-35.
6. Kindberg, T., K. Zhang, and N. Shankar. *Context authentication using constrained channels*. in *Proceedings of Fourth IEEE Workshop on Mobile Computing Systems and Applications*. 2002.
7. Saxena, N., et al., *Secure Device Pairing based on a Visual Channel.* in *IEEE Symposium on Security and Privacy.* June, 2006. Berkeley/Oakland, CA.
8. Saxena, N., M.B. Uddin, and J. Voris. *Universal Device Pairing using an Auxiliary Device*. in *Symposium On Usable Privacy and Security (SOUPS)*. 2008.
9. Saxena, N. and J. Voris. *Pairing Devices with Good Quality Output Interfaces*. in *International Workshop on Wireless Security and Privacy (WISP) (co-located with ICDCS).* 2009.
10. Prasad, R. and N. Saxena. *Efficient Device Pairing using Synchronized "Human-Comparable" Audiovisual Patterns*. in *Applied Cryptography and Network Security (ACNS)*. 2008.

11. Saxena, N. and M. Uddin, *Automated Device Pairing for Asymmetric Pairing Scenarios*, in *Information and Communications Security*. 2008. p. 311-327.
12. Soriente, C., G. Tsudik, and E. Uzun. *BEDA: Button-Enabled Device Association*. in *International Workshop on Security and Spontaneous Interaction (IWSSI 2007)*. 2007.
13. Holmquist, L.E., et al., *Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts*, in *Proceedings of the 3rd International Conference on Ubiquitous Computing*. 2001, Springer-Verlag: Atlanta, Georgia, USA.
14. Castelluccia, C. and P. Mutaf, *Shake Tthem Up!: A Movement-based Pairing Protocol for CPU-constrained Devices*, in *Proceedings of the 3rd International Conference on Mobile systems, Applications, and Services*. 2005, ACM: Seattle, Washington.
15. Mayrhofer, R. and H. Gellersen, *Shake Well Before Use: Authentication Based on Accelerometer Data*, in *5th International Conference on Pervasive Computing (Pervasive 2007)*. 2007.
16. Mayrhofer, R. and H. Gellersen. *Shake well before use: two implementations for implicit context authentication*. in *Adjunct Proceedings of Ubicomp'07*. 2007. Innsbruck, AT.
17. Shaked, Y. and A. Wool. *Cracking the Bluetooth PIN*. in *Proceedings of the 3rd International Conference on Mobile systems, Applications, and Services (MobiSys '05)*. 2005. Seattle, Washington: ACM.
18. Kirovski, D., M. Sinclair, and D. Wilson, *The Martini Synch: Using Accelerometers for Device Pairing*. Technical Report MSR-TR-2007-123, Microsoft Research. September 2007.
19. Soriente, C., G. Tsudik, and E. Uzun. *HAPADEP: Human Asisted Pure Audio Device Pairing*. Cryptology ePrint Archive, Report 2007/093.
20. Buhan, I., et al. *Secure Ad-hoc Pairing with Biometrics: SAfE*. in *Proceedings of First International Workshop on Security for Spontaneous Interaction (IWSSI ’07)*. 2007. Innsbruck, Austria.

21. Balfanz, D., et al. *Talking to strangers: Authentication in Adhoc Wireless Networks*. in *Symposium on Network and Distributed Systems Security (NDSS '02)*. 2002. San Diego, California.

22. Nicholson, A., et al., *LoKey: Leveraging the SMS Network in Decentralized, End-to-End Trust Establishment*, in *Pervasive Computing*. 2006. p. 202-219.

23. Buhan, I., et al., *Feeling is Believing: A Location Limited Channel Based on Grip Pattern Biometrics and Cryptanalysis.* Advances in Biometrics, 2007.

24. Spahic, A., et al., *Pre-Authentication using Infrared.* Privacy, Security, and Trust Within the Context of Pervasive Computing, 2005. Vol. 780: p. 105-112.

25. Mayrhofer, R., M. Hazas, and H. Gellersen, *An Authentication Protocol using Ultrasonic Ranging : Technical Report*. 2006, Lancaster University.

26. Mayrhofer, R. and M. Welch. *A Human-Verifiable Authentication Protocol Using Visible Laser Light*. in *the 2nd International Conference on Availability, Reliability and Security (ARES'07).* 2007.

27. Gehrmann, C. and C.J. Mitchell, *Manual Authentication for Wireless Devices.* RSA Cryptobytes, 2004. Vol. 7(1): p. 29–37.

28. Varshavsky, A., et al., *Amigo: Proximity-Based Authentication of Mobile Devices*, in *Ubiquitous Computing (UbiComp'07)*. 2007. p. 253-270.

29. McCune, J.M., A. Perrig, and M.K. Reiter, *Seeing-is-Believing: Using Camera Phones for Human-Verifiable Authentication.* in *IEEE Symposium on Security and Privacy (SP'05)*. 2005. p. 110 - 124.

30. Ringwald, M. *Spontaneous Interaction with Everyday Devices Using a PDA.* in *Proceedings of the Workshop on Supporting Spontaneous Interaction in Ubiquitous Computing Settings, (Co-located with Ubicomp02)*. 2002. Gothenburg, Sweden.

31. Goodrich, M.T., et al. *Loud and Clear: Human-Verifiable Authentication Based on Audio*. in *Proceedings of 26th IEEE International Conference on Distributed Computing Systems (ICDCS'06).* 2006.

32. Katz, J. and Y. Lindell, *Introduction to Modern Cryptography: Principles and Protocols*. Cryptography and Network Security Series. August 2007: Chapman & Hall/CRC, ISBN: 978-1584885511. 552.

33. Menezes, A., P.v. Oorschot, and S. Vanstone, *Handbook of Applied Cryptography* 1996: CRC Press.

34. Schneier, B., *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. second ed. 1996: John Wiley & Sons, Inc.

35. Nguyen, L.H. and A.W. Roscoe, *Authenticating Ad hoc Networks by Comparison of Short Digests.* Inf. Comput., 2008. 206(2-4): p. 250-271.

36. Creese, S., et al., *Authentication for Pervasive Computing*, in *Security in Pervasive Computing*. 2004. p. 116-129.

37. Zakiuddin, I., et al., *Authentication in Pervasive Computing, Position Paper*, in *First International Conference on Security in Pervasive Computing*. 2003. Germany.

38. Staffans, L.A. and T. Saridakis, *An Authorization and Access Control Scheme for Pervasive Computing*, Telecommunications Software and Multimedia Laboratory, Helsinki University of Technology: Finland.

39. Jos, R. and N. Davies, *Scalable and Flexible Location-Based Services for Ubiquitous Information Access*, in *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*. 1999, Springer-Verlag: Karlsruhe, Germany.

40. Tao, S. and M.K. Denko. *A Distributed Trust Management Scheme in the Pervasive Computing Environment*. in *Canadian Conference on Electrical and Computer Engineering (CCECE'07).* 2007.

41. Robinson, P. and M. Beigl. *Trust Context Spaces: An Infrastructure for Pervasive Security in Context-Aware Environments*. in *1st International Conference on Security in Pervasive Computing*. 2003.

42. Kagal, L., T. Finin, and A. Joshi, *Moving from Security to Distributed Trust in Ubiquitos Computing Environments.* IEEE Computer, 2001.

43. Kagal, L., T. Finin, and A. Joshi, *Trust-based Security in Pervasive Computing Environments.* Computer, 2001. 34(12): p. 154-157.

44. Kirovski, D., M. Sinclair, and D. Wilson, *The Martini Synch: Joint Fuzzy Hashing Via Error Correction*, in *Security and Privacy in Ad-hoc and Sensor Networks*. 2007. p. 16-30.

45. Lester, J., B. Hannaford, and G. Borriello. *Are You with Me?" - Using Accelerometers to Determine If Two Devices Are Carried by the Same Person*. in *Pervasive Computing*. 2004: Springer-Verlag (2004) pg. 33-50.

46. Mayrhofer, R. and H. Gellersen. *On the Security of Ultrasound as Out-of-Band Channel*. in *IEEE International Symposium on Parallel and Distributed Processing (IPDPS'07)*. 2007.

47. Mayrhofer, R., H. Gellersen, and M. Hazas, *Security by Spatial Reference: Using Relative Positioning to Authenticate Devices for Spontaneous Interaction*, in *Ubiquitous Computing (UbiComp'07)*. 2007. p. 199-216.

48. Ateniese, G., M. Steiner, and G. Tsudik. *Authenticated Group Key Agreement and Friends*. in *Proceedings of the 5th ACM Conference on Computer and Communications Security*. November 1998, San Francisco, CA.

49. Rafaeli, S. and D. Hutchison, *A Survey of Key Management for Secure Group Communication.* ACM Computing Surveys (CSUR), September 2003. Vol.35, Issue 3: p. 309-329.

50. Kindberg, T. and K. Zhang. *Validating and Securing Spontaneous Associations between Wireless Devices*. in *Proceedings of 6th International Conference on Information Security (ISC'03)*. October 2003. Bristol, UK.

51. *Kirovski, D., M. Sinclair*, and *D. Wilson*, *The Martini Synch.* Technical Report MSR-TR-2007-123, Microsoft Research, September 2007.

52. Kumaran, S.I., *Jini Technology: An Overview*. 2001: Prentice Hall PTR, Upper Saddle River, NJ, USA.

53. Jakobsson, M. and S. Wetzel, *Security Weaknesses in Bluetooth.* Lecture Notes in Computer Science, 2001. 2020: p. 176+.

54. Bluetooth Special Interest Group (SIG), *Bluetooth specifications 1.0 – 2.1+EDR. Technical specifications, 1999–2007.* http://www.bluetooth.com.

55. Haataja, K. and P. Toivanen. *Practical Man-in-the-Middle Attacks Against Bluetooth Secure Simple Pairing*. in *Proceedings of 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'08)*. October 2008.

56. Lindell, A.Y., *Attacks on the Pairing Protocol of Bluetooth v2.1*. June, 2008, URL:http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.146.4653.

57. Haselsteiner, E. and K. Breitfuß. *Security in Near Field Communication (NFC)*. in *Proceedings of Workshop on RFID Security*, pp 3-13. July 2006.

58. Goldberg, I., *Visual key fingerprint code*. 1996, Available at: http://www.cs.berkeley.edu/iang/visprint.c.

59. Levien, R., *PGP snowflake*. 1996, Personal Communication.

60. Ellison, C. and S. Dohrmann, *Public-key Support forGgroup Collaboration.* ACM Transactions on Information and System Security (TISSEC), November 2003. Vol. 6 , Issue 4: p. 547 - 565.

61. Perrig, A. and D. Song. *Hash Visualization: A New Technique to Improve Real-world Security*. in *International Workshop on Cryptographic Techniques and E-Commerce*. 1999.

62. James Kempf and P.S. Pierre, *Service Location Protocol for Enterprise Networks: Implementing and Deploying a Dynamic Service Finder*, ed. C.A. Long. 1999, Canada: John Wiley & Sons, Inc.

63. Miller, B.A. and C. Bisdikian, *Bluetooth Revealed*. 2nd edition. 2001: Prentice Hall PTR, Upper Saddle River, NJ, USA.

64. Bluetooth SIG (Special Interest Group). 2010, URL:https://www.bluetooth.org/apps/content/.

65. Universal Plug and Play (UPnP) Forum. 2010, URL: http://www.upnp.org/.

66. The Community Resource for Jini Technology. 2010, URL:http://www.jini.org.

67. Zhu, F., M. Mutka, and LionelNi, *Classification of Service Discovery in Pervasive Computing Environments*. Technical Report, Michigan State University, 2002.

68. Bettstetter, C. and C. Renner. *A Comparison of Service Discovery Protocols and Implementation of the Service Location Protocol*. in *Proceedings of 6th EUNICE Open European Summer School (EUNICE-2000)*. 2000. Twente, Netherlands.

69. Ververidis, C.N. and G.C. Polyzos, *Service Discovery for Mobile Ad Hoc Networks: A Survey of Issues and Techniques.* Communications Surveys & Tutorials, IEEE, 2008. 10(3): p. 30-45.

70. E. Guttman, C.P., J. Veizades, M. Day, *Service Location Protocol Version 2*. June 1999, URL: http://tools.ietf.org/html/rfc2608, IETF RFC 2608.

71. Droms, R., *Automated Configuration of TCP/IP with DHCP.* IEEE Internet Computing. 2(4): p. 45-53.

72. Cai, T., et al. *Simple Service Discovery Protocol*. Internet Engineering Task Force (IETF), INTERNET DRAFT: draft-cai-ssdp-v1-01.txt, October 1999.

73. Introduction to XML Schema, W3C School, 2010, URL: http://www.w3schools.com/schema/schema_intro.asp.

74. XML Utilities, *HiTSOFTWARE*. 2010, URL:http://www.hitsw.com/xml_utilites/.

75. Dutta, R. and R. Barua, *Password-Based Encrypted Group Key Agreement.* International Journal of Network Security, July 2006. vol. 3, No. 1, pp. 23-34.

76. Raju, D.V.N., V.V. Kumari, and K.V. Raju, *Efficient Distribution of Conference Key for Dynamic Groups.* International Journal of Computer Theory and Engineering, August 2010. Vol. 2, No. 4, pp.559-563.

77. Augot, D., et al., *A Three Round Authenticated Group Key Agreement Protocol for Ad hoc Networks.* Pervasive and Mobile Computing, 2007. 3(1): p. 36-52.

78. Rahman, R.H. and M.L. Rahman, *A Passwork-Based Group Key Agreement Protocol for Wireless Ad-Hoc Networks.* Asian Journal of Information Technology, 2007. 6(11): p. 1181-1186.

79. Tseng, Y.-M., *A Secure Authenticated Group Key Agreement Protocol for Resource-limited Mobile Devices.* The Computer Journal, Oxford University Press, Oxford, UK, January 2007. 50(1): p. 41-52.

80. *Phidgets: Products for USB Sensing and Control*. 2010, URL:http://www.phidgets.com/index.php.

81. *SQLite*. 2010, URL: http://www.sqlite.org/.

82. *HyperSQL*. 2010, URL: http://hsqldb.org/.

83. McObject, *Perst/Perst Lite*. 2010, URL: http://www.mcobject.com/perst.

84. Vohra, D. *An embedded XML Database: Oracle Berkeley DB XML*. July 2007, URL: http://www.theregister.co.uk/2007/07/18/berkeley_db_xml/.

85. A Comparison of Oracle Berkeley DB and Relational Database Management Systems, *An Oracle Technical White Paper*, March 2009.

86. Olson, M.A., K. Bostic, and M. Seltzer. *Berkeley DB*. in *USENIX Annual Technical Conference (FREENIX Track)*. June 6–11, 1999. Monterey, California, USA.

87. Oracle Berkeley DB XML. 2010, URL:http://www.oracle.com/us/products/database/berkeley-db/index-066571.html.

88. Kumar, A., et al. *Caveat Emptor: A Comparative Study of Secure Device Pairing methods*. in *IEEE International Conference on Pervasive Computing and Communications (PerCom-09)*. 2009.

89. Uzun, E., K. Karvonen, and N. Asokan, *Usability Analysis of Secure Pairing Methods*, Financial Cryptography and Data Security, S. Dietrich and R. Dhamija, Editors. 2007, Springer Berlin / Heidelberg. p. 307-324.

90. Valkonen, J., A. Toivonen, and K. Karvonen. *Usability Testing for Secure Device Pairing in Home Networks*. in *UbiComp'07 Workshop Proceedings*. 2007. Innsbruck, Austria.

91. Kumar, A., et al., *A Comparative Study of Secure Device Pairing Methods.* Pervasive and Mobile Computing, 2009. 5(6): p. 734-749.

92. Kobsa, A., et al., *Serial Hook-ups: A Comparative Usability Study of Secure Device Pairing Methods*, in *Proceedings of the 5th Symposium on Usable Privacy and Security*. 2009, ACM: Mountain View, California. p. 1-12.

93. Kainda, R., I. Flechais, and A.W. Roscoe, *Usability and Security of Out-of-Band Channels in Secure Device Pairing Protocols*, in *Proceedings of the 5th Symposium on Usable Privacy and Security*. 2009, ACM: Mountain View, California. p. 1-12.

94. Faulkner, L., *Beyond the Five-user Assumption: Benefits of Increased Sample Sizes in Usability Testing.* Behavior Research Methods, Instruments, & Computers, 2003. 35(3): p. 379-383.

95. Lewis, J.R., *IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instruction for Use.* International Journal of Human-Computer Interaction, L. Erlbaum Associates Inc.; Hillsdale, NJ, USA, Jan-March 1995. 7(1): p. 57 - 78.

96. Nunnaly, J.C., *Psychometric Theory*, ed. R.R. Wright and M. Gardner. 1978, New York, USA: McGraw-Hill, Inc.

97. Bierton, R. and R. Bates. *Experimental Determination of Optimal Scales for Usability Questionnaire Design*. in *Proceedings of Human Computer Interaction (HCI-2000)*. 2000.

98. Nielsen, J. and J. Levy, *Measuring Usability: Preference vs. Performance.* Communications of the ACM, April, 1994. 37(4): p. 66-75.

99. Uzun, E., K. Karvonen, and N. Asokan, *Usability Analysis of Secure Pairing Methods*, Financial Cryptography and Data Security. 2008. p. 307-324.

100. Sauro, J. and E. Kindlund, *A Method to Standardize Usability Metrics into a Single Score*, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2005, ACM: Portland, Oregon, USA.

101. Sauro, J. and E. Kindlund. *Using a Single Usability Metric (SUM) to Compare the Usability of Competing Products*. February, 2010,

URL:http://www.measuringusability.com/papers/HCII2005_sauro_kindlund-V9.pdf.

102. Common Industry Format for Usability Test Reports (ANSI-NCITS 354-2001). *American National Standards Institute*. 2001, Washington, DC.

103. Ergonomic Requirements for Office Work with Visual Display Terminal (VDTs), in *Part 11: Guidance on usability (ISO 9241-11:1998(E))*. 1998: Geneva, Switzerland.

104. Sauro, J. and E. Kindlund. *Making Sense of Usability Metrics: Usability and Six Sigma*. in *Proceedings of the 14th Annual Conference of the Usability Professionals Association*. 2005. Montreal, Canada.

105. Sauro, J. *Measuring Usability: Quantitative Usability, Statistics & Six Sigma*. February, 2010, URL: http://www.measuringusability.com/SUM/index.htm.

106. Suomalainen, J., J. Valkonen, and N. Asokan, *Security Associations in Personal Networks: A Comparative Analysis.* Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Volume 4572/2007: p. 43-57.

107. Reynolds, F., et al., *Composite Capability / Preference Profiles (CC/PP): A User Side Framework for Content Negotiation*, W3C NOTE-CCPP-19990727, July 1999, URL: http://www.w3.org/TR/NOTE-CCPP/.

APPENDIX A ABBREVIATIONS

B-to-B	Button to Button
CaS	Capture and Show
CC/PP	Composite Capabilities/Preference Profile
CoLoc	Co-Location Protocol
DoS	Denial of Service
DTD	Document Type Definition
GUI	Graphical User Interface
IETF	Internet Engineering Task Force
LED	Light Emitting Diode
MAC	Message Authentication Code
MiTM	Man-in-The-Middle
NFC	Near Field Communication
OOB	Out-of-Band
PDA	Personal Digital Assistant
PoP	Proof-of-Proximity
SiB	Seeing-is-Believing
SIC	Selective Image Comparison
SIG	Special Interest Group
SLP	Service Location Protocol
SSP	Secure Simple Pairing
UPnP	Universal Plug and Play
XML	Extensible Markup Language

APPENDIX B.1 PRE-TEST QUESTIONNAIRE

Participant's Demographic Information

Please circle (e.g. Ⓒ) an option as appropriate for each of the following.

(1) Please specify your age group:

a. 18 – 25
b. 26 – 40
c. 41 or above

(2) Sex:

a. Male
b. Female

(3) Highest academic qualification:

a. High School/College
b. Bachelors Degree
c. Masters Degree
d. Doctorate (PhD)

(4) How many hours usually do you work with computer in a day since last one year? (You can give an average estimate).

a. 2 or less hours
b. 3 – 5 hours
c. 6 or above hours

(5) Do you have experience of pairing devices (e.g. pairing a Bluetooth-enabled headset with MP3 player or the pairing of two mobile phones)?

a. Yes
b. No

APPENDIX B.2 POST-TEST QUESTIONNAIRE

Please answer the following questions based on your experience of the performed usability test of pairing schemes/methods.

(1) Please rank the following three categories of pairing schemes by numbering them (i.e. from 1 to 3).

a. Selective Image Comparison and Capture and Show: __________
b. Button-to-Button, Blink-to-Button and Seeing is Believing: __________
c. Blink-Blink, Display-Display and Display-Speaker: __________

(2) Consider the scenario that you have to pair a mobile phone having display, speaker and keypad capabilities with a printer having very limited display, LEDs and buttons on it in a noisy environment. Please select all of the pairing schemes that you think are applicable in this scenario.

a. Button-to-Button (B-to-B)
b. Blink-to-Button (Blink-to-B)
c. Seeing is Believing (SiB)
d. Blink-Blink
e. Display-Display
f. Display-Speaker
g. Selective Image Comparison (SiC)
h. Capture and Show (CaS)
i. Don't know

(3) If you have to pair a mobile phone having display, speaker and keypad capabilities with a printer having very limited display, LEDs and buttons on it, then which of the following do you think is the best scheme to pair these two devices?

a. Button-to-Button (B-to-B)
b. Display-Speaker
c. Blink-to-Button (Blink-to-B)
d. Blink-Blink

(4) We would appreciate it, if you could give any suggestions or comments that you think will help us to improve the proposed system. Please also feel free to discuss your general feedback orally.

Thanks for giving your valuable time and feedback

APPENDIX B.3 AFTER SCENARIO QUESTIONNAIRE – 1

Please rate the usability of the pairing schemes used during the previous phase. Please circle the score/number (i.e. 1 is the lowest score and 7 is the highest score) that you think is appropriate for each of the presented items.

(A) Button-to-Button (B-to-B) Pairing Scheme

(1) How would you describe how difficult or easy it was to complete the Button-to-Button pairing scheme?

Very Difficult **Very Easy**

1 2 3 4 5 6 7

(2) How satisfied are you with using Button-to-Button pairing scheme to pair the two devices?

Very Unsatisfied **Very Satisfied**

1 2 3 4 5 6 7

(3) How would you rate the amount of time Button-to-Button pairing scheme took to complete the pairing process?

Too Much Time **Very Little Time**

1 2 3 4 5 6 7

(B) Blink-to-Button (Blink-to-B) Pairing Scheme

(4) How would you describe how difficult or easy it was to complete the Blink-to-Button pairing scheme?

Very Difficult **Very Easy**

1 2 3 4 5 6 7

(5) How satisfied are you with using Blink-to-Button pairing scheme to pair the two devices?

Very Unsatisfied **Very Satisfied**

1 2 3 4 5 6 7

(6) How would you rate the amount of time Blink-to-Button pairing scheme took to complete the pairing process?

Too Much Time **Very Little Time**

1 2 3 4 5 6 7

(C) Seeing is Believing (SiB) Pairing Scheme

(7) How would you describe how difficult or easy it was to complete the Seeing is Believing pairing scheme?

Very Difficult						**Very Easy**
1	**2**	**3**	**4**	**5**	**6**	**7**

(8) How satisfied are you with using Seeing is Believing pairing scheme to pair the two devices?

Very Unsatisfied						**Very Satisfied**
1	**2**	**3**	**4**	**5**	**6**	**7**

(9) How would you rate the amount of time Seeing is Believing pairing scheme took to complete the pairing process?

Too Much Time						**Very Little Time**
1	**2**	**3**	**4**	**5**	**6**	**7**

(D) Blink-Blink Pairing Scheme

(10) How would you describe how difficult or easy it was to complete the Blink-Blink pairing scheme?

Very Difficult						**Very Easy**
1	**2**	**3**	**4**	**5**	**6**	**7**

(11) How satisfied are you with using Blink-Blink pairing scheme to pair the two devices?

Very Unsatisfied						**Very Satisfied**
1	**2**	**3**	**4**	**5**	**6**	**7**

(12) How would you rate the amount of time Blink-Blink pairing scheme took to complete the pairing process?

Too Much Time						**Very Little Time**
1	**2**	**3**	**4**	**5**	**6**	**7**

(E) Display-Display Pairing Scheme

(13) How would you describe how difficult or easy it was to complete the Display-Display pairing scheme?

Very Difficult						**Very Easy**
1	**2**	**3**	**4**	**5**	**6**	**7**

(14) How satisfied are you with using Display-Display pairing scheme to pair the two devices?

Very Unsatisfied **Very Satisfied**

1 2 3 4 5 6 7

(15) How would you rate the amount of time Display-Display pairing scheme took to complete the pairing process?

Too Much Time **Very Little Time**

1 2 3 4 5 6 7

(F) Display-Speaker Pairing Scheme

(16) How would you describe how difficult or easy it was to complete the Display-Speaker pairing scheme?

Very Difficult **Very Easy**

1 2 3 4 5 6 7

(17) How satisfied are you with using Display-Speaker pairing scheme to pair the two devices?

Very Unsatisfied **Very Satisfied**

1 2 3 4 5 6 7

(18) How would you rate the amount of time Display-Speaker pairing scheme took to complete the pairing process?

Too Much Time **Very Little Time**

1 2 3 4 5 6 7

(G) Selective Image Comparison (SiC) Pairing Scheme

(19) How would you describe how difficult or easy it was to complete the Selective Image Comparison pairing scheme?

Very Difficult **Very Easy**

1 2 3 4 5 6 7

(20) How satisfied are you with using Selective Image Comparison pairing scheme to pair the two devices?

Very Unsatisfied **Very Satisfied**

1 2 3 4 5 6 7

(21) How would you rate the amount of time Selective Image Comparison scheme took to complete the pairing process?

Too Much Time **Very Little Time**

1 2 3 4 5 6 7

(H) Capture and Show (CaS) Pairing Scheme

(22) How would you describe how difficult or easy it was to complete the Capture and Show pairing scheme?

Very Difficult **Very Easy**

1	**2**	**3**	**4**	**5**	**6**	**7**

(23) How satisfied are you with using Capture and Show pairing scheme to pair the two devices?

Very Unsatisfied **Very Satisfied**

1	**2**	**3**	**4**	**5**	**6**	**7**

(24) How would you rate the amount of time Capture and Show pairing scheme took to complete the pairing process?

Too Much Time **Very Little Time**

1	**2**	**3**	**4**	**5**	**6**	**7**

APPENDIX B.4 AFTER SCENARIO QUESTIONNAIRE – 2

Please rate the usability of the device pairing system used during the previous phase (second part). Please circle the score/number (i.e. 1 is the lowest score and 7 is the highest score) that you think is appropriate for each of the presented items.

(A) Device Pairing by Demonstration of Physical Proximity System

(1) How would you describe how difficult or easy it was to complete the device pairing process using that system?

Very Difficult **Very Easy**

1 2 3 4 5 6 7

(2) How satisfied are you with using the system to pair the two devices?

Very Unsatisfied **Very Satisfied**

1 2 3 4 5 6 7

(3) How would you rate the amount of time the system took to complete the pairing process?

Too Much Time **Very Little Time**

1 2 3 4 5 6 7

APPENDIX C DOCUMENT TYPE DEFINITIONS (DTDS)

1. DTD for PoP Protocol Specification and Selection Policy

```
<!ELEMENT PSPolicy ( Protocol+ ) >
<!ELEMENT Protocol ( Name, Type, ClCapabilities, ResCapabilities,
ProximityLimit, UILevel, Constraints?) >
<!ELEMENT Name ( #PCDATA ) >
<!ELEMENT Type ( #PCDATA ) >
<!ELEMENT ClCapabilities ( #PCDATA ) >
<!ELEMENT ResCapabilities ( #PCDATA ) >
<!ELEMENT ProximityLimit ( #PCDATA ) >
<!ELEMENT UILevel ( #PCDATA ) >
<!ELEMENT Constraints ( #PCDATA ) >
```

APPENDIX D RAW DATA OBTAINED FROM QUESTIONNAIRES

D.1: Table of raw data for question 1 of the post-test questionnaire

User	Ranked as 1	Ranked as 2	Ranked as 3
1	1	3	2
2	1	3	2
3	1	3	2
4	1	3	2
5	1	2	3
6	1	3	2
7	1	2	3
8	1	3	2
9	2	1	3
10	1	2	3
11	1	2	3
12	1	2	3
13	2	1	3
14	1	3	2
15	1	3	2
16	1	2	3
17	2	1	3
18	1	2	3
19	1	3	2
20	1	3	2
Key for columns 2, 3 and 4 : 1 denotes category-1 2 denotes category-2 3 denotes category-3			

D.2: Table of raw data for question 2 of the post-test questionnaire

User	Button-to-Button	Blink-to-Button	SiB	Blink-Blink	Display-Display	Display-Speaker	SiC	CaS	Don't Know
1	1	1	0	1	1	1	0	0	0
2	1	1	0	0	1	0	0	0	0
3	1	1	0	0	1	1	0	0	0
4	1	1	0	0	1	0	0	0	0
5	1	1	0	0	0	0	0	0	0
6	1	1	0	0	0	1	0	0	0
7	1	1	0	1	1	1	0	0	0
8	1	1	0	0	1	0	0	0	0
9	1	1	0	1	1	1	0	0	0
10	1	1	0	0	0	0	0	0	0
11	1	1	0	0	1	0	0	0	0
12	1	1	0	0	1	1	0	0	0
13	1	1	0	0	1	0	0	0	0
14	1	0	0	0	1	1	0	0	0
15	1	1	0	0	0	0	0	0	0
16	1	1	0	0	1	1	0	0	0
17	1	1	0	0	0	0	0	0	0
18	1	1	0	0	1	0	0	0	0
19	1	1	0	1	1	1	1	0	0
20	1	1	0	0	0	0	0	0	0

Key for columns 2 to 9:

1 denotes that user selected the choice that is shown in the column heading

0 denotes that user did not select the choice that is shown in the column heading

D.3: Table of raw data for question 3 of the post-test questionnaire

User	B-to-B	Display-Speaker	Blink-to-B	Blink-Blink
1	1	0	0	0
2	1	0	0	0
3	1	0	0	0
4	1	0	0	0
5	1	0	0	0
6	1	0	0	0
7	1	0	0	0
8	1	0	0	0
9	1	0	0	0
10	1	0	0	0
11	1	0	0	0
12	1	0	0	0
13	1	0	1	0
14	1	1	0	0
15	1	0	0	0
16	1	0	1	0
17	1	0	0	0
18	1	0	0	0
19	1	0	0	0
20	1	0	0	0

Key for columns 2 to 5:

1 denotes that user selected the choice that is shown in the column heading

0 denotes that user did not select the choice that is shown in the column heading

KEY FOR SUBSEQUENT TABLES:

Sat-Score1: user rating for the question:

*How would you describe how difficult or easy it was to complete the **<PairingSchemeName>** pairing scheme?*

Sat-Score2: user rating for the question:

*How satisfied are you with using **<PairingSchemeName>** pairing scheme to pair the two devices?*

Sat-Score3: user rating for the question:

*How would you rate the amount of time **<PairingSchemeName>** pairing scheme took to complete the pairing process?*

Also note that values in columns 1 to 3 indicates the score as given by the user where 1 is the lowest score and 7 is the highest score.

D.4: Table of raw data for Button-to-Button scheme obtained from ASQ1

User	Sat-Score1	Sat-Score2	Sat-Score3
1	6	6	6
2	7	6	7
3	5	6	6
4	7	7	6
5	6	6	6
6	7	7	7
7	7	7	7
8	7	6	6
9	6	6	6
10	6	7	7
11	7	7	7
12	5	5	5
13	6	6	6
14	7	6	6
15	6	6	6
16	7	6	7
17	5	5	5
18	6	6	6
19	6	6	6
20	7	6	6

D.5: Table of raw data for Blink-to-Button scheme obtained from ASQ1

User	Sat-Score1	Sat-Score2	Sat-Score3
1	5	6	6
2	6	6	6
3	5	5	6
4	6	6	5
5	6	6	6
6	5	6	6
7	6	5	6
8	6	5	6
9	5	6	5
10	6	6	6
11	5	5	5
12	6	6	6
13	6	5	5
14	5	6	6
15	5	4	5
16	5	6	5
17	5	4	5
18	6	5	5
19	4	4	5
20	5	5	5

D.6: Table of raw data for SiB scheme obtained from ASQ1

User	Sat-Score1	Sat-Score2	Sat-Score3
1	6	7	6
2	7	6	6
3	6	6	6
4	6	6	6
5	6	6	7
6	7	7	7
7	6	7	6
8	6	6	6
9	6	6	7
10	6	6	6
11	6	6	6
12	7	6	6
13	6	6	6
14	6	6	6
15	6	6	6
16	6	6	6
17	6	6	6
18	6	6	6
19	6	6	6
20	6	6	6

D.7: Table of raw data for Blink-Blink scheme obtained from ASQ1

User	Sat-Score1	Sat-Score2	Sat-Score3
1	5	6	5
2	6	6	5
3	5	5	5
4	6	5	5
5	6	6	5
6	5	5	5
7	5	6	5
8	5	4	5
9	5	5	4
10	5	4	5
11	6	5	5
12	5	5	5
13	6	5	6
14	5	5	5
15	6	5	4
16	4	4	4
17	5	6	6
18	4	4	5
19	5	5	4
20	5	4	4

D.8: Table of raw data for Display-Display scheme obtained from ASQ1

User	Sat-Score1	Sat-Score2	Sat-Score3
1	7	6	6
2	6	6	6
3	7	6	6
4	7	6	6
5	5	6	6
6	6	6	6
7	7	6	6
8	6	6	6
9	6	6	6
10	6	6	6
11	7	7	6
12	6	6	6
13	7	6	7
14	6	6	6
15	7	6	6
16	7	7	6
17	6	6	6
18	6	6	5
19	5	5	5
20	6	6	6

D.9: Table of raw data for Display-Speaker scheme obtained from ASQ1

User	Sat-Score1	Sat-Score2	Sat-Score3
1	5	4	6
2	5	5	6
3	5	4	5
4	5	4	5
5	5	4	6
6	5	4	6
7	4	4	5
8	5	4	5
9	5	4	5
10	5	4	6
11	4	4	5
12	4	4	5
13	6	5	5
14	5	4	5
15	5	5	5
16	6	4	5
17	6	5	6
18	6	5	5
19	4	4	5
20	5	4	5

D.10: Table of raw data for SIC scheme obtained from ASQ1

User	Sat-Score1	Sat-Score2	Sat-Score3
1	5	5	6
2	6	5	6
3	5	6	6
4	6	6	6
5	5	6	6
6	6	6	6
7	4	4	5
8	6	5	6
9	5	5	5
10	5	5	5
11	5	5	5
12	5	5	4
13	5	4	5
14	5	5	5
15	5	5	6
16	6	5	5
17	6	5	6
18	5	5	5
19	5	5	5
20	5	5	5

D.11: Table of raw data for CaS scheme obtained from ASQ1

User	Sat-Score1	Sat-Score2	Sat-Score3
1	6	5	5
2	6	6	6
3	6	6	6
4	6	7	6
5	6	5	6
6	6	6	6
7	5	6	5
8	6	6	6
9	6	5	6
10	6	5	5
11	6	6	5
12	5	5	5
13	5	4	5
14	6	5	5
15	6	5	6
16	6	6	5
17	6	6	6
18	6	5	5
19	5	5	5
20	6	5	5

D.12: Table of raw data for CoLoc scheme obtained from ASQ2

User	Sat-Score1	Sat-Score2	Sat-Score3
1	6	6	5
2	6	6	5
3	5	6	5
4	5	6	6
5	6	5	6
6	7	7	7
7	6	6	7
8	6	6	5
9	5	5	5
10	5	6	6
11	6	7	7
12	5	6	5
13	5	6	5
14	6	5	6
15	5	6	5
16	5	6	6
17	5	5	5
18	6	5	5
19	4	5	6
20	5	6	5