

**Simple and Secured Access to Networked Home  
Appliances *via* Internet using SSL, BioHashing  
and single Authentication Server**

Arpita Mondal

A Thesis

in

The Department

of

Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements  
for the Degree of Master of Computer Science at  
Concordia University  
Montreal, Quebec, Canada

July 2009

© Arpita Mondal, 2009



Library and Archives  
Canada

Published Heritage  
Branch

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

Bibliothèque et  
Archives Canada

Direction du  
Patrimoine de l'édition

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file* *Votre référence*  
ISBN: 978-0-494-63049-5  
*Our file* *Notre référence*  
ISBN: 978-0-494-63049-5

**NOTICE:**

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

**AVIS:**

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

---

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

  
**Canada**

# Abstract

## **Simple and Secured Access to Networked Home Appliances *via* Internet using SSL, BioHashing and single Authentication Server**

Arpita Mondal

This thesis describes a web-based application that will enable users to access their networked home appliances over the Internet in an easy, secured, accessible and cost effective manner, using the user's iris image only for authentication. As Internet is increasingly gaining significance and popularity in our daily lives, various home networking technologies also started gaining importance from consumers, which helped in facilitating interoperability, sharing of services and exchange of information between different electronic devices at home. As a result, the demand to be able to access home appliances or security cameras over the Internet gradually grew. In this research, we propose an efficient, secured, low-cost and user-friendly method to access networked home appliances over the Internet, providing strong, well integrated, three levels of security to the whole application and user data. According to our design, the user's iris data after hashing (using BioHashing) is sent through a secure communication channel utilizing Secure Sockets Layer v-3.0. The deterministic feature sequence from the iris image is extracted using 1D log-Gabor filters and while performing BioHashing, the orthonormalization of the pseudo-random number is implemented employing Gram-Schmidt orthonormalization algorithm. In addition to this protected data transfer mechanism, we propose the design of an Authentication Server that can be shared among multiple homes, allowing numerous users to access their home appliances in a trouble-free and secured manner. It can also bring down the cost of commercial realization of this endeavor and increase its accessibility without compromising on system security. We demonstrate that the recognition efficiency of this system is computationally effective with equal error rate (EER) of 0% and 6.75% (average) in two separate conditions on CASIA 1 and CASIA 2 iris image datasets.

# Acknowledgements

This thesis would not have been accomplished without the love, support, guidance and encouragement of multitude of individuals. I would like to begin by thanking my parents, even though I understand that any amount of gratitude shown to them is woefully inadequate. My father's unconditional support is largely the reason that this masters is completed in Canada. No words are sufficient to describe my mother's contribution to my life. I owe every bit of my existence to her. This thesis is dedicated to both of them. My sister has always added a sweet and sour touch to my life whenever life was becoming bland. I owe a lot of smiles to her. I would also like to thank Chandu for keeping me entertained for all these years with his prank stories.

I am immensely indebted to my advisor, Prof. Prabir Bhattacharya. He has influenced not only my graduate studies, but my whole life. He has instilled in me by example, a strong sense of discipline, principle and integrity, for which I am eternally grateful to him. Prof. Bhattacharya is a deeply committed researcher, teacher, and advisor and has helped me grow with his invaluable advice, guidance, support and criticism. It is because of him that my graduate studies in Concordia have been so enjoyable and rewarding. I feel myself privileged for being able to work under his experienced supervision towards accomplishing this research successfully.

I am also grateful to my elder brother Mr. Ranjan Biswas for believing in me, supporting me and helping me in every way he could for fulfilling my dreams. It is only because of him that my graduate study was feasible.



I would like to thank all my lab members for their suggestions and help in ways not only relating to my research but also in other ways that enriched my experience in Montreal. I am especially thankful to my two lifelong friends, Arghya Paul and Samrat Jha for being two strong pillars in my life, sharing my happiness & sorrows, tensions & concerns at all times, and above all, bearing with me through all my idiosyncrasies.

Lastly, I would like to thank Concordia University, for selecting me for Concordia University International Tuition Fee Remission Award twice, which not only encouraged me to work harder, but also helped me immensely to concentrate completely on my research and perform well, without having to worry much about my finances. I am lucky and proud to be a graduate student in Concordia University, Montreal.

**Arpita Mondal**

**21<sup>st</sup> July, 2009**

**Dedicated**  
**To My Parents**

# Table of Contents

**List of Figures** ..... x

**List of Tables** ..... xiv

**List of Acronyms** ..... xv

**Chapter 1:INTRODUCTION** ..... 1

1.1. Motivation ..... 1

1.2. Example Scenario ..... 2

1.3. Applications ..... 4

1.4. Related Work ..... 7

1.5. Proposed Architecture ..... 10

    1.5.1. SSL Communication Channel ..... 12

    1.5.2. Biometric Authentication ..... 13

    1.5.3. Authentication Server ..... 19

    1.5.4. Home Networks ..... 21

    1.5.5. Session Initiation Protocol ..... 22

1.6. Objective of the Research ..... 22

1.7. Original Contributions to the Knowledge ..... 24

1.8. Organization of the Thesis ..... 25

**Chapter 2: SECURE COMMUNICATION CHANNEL** ..... 27

2.1. An Overview ..... 27

    2.1.1. Secure HTTP ..... 29

2.1.2. Virtual Private Network .....	31
2.2. Secure Sockets Layer .....	33
2.3. Secure Bookmark .....	39
<b>Chapter 3: BIOMETRIC AUTHENTICATION</b>	<b>40</b>
3.1. An Overview .....	40
3.1.1. Iris for Biometric Recognition .....	42
3.1.2. Issues in Iris Recognition .....	44
3.2 Related work in Iris Recognition .....	45
<b>Chapter 4: HASHING</b>	<b>50</b>
4.1. An Overview .....	50
4.2. Potential threats to Biometric data .....	51
4.3. Methods to safeguard attack points .....	54
4.3.1. Image Hashing .....	57
4.3.2. BioHashing .....	59
4.3.3. S-Iris .....	62
4.4. Proposed BioHashing Method .....	64
4.4.1. Feature Extraction from iris images .....	65
4.4.2. Randomization using pseudorandom number .....	71
4.4.3. User and Authentication Server Interaction Scenarios .....	78
<b>Chapter 5: AUTHENTICATION SERVER</b>	<b>80</b>
5.1. An Overview .....	80
5.2. Home Gateway .....	81
5.3. Popular Home Servers .....	83
5.3.1. Home Server .....	84
5.3.2. Internet Home Server .....	87

5.3.3. Remote Management Server .....	89
5.4. Proposed Authentication Server .....	92
5.5. Session Initiation Protocol .....	97
<b>Chapter 6: HOME NETWORKS</b> .....	<b>100</b>
6.1. An Overview .....	100
6.1.1. Jini .....	102
6.1.2. HAVi .....	105
6.2. Universal Plug and Play .....	107
<b>Chapter 7: RESULTS &amp; DISCUSSIONS</b> .....	<b>114</b>
7.1. An Overview .....	114
7.2. Client Server Program .....	115
7.2.1. Client Program (SSLClient.c) .....	117
7.2.2. Server Program (SSLServer.c) .....	121
7.2.3. Client and Server Certificate .....	123
7.3. Performance Evaluation .....	126
7.3.1. Support Vector Machine (SVM) .....	126
7.3.2. Hausdorff Distance .....	130
7.4. Security Analysis .....	137
7.5. Conclusions .....	139
7.6. Contributions to the Knowledge .....	141
7.7. Future Research .....	142
<b>REFERENCES</b> .....	<b>145</b>
<b>APPENDIX A</b> .....	<b>156</b>

# List of Figures

1.1. Accessing chemical reactors (indoor) of an Industrial Plant via Internet	4
1.2. Accessing various reactors (outdoor) of an Industrial Plant via Internet	5
1.3. Accessing multiple instruments inside an Industrial complex via Internet .....	6
1.4. Proposed Architecture - Authentication Server (AS) shared among multiple residences with Home Gateway (HG) in each residence .....	10
1.5. Steps performed in our architecture to enable secure access to home appliances via Internet .....	11
1.6. Process of enrolment and authentication in a generic biometric authentication system .....	17
2.1. Virtual Private Network (VPN) operating over the Internet .....	31
2.2. SSL handshaking procedure .....	36
3.1. Sample biometric traits: (a) fingerprint, (b) face, (c) palmprint, (d) signature, (e) iris and (f) voice .....	41
3.2. Structure of an eye .....	42

3.3.	Sample picture of an iris .....	43
3.4.	Samples of iris images from CASIA 1 and CASIA 2 iris image databases ..	44
4.1.	Points of attack in a typical web application that requires user authentication	51
4.2.	Block diagram of a two stage image hashing system .....	61
4.3.	A generic hashing procedure where the iris image is binarized using a biometric hashing scheme .....	62
4.4.	Procedure followed for performing BioHash of iris image submitted by the user .....	63
4.5.	User U and Authentication Server S interaction diagram (USID) during the user authentication process .....	64
4.6.	Iris Image preprocessing on CASIA 2 dataset .....	68
4.7.	Unwrapping and enhancement of an iris image on CASIA 2 dataset .....	70
4.8.	Procedure of new user registration in our proposed architecture .....	73
4.9.	Steps performed during user authentication using our proposed .....	74
4.10.	The process of Random Number generation during the actual user authentication process .....	75
5.1.	Access to Home Network using a Home Gateway .....	82
5.2.	A traditional home network including both a home server and a home gateway .....	85

5.3.	A virtual home using Internet Home Server (IHS) .....	87
5.4.	Remote authentication using Remote Management Server (RMS) .....	89
5.5.	Authentication Server (AS) shared among multiple residences with Home Gateway (HG) in each residence .....	93
5.6.	Client - Authentication Server interaction diagram and finally, access to the Home Gateway (HG) .....	94
6.1.	Architecture of Jini .....	103
6.2.	Communication in a HAVi network .....	106
6.3.	UPnP high level architecture .....	108
7.1.	Common.h program – a header file .....	116
7.2.	Common.c program – contains common functions between the client and server program .....	117
7.3.	A snapshot of SSLClient.c program .....	118
7.4.	A snapshot of SSLServer.c program .....	121
7.5.	Certificate Request generated .....	124
7.6.	Server Certificate with both the private key and public key .....	125
7.7.	SVM: Selection of optimal values of 1D log-Gabor parameters in CASIA 1 image dataset .....	127
7.8.	SVM: Selection of optimal values of 1D log-Gabor parameters in CASIA 2	



image dataset .....	127
7.9. SVM: ROC curve shows the comparison between GAR and FAR on CASIA 1 iris image database .....	128
7.10. SVM: ROC curve shows the comparison between GAR and FAR on CASIA 2 iris image database .....	129
7.11. Selection of optimal length of BioHash code in CASIA 1 iris image dataset .....	130
7.12. Selection of optimal length of BioHash code in CASIA 2 image dataset ....	131
7.13. (Distance matching) Selection of optimal values of 1D log-Gabor parameters in CASIA 1 image dataset .....	132
7.14. (Distance matching) ROC curve for BioHash on CASIA 1 image dataset ...	133
7.15. (Distance matching) Selection of optimal values of 1D log-Gabor parameters in CASIA 2 image dataset .....	134
7.16. (Distance matching) ROC curve for BioHash on CASIA 2 image dataset ...	135
7.17. (Same random number) ROC curve for BioHash on CASIA 1 image dataset .....	136
7.18. (Same random number) ROC curve for BioHash on CASIA 2 image dataset .....	136

## List of Tables

4.1. Different types of attack on Application data in Client-Server architecture .....	52
4.2. Variables used in the User interaction with Server during authentication .....	76
7.1. Interaction between the Client and Server program exchanging UserID	119
7.2. Checking the Server Certificate in the Client program (SSLClient.c) .....	120
7.3. Client authentication in the Server program (SSLServer.c) .....	122
7.4. Checking the Client Certificate in the Server program (SSLServer.c) .....	123

## List of Acronyms

AACS	.....	Advanced Access Content System
AES	.....	Advanced Encryption Standard
API	.....	Application Programming Interfaces
AS	.....	Authentication Server
B2B	.....	Business-to-Business
B2C	.....	Business-to-Consumer
CA	.....	Certificate Authority
CASIA	.....	Chinese Academy of Sciences - Institute of Automation
CBIR	.....	Content Based Image Retrieval
DCT	.....	Discrete Cosine Transform
DES	.....	Data Encryption Standard
DHCP	.....	Dynamic Host Control Protocol
DHCP	.....	Dynamic Host Control Protocol
EER	.....	Equal Error Rate
FAR	.....	False Accept Rate
FRR	.....	False Reject Rate

FTP	.....	File Transfer Protocol
GAR	.....	Genuine Acceptance Rate
HAVi	.....	Home Audio Video interoperability
HG	.....	Home Gateway
HMAC	.....	Hashing for Message Authentication Code
HomePNA	.....	Home Phonenumber Networking Alliance
HomeRF	.....	Home Radio Frequency
HS	.....	Home Server
HTTP	.....	Hypertext Transfer Protocol
HVAC	.....	Heating Ventilation And Air Conditioning
IHS	.....	Internet Home Server
IP	.....	Internet Protocol
IPSec	.....	IP Security
ISP	.....	Internet Service Providers
Java RMI	.....	Java Remote Method Invocation
L2TP	.....	Layer 2 Tunnel Protocol
MAC	.....	Message Authenticity Check
NIC	.....	Network Interface Card
NLPR	.....	National Laboratory of Pattern Recognition
OTP	.....	One-Time-Password
PPTP	.....	Point to Point Tunneling Protocol
QoS	.....	Quality of Service
RG	.....	Residential Gateway

RMS	.....	Remote Management Server
ROC	.....	Receiver Operator Characteristics
SHTTP	.....	Secure Hypertext Transfer Protocol
SIP	.....	Session Initiation Protocol
SMTP	.....	Simple Mail Transfer Protocol
SSL	.....	Secure Sockets Layer
SVM	.....	Support Vector Machines
TCP/IP	.....	Transfer Control Protocol/Internet Protocol
TLS	.....	Transport Layer Security
TRN	.....	Tokenized Random Number
UAC	.....	User Agent Client
UAS	.....	User Agent Server
UPnP	.....	Universal Plug and Play
URI	.....	Uniform Resource Identifier
URL	.....	Uniform Resource Locator
VPN	.....	Virtual Private Network
WAP	.....	Wireless Application Protocol

## **Chapter 1**

# **Introduction**

### **1.1. Motivation**

Moving towards an 'always-on', 'mobile' future and a technology driven lifestyle, people are demanding greater technical triumphs to make life more exciting, expedient and convenient. Automation at home has already started catering to this growing need. With increasing number of manufacturers developing appliances that can be controlled remotely (popularly known as smart appliances), almost every home now owns multiple intelligent, network enabled appliances like fridge, microwave, light dimmers, heating system, air conditioners, dishwasher, window shutters, TV, music systems, computers, etc. Moreover, as broadband connectivity in homes got affordable, different home networking technologies started gaining popularity and a lot of attention from researchers all over the world. As a result, the desire to control these appliances from remote locations over the widely available Internet gained momentum.

The major motivating factor for remote access to home was the possibility of greater control of the appliances inside the house without being physically present. Closing the window shutters at home from a distant holiday resort or starting the

room's heating system before coming back from office is not a luxury anymore, but a consumer demand. Additionally, the reduction in the cost of these information appliances after the discovery of microcontrollers (used in developing these appliances) is a major encouraging reason for driving home automation [1]. Microcontrollers revolutionized the way control systems performed earlier, by markedly reducing power consumption and improving performance, reliability, stability and energy efficiency. In addition to these cheaper smart appliances, the prospect of higher energy efficiency increased the usability and practicality of this endeavor and hence, boosted research in this domain. In the following section, *Section 1.2*, we discuss more about the various applications of this research and then in *Section 1.4*, we explain the exploration already done in each of the constituent fields of this topic.

## **1.2. Example Scenario**

We describe a basic scenario to demonstrate what our research translates to in real life. This scenario demonstrates the capabilities of this architecture as well as the ease with which this endeavor can be used by a general user.

Sam wakes up at 7 AM and gets ready to catch a flight as he is going for a short vacation with his friends. By the time he got ready, all the window blinds opened automatically in every room and the coffee maker announced through the music system in his drawing room that it has finished brewing a fresh cup of coffee for him in the kitchen. At 9 AM, his music system declares that he has only two hours left for his flight, while a little later, his television turns on and tells Sam about

the weather forecast for that day and the traffic condition of highway 40 that he will be taking to go to the airport. As Sam steps out of the house, the security system turns on in the vacation mode, where it has to send daily summary to Sam on his mobile about the status in his home. It also updates the emergency numbers to dial in case of any emergency during Sam's absence.

But in a hurry, Sam forgets to close one of his bedroom windows and turn off the air conditioner. He also forgets to start the dishwasher before he left home as he had stuff to clean. But no worries! Sam reaches the airport terminal within 20 minutes and gets online on the Internet using his laptop. He accesses his home with just a click on the browser and sends the command to close the bedside window that he has forgotten to close, shut all the window blinds and also, start the dishwasher (stuffed with used utensils). He also checks if any other device is turned on or not as in that case, he would shut it down over the Internet. Every day, during the vacation, Sam checked online if everything in his home is fine or not. He also listened in to the voice messages on his home phone for any important message.

While coming home from vacation, Sam accessed his home's appliances once more from the airport terminal and started the air conditioner and the coffee machine. He also checked his fridge to see if he has enough food or not and then, turned on his porch lights. As soon as he reaches home, he gets a perfectly conditioned cool home and a fresh cup of coffee waiting for him!

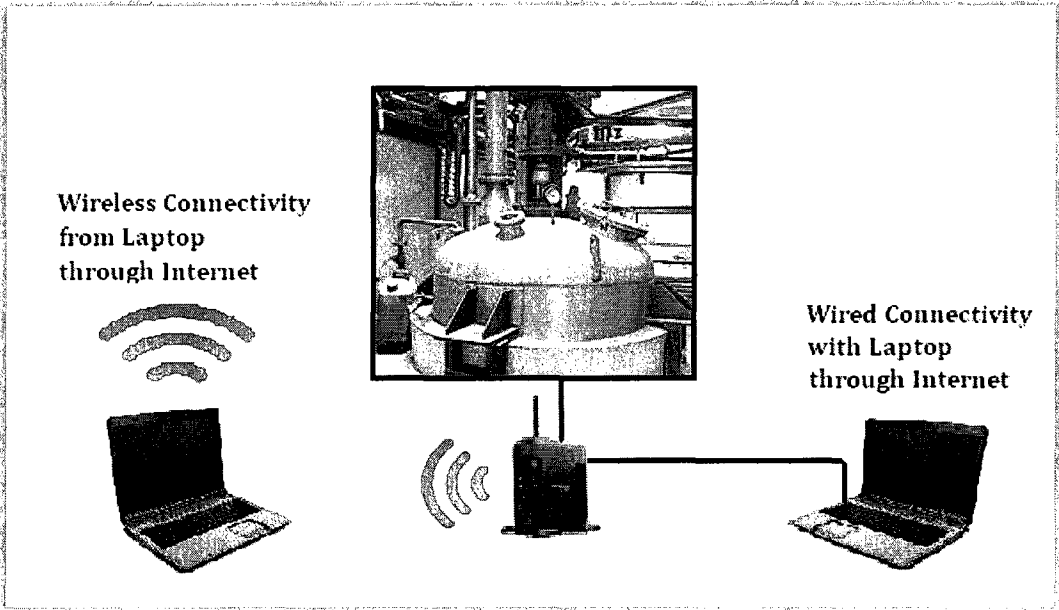
This example scenario motivates a number of requirements on the design of the application. Firstly, all the appliances in the home must be able to communicate with each other and be able to use each other's services in order to provide better



user experience. Secondly, the method of remote access to home appliances must be very secure as otherwise; a hacker can get complete control of a person's home and create havoc. Thirdly, the method to see the status of each appliance on the web must be easy and starting it up or shutting it down should be uncomplicated.

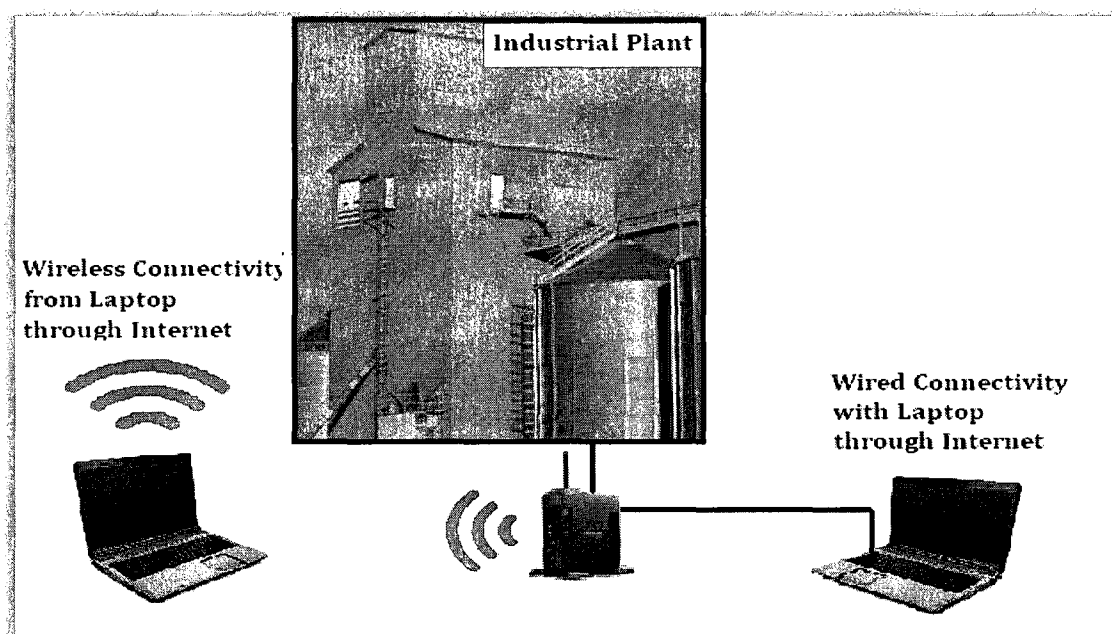
### 1.3. Applications

This research focuses on the control of home appliances remotely via Internet. However, the same architecture can also be used to control industrial furnaces, nuclear reactors, explosive chemical reactors, blast furnaces, security cameras, etc. present in both indoor and outdoor areas, from distant locations over the Internet in a secured way (see Figure 1.1 and Figure 1.2). Our proposed architecture enables remote control of these systems or instruments, adequately



**Figure 1.1:** Accessing chemical reactors (indoor) of an Industrial Plant via Internet

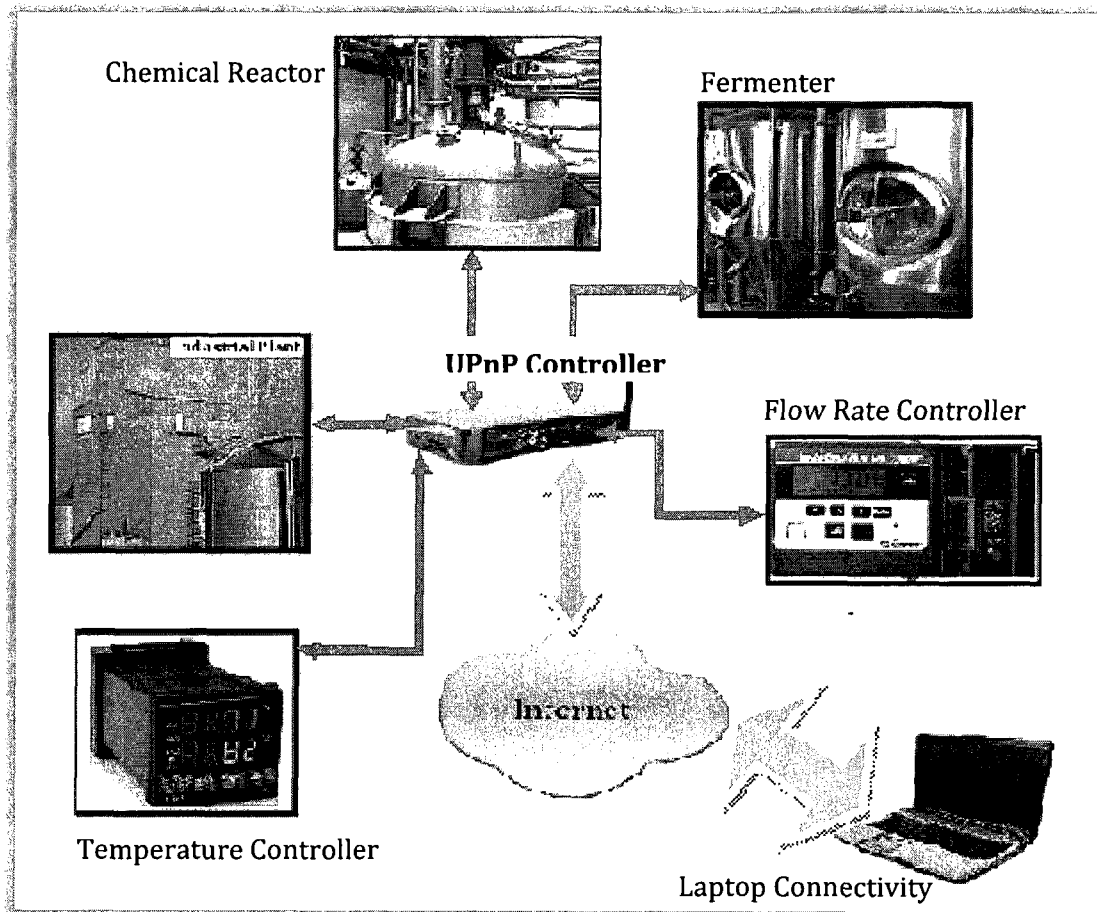
satisfying the security requirements that can be associated with these devices for operating on it. As in a model situation shown in Figure 1.1, an individual may try to activate a high-risk chemical reaction in a reactor inside an industrial complex, remotely from a distant office using our architecture. Since it is a very costly and risky apparatus to work on, in reality, only a few selected, knowledgeable individuals will be given access to it physically.



**Figure 1.2:** Accessing various reactors (outdoor) of an Industrial Plant via Internet

Therefore, the same level of security needs to be maintained while accessing it over the web; otherwise it will become unrealistic to implement such remote access. Our architecture provides three-tiered security beneficial for such situations and hence, enables secured and smooth, remote control of such instruments. Moreover, with this technique, it is possible for a single person to monitor multiple instruments or reactions occurring in different locations efficiently and securely

with the help of other devices if required, like monitoring cameras, thermometers, pressure gauges, etc. depending on the equipment being accessed. This system can also perform well in working with bio-hazard materials or in hazardous environments as it can successfully reduce the life risk of people working on it. Being positioned in a remote location with complete knowledge of the internal mechanism and state of an unsafe reaction will cause minimum damage to human life and improve safety of the whole machinery.



**Figure 1.3:** Accessing multiple instruments inside an Industrial complex via Internet

In an advanced networking design, multiple instruments in an industrial complex can be connected together to form a network and then be accessed over the Internet, using both wired and wireless connectivity. This situation is depicted in Figure 1.3, in which, multiple instruments such as chemical reactors, fermenters, bioreactors, flow rate controllers, temperature controllers, etc. are connected to a Universal Plug and Play (UPnP) controller to form a private UPnP network that is accessible over the Internet.

#### **1.4. Related Work**

There have been many attempts over the years to develop remote access to home appliances via Internet, but most of them failed because of accessibility, reliability, feasibility, high implementation cost, and security problems. This endeavor comprises of many dissimilar components that needs to be wisely selected and properly integrated in order to facilitate interoperability, achieve scalability and be usable. A lot of research has been done independently in each of the related fields like secure transfer of data, user authentication in web environment, home networking, interaction between smart appliances, message exchange using home gateway, etc. but little has been done for the realistic integration of all these domains as required in this project.

The very first attempt to realize part of this design was made using a home server that was placed inside a home [2], where a user can access an appliance connected to this home server over the Internet. The home server was responsible

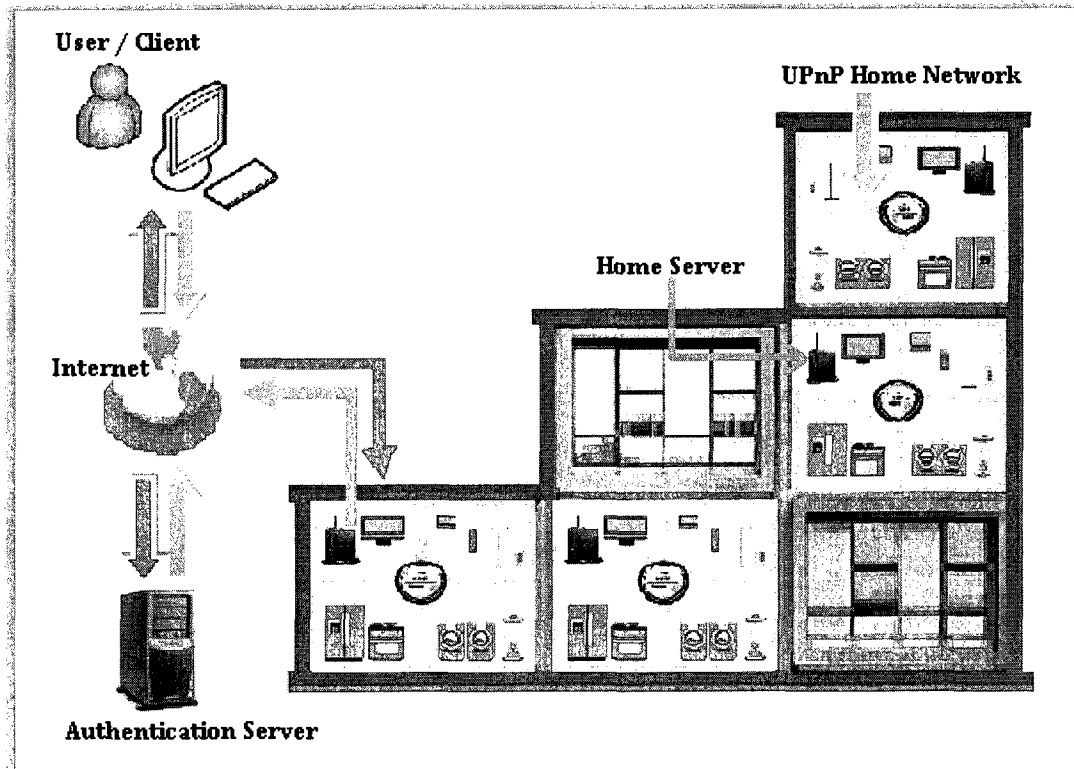
for connecting and managing all the information appliances present in the home [2], [3], [4] and directing the user commands sent through the Internet to the target appliance in a language comprehensible to it. But there were no provision for user authentication, or security for the whole interaction. Moreover, each apartment was required to install this home server [1], [5], [6] in order to implement home automation, which increased the installation cost of this endeavor and decreased its practicability restricting its large scale implementation. Few years later, a group of researchers made a promising implementation [2] of accessing home appliances with Internet Home Server (IHS) that solved most of the problems encountered in the first home server. They successfully controlled a washing machine from LG company remotely over the Internet in 2002 [2], [6]. But IHS had many unresolved fallouts like requiring installation of one server per home, increased user responsibility for maintaining the server, etc. that rendered it expensive and unfit for extensive, commercial use [6]. Then came the Remote Management Server (RMS) which was proposed in a similar milieu of accessing home security system [6] that was able to solve few of the problems faced in earlier attempts in home automation. But it could not solve the problem of dynamic allocation of Internet Protocol (IP) address to Home Gateways, for which, it failed to become popular among consumers. Simultaneously, by the year 2000, a number of researchers started working towards integration of intelligent devices and the medium that can be used to connect them, like wireless medium, power line, Ethernet, dedicated bus, etc. [2], [4], [7], [8], [9]. During the same time, another initiative started towards extending the control of home appliances in the home networks beyond the

boundaries of the home to mobile networks [10]. It was done using a HAVi-WAP (Home Audio Video interoperability, Wireless Application Protocol) User Interface gateway that communicates between a wired home entertainment network and a wireless mobile communication network using HAVi and WAP designs respectively [10]. Strikingly, most of these implementations did not consider security in their system architecture and did not implement any user verification or privilege determination during access to the home networks.

Since applications on the Internet are accessible to anybody connected to it irrespective of his/her motive, security threats to this application can come from any individual from any part of the globe. As a result, security of this application should be a primary concern and needs to be strong enough to be able to thwart diverse kinds of malicious attacks. In this research, our primary objective was to develop a web application that will enable user access to their home appliances via Internet in a very secure but simple way, with minimum user responsibility and minimum installation cost to configure the system. We also aimed to develop it in such a way that it needs least or no professional knowledge to update, modify and maintain it. Here, with the term 'security of a web application', we refer to the methods used or required to provide authenticity, confidentiality, integrity and availability of data, to and from the application on the Internet. In the next sections, we mainly talk about the proposed architecture and give a brief overview of the components of this design.

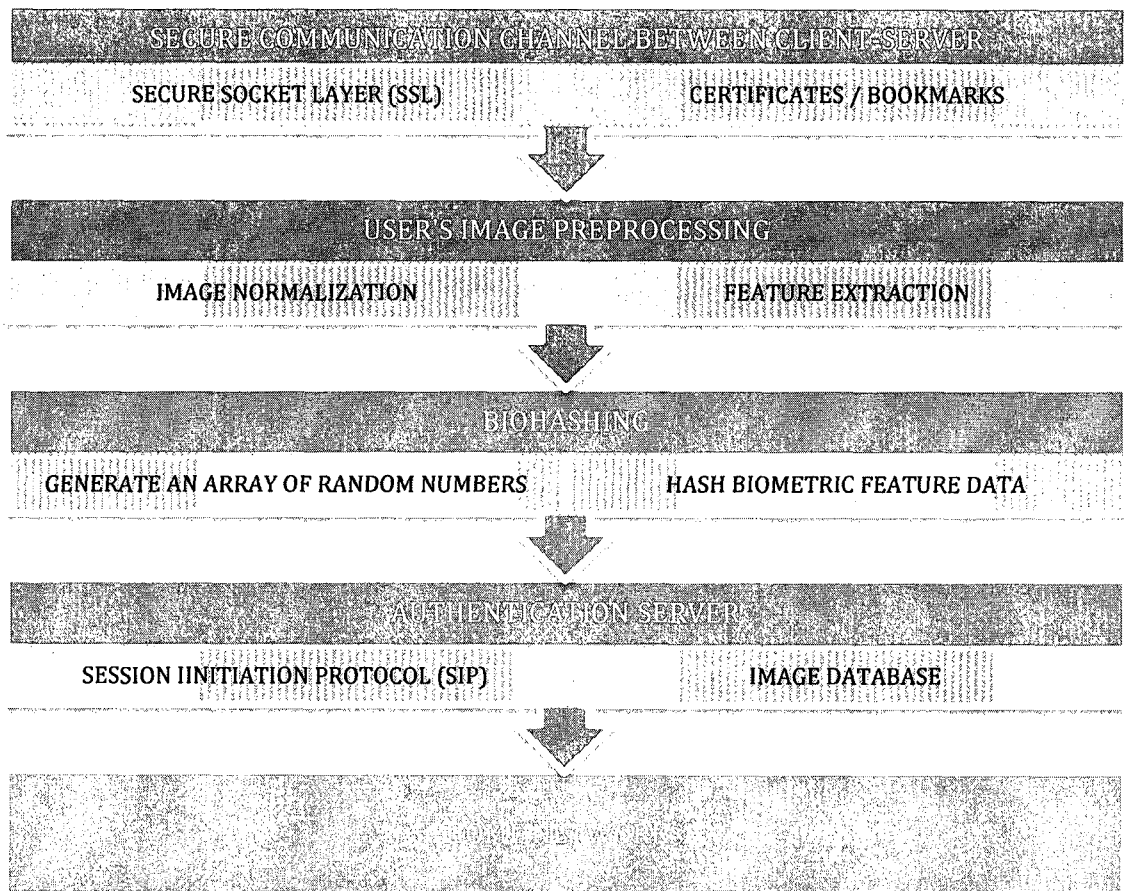
## 1.5. Proposed Architecture

We illustrate the proposed architecture in five separate sections, describing one component in each segment, all of which integrates together to safeguard the whole application in three different stages. In this section (*Section 1.5*), we give a brief overview of the modules we have used to develop this thin client web application, in the order in which these features are utilized in the system. We constructed a secure communication channel between the user terminal and the home gateway using Secure Sockets Layer (SSL v-3.0) initially using certificates and later utilizing bookmarks, which is more convenient to use. In order to authenticate



**Figure 1.4:** Proposed Architecture - Authentication Server (AS) shared among multiple residences with Home Gateway (HG) in each residence

a user, we use biometric authentication via iris, which mainly involves initial image normalization steps and its corresponding feature extraction for which we used 1Dlog-Gabor filters. Then we perform hashing of the biometric feature data using a novel method so as to increase the system's usability and simultaneously, minimize user responsibility without compromising on the security of the whole architecture. We employ hashing of the biometric data in order to avoid transmitting the raw data through the communication channel and extend the security of the whole system.



**Figure 1.5:** Steps performed in our architecture to enable secure access to home appliances via Internet



We developed a novel design of sharing the user-verification server, as shown in Figure 1.4, within multiple apartments, which successfully can bring down the implementation cost and individual user responsibility in maintaining it by manifolds. Along with this shared authentication server, we use Session Initiation Protocol (SIP) as a way to determine the variable IP address of the connected homes and enable user access to home networks possible (see Figure 1.5). Lastly, we talk about Universal Plug and Play (UPnP) home network, which is the preferred home network for this system because of its plug and play characteristic and dynamic device discovery, which helps in increasing the accessibility of this architectural solution.

### **1.5.1. SSL Communication Channel**

Secure Sockets Layer (SSL) is a lower layer protocol, a layer between the Transport and Application layer, which can be used with all other protocols without the user's knowledge. We have used Secure Sockets Layer version 3.0 [11] to protect the data that is transferred between the client and server. SSL works on top of http and its certificate helps to reinstate the authenticity of the server and if needed the client also. Each SSL certificate is verified and signed by a trusted third party (called Certificate Authority, CA) who confirms that the certificate owner is really the one it claims to be. This certificate contains unique verified information about the certificate owner (verified by a CA) like email address, owner's name, certificate validity period, etc [12]. After both the client and server have verified the certificates, SSL establishes a session key that is used to encrypt all the data

exchanged between them. However, if not implemented properly, SSL is prone to man-in-the-middle attack [13], [14] in which a man (a terminal in reality) sits in between the client and server and poses to be a client to the server and a server to the client, and in this way, manipulates or listens to the messages being exchanged between the real client-server. To prevent this attack, we have used customary verification of certificates for both client and server, where the client certificates should be delivered to the registered clients during or after user registration.

After an SSL connection has been established between the client and the server post certificate verification, the client sends his/her User ID or username to the server, which is checked with the server's database to see the existence of that user, after which, the server asks the user to send his/her biometric data (if the user account exists in the server database). The username can be obtained during the registration of a user with the server, whose process is discussed in the next section.

### **1.5.2. Biometric Authentication**

With recent improvements in security measures in almost all applications, personal identification systems founded on biometrics have also undergone an enormous growth in technology and awareness [15], [16], [17], [18]. Gradually, biometric authentication is becoming a more sought-after means of authentication for authenticating one's identity, for both in-place and remote authentication. One of the major reasons for its popularity can be contributed to its enhanced security as compared to passwords (which can be forgotten, stolen, guessed or even hacked) and possession-cards like Smart Card, SecureID, Biometric ID, etc. (which can in

turn be stolen, replicated and tampered) [13], [19], [20]. Instead of identifying somebody based on what he/she possesses (like card or key) or what he/she knows (like passwords), biometric authentication verifies the person based on who he is and thus is a better and more meaningful way to authenticate a person.

Biometrics is used to identify a person both using his physical and behavioral characteristics. The various biometric technologies that are available today comprise physiological traits like iris, retina, face, fingerprint, palm print, hand geometry, etc and behavioral traits like voice, signature, handwriting, gait, etc. Since biometric data cannot be captured to furnish the same data twice, one can never do an exact matching even between two exceedingly similar biometric data, and so, it's matching is based on fuzzy comparison or distance measures. Among the different biometric traits available for authenticating an individual, iris is the most popular and accurate method of identification due to its unobtrusiveness, permanence, minimum user discomfort while measuring and uniqueness [15], [16], [17], [19], [21], [22]. It is also well placed within the natural shield of the eye, which protects it very securely and prevents alteration by injury. In fact, it is unique enough to be able to distinguish between two twins, who will have different iris footprint and thus are differentiable [19], [23]. It has been calculated that two iris minutiae would be identical at approximately 1 in  $10^{52}$  cases [19]. In addition to the above stated reasons, iris verification is comparatively more suited than face recognition as it does not compromise the religious constraints involved while capturing a woman's face in many countries. Moreover, most of the computers and laptops nowadays come with inbuilt camera, a satisfactory image capturing device that removes the

need of any specialized equipment, thus minimizing cost for using our application. Fingerprint and palm prints do avoid the above mentioned problem encountered during face recognition, but they necessitate specialized readers to capture its data, which increases the cost for using this application and also reduces its acceptance among consumers. Among the behavioral features, hand and finger geometry are a good way to authenticate a person. But they are not distinct enough to be used in this application and like fingerprint based recognition, requires additional devices to acquire its data. Same problem is faced with handwriting analysis and so, after much consideration, we settle to implement iris recognition in this application.

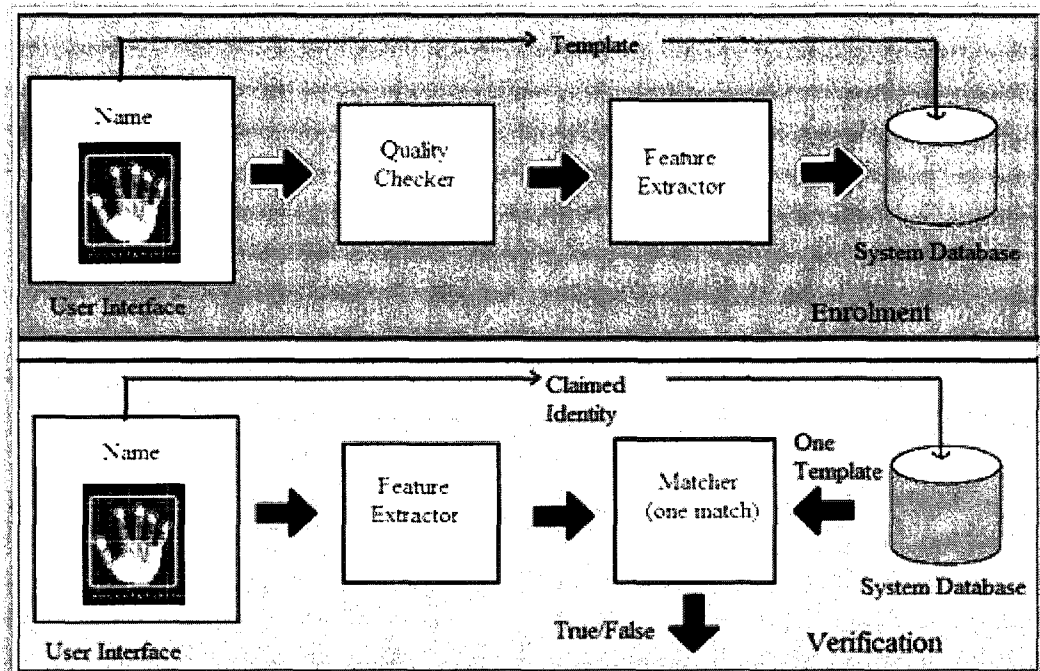
For improved performance, during the complicated process of biometric authentication, we apply verification of a user rather than his/her identification. Authentication is the scheme by which we can check whether the user is actually the person who he/she claims to be; identification is the method by which, we can compare the information given by the user with information on all other users and try to find the perfect match and thus spot the user. For the latter, the user is identified without him trying to claim any identity [24]. Thus we see, that, authentication involves one-to-one comparison between two different data, whereas, identification involves one-to-many comparisons between varying number of data. In order to implement user authentication in this application (and not user identification), we require only one additional message (the user's User ID) to be sent from the user to the server, which we do not need when implementing identification. But as discussed above, it takes more time and complexity to find a close match between the submitted image and all the stored images in the server

(identification), than verifying the submitted image of user  $U$  with the stored image corresponding to  $U$  (verification or authentication). This time and complexity overhead compared to an extra message sent between client and server is far more severe and hence, we use 'authentication' of a user rather than his/her 'identification'.

An authentication system based on biometric takes the following three steps to be functional [24], [25] which is also the same for our application. They are listed as follows:

- a) **Enrolment or Registering User** – In order to be able to use the services of biometric-based authentication system, a user has to enroll himself/herself with the server at the very beginning. This enrollment procedure involves the user submitting information like name, preferred user id, email address, home address, number of appliances at home, etc. to the server.
- b) **Storage of biometric data** – This involves the user entering his/her iris data with the system for storage, with the intention that, this data can be later used as his/her reference data and afterwards compared with iris images captured during user authentication of the user requesting access to the application.
- c) **Comparison** – This phase is used in order to measure the extent of similarity between the stored iris image in the database corresponding to a particular user and the submitted image of a user, who claims to be the same registered user in order to enable user verification.

In this application, every user will need to go through the above process in order to register themselves with the server and later authenticate oneself via biometric authentication safely while accessing their home appliances over the Internet.



**Figure 1.6:** Process of enrolment and authentication in a generic biometric authentication system

### 1.5.3. Hashing

As already described, the user will have to send his/her iris image to the server in order to let the server compare the user submitted image and the one stored in its database corresponding to that particular user (recognized by his/her User ID); then decide the authenticity of the user. Thus in this architecture, user's biometric (iris) data needs to be sent to the server over the widely accessible Internet. Pilfering of this iris data can cause major crimes like identity theft, bank

fraud and terrorism, etc. and hence, this data requires secured transfer over the wire. Using simple data encryption techniques can secure this data in transit; however, its strength depends mostly on the length of the keys used. Furthermore, if an attacker by some means compromises the encryption keys, he/she will get direct access to the raw biometric information of the genuine user that he/she can use in future to craft unauthorized access in similar biometric authentication based applications. In order to avoid this replay attack, hash functions can be used on this biometric data ( $B$ ) to produce a hash value ( $f$ ), which can then be transmitted to the server for authentication. Evident from the mechanism in which hash function works, the original data can never be extracted from the hash value, which is produced after executing hash on the input data. This removes the threat of  $B$  getting exposed in the hands of an adversary if he is able to decode  $f$  by any means. Common hash functions like MD5, SHA-1 are very sensitive to even 1 bit change in input data [25-29] and hence, unfit for use in this context, as biometric image data can never be the same twice (picture taken during user registration and authentication). Consequently, for our application, we use a novel, user-friendly and robust hashing technique that is a modified version of image hashing technique, which is sensitive to the iris image's visual appearance and not susceptible to changes in illumination, darkness, magnification or reduction, etc. [28]. This hash function is able to produce similar hash value for perceptually identical iris images and different hash value for visually dissimilar iris images [28], [30] with a very high accuracy as discussed later in the results section.

Along with this hash function, we propose the use of a non-secret hash algorithm, thereby stressing on the randomness of the value/token and not on the secrecy of the algorithm. A separate secret key is used for each user registered to access home appliances that is stored in the database of the authenticating server, and not entrusted on the client to present it during authentication. When a user submits a valid username into the web application, this unique number is sent to the client terminal after a small randomization procedure that prevents any replay attack. Thus, at both the client and server ends, the same random number is used to generate randomness in the hash value and then perform matching between the two to determine the authenticity of the user.

### **1.5.3. Authentication Server**

A key element that directs appliance control requests from the Internet to the appropriate home appliances in a language comprehensible to the devices is the home server and Internet home server [2]. Internet Service Providers (ISPs) use Dynamic Host Control Protocol (DHCP) to allocate IP addresses to hosts connecting to the Internet. Every time a home gateway installed in a home boots up or connects to the Internet, it typically receives a new IP address from the ISPs that is selected randomly from their unallocated IP address pool. This dynamic IP address of the home gateway causes a major problem in identifying a particular home from the Internet and this is what made the home server unusable. Moreover, till date, all the home servers had to be installed on a per home basis, thus increasing the cost of



implementation. As a result, a better and more accessible design was necessary to implement automation at home via Internet.

We propose the design of a novel home server called 'Authentication Server' (AS) that is used together with Session Initiation Protocol (SIP) that operates between this AS and home gateway. An AS is responsible for authenticating users, provide access control information, quality of service, and for controlling home appliances in multiple residences similar to an apartment/housing complex. The client communicates with the AS till it is authenticated. On successful completion of authentication, the AS retrieves the IP address of the corresponding home's gateway using SIP, and then transfers the connection to the client application and home gateway, thus freeing itself for taking new connections from other users. The client and home gateway now transmits command messages through small, encrypted packets using SIP and thus harnesses the advantages of an event driven protocol. This is done by the client closing the connection with AS after it receives the IP address of the home gateway and then, connecting directly to this new address which is already open to receive connections (home gateway).

Since this AS will store iris image data for several users for authentication, the database will be large and a professional can be employed to maintain it, take encrypted backups [31] and carry out necessary updates after proper approval/request from the users. The decision to encrypt the whole database or only the sensitive columns containing user's iris data is a design issue and will depend on the particular implementation and out of scope of this research. All the apartments using it can effectively share the cost of buying, installing and

maintaining it, relieving any individual user from spending huge amount of money, time and energy on this. Thus this proves to be a user-friendlier, cost-effective and safer implementation of remote home automation.

#### **1.5.4. Home Networks**

Home appliances can be connected to each other in various ways like using power line, phone line, Ethernet, wireless, USB, Firewire, etc. [4], which enables them to connect to the Internet and even share information and communicate with each other. The diverse interconnections standards for today's home networks include Bluetooth, HAVi, UPnP, Jini, OSGi, HomeAPI, X-10, HomePNA, IEEE 802.11b, and many more [4], [8], [32]. Among these, we propose using UPnP (zero-configuration networking protocol) in our architecture. It is a distributed, higher layer protocol built over the simple standards of TCP/IP, XML, HTML, DNS, LDAP and HTTP [4], [8]. It allows appliances to join dynamically without prior configuration and then declare its description and capabilities to the control points in the network. It allocates the appliances a permanent IP address (using DHCP server) or a temporary address (using Automatic IP protocol) [8] depending on the need. When user's command messages are sent through SIP, the former is able to harness the security capabilities of the latter and still establish a very flexible home network. Additionally, UPnP home network can be built over most physical media like wired and wireless network, phone line [33], power line, etc. which makes it a favorite choice for many.

### **1.5.5. Session Initiation Protocol**

One of the major reasons we need to use Session Initiation Protocol (SIP) in our architecture is to determine the IP addresses of the home gateways at any point of time. There are a number of advantages for using SIP in this architecture. The biggest advantage is its event based notification system and name-address resolution scheme similar to e-mails, because of which SIP works well in mobile environments [34], [35]. Moreover, both data and header information in each SIP message is encrypted using strong encryption algorithm, which can be deployed in a hop-by-hop or end-to-end basis. The encryption of the header information makes it more difficult for an eavesdropper to figure out the location and/or commands sent by the user [34]. Another advantage of using SIP is that, it is independent of the Transport Layer protocol being used (can be used with UDP, TCP, SCTP, etc.) and also does not depend on the type of session established. SSL also provides excellent security but unlike SIP, which uses small packets to transfer data, it needs an elaborate connection-establishment phase even for transferring small messages, which can cause heavy load on the residential gateway and consume unnecessary bandwidth. For all these reasons, we propose the use of SIP by the authentication server.

### **1.6. Objective of the Research**

From the proposed architecture discussion in *Section 1.5*, it can be observed that most of the previous researches, analysis and work were done on various domains like on secure communication, biometrics, iris recognition, hashing, image

hashing, home networks, etc. independently, with little effort given on the practical integration of these diverse domains to provide a complete and compact solution for the development of a scheme to access home appliances over the Internet in a very secure and convenient fashion. A lot of literature can be found on iris recognition, on secure communication, little on secure communication of biometric data, etc. but all of them were never integrated together to complete the architecture for this secure home network access. Previous efforts suffer from the drawback of being very user dependent, missing application security, being time consuming to setup and even requiring manual setting up of the different components to make the whole architecture work. These usually made users (probable customers) spend huge amount of time and money on buying equipments, its setup and its maintenance necessitated specialized knowledge in order to operate on them. Moreover, most of the previous designs require a separate dedicated server for each home thus escalating cost for this kind of endeavor, and hence, making it commercially unrealistic. Each of those architectures also supported access to only a single user at a time, which can sometimes become a problem for families implementing this system. They also did not have any security or access control mechanism, which made these methods vulnerable and difficult to use commercially without further work. So, the principle objective of our research was to attempt towards eliminating these problems by incorporating the diverse fields of secure communication, biometric authentication, image hashing, home server and home networks, and selecting the best alternative from each of the different components, in order to devise a smooth, well integrated, secure, user-friendly and cost effective means of

accessing home appliances over the Internet. We were able to make a number of interesting innovative additions in this whole design so as to benefit the topic of our research. One of our main concerns during this design was to keep user responsibility to the minimum and minimize the response time of the system, thus requiring minimum time complexity of algorithms. Since security module was never implemented in any of the similar previous researches, we tried to develop it and integrate it well with the architecture such that enhanced security does not hamper the functioning and usability of this endeavor.

## **1.7. Original Contributions to the Knowledge**

The following principal contributions have been made towards achieving the objectives of the research described in this thesis:

- A novel architecture is developed to implement a secured remote access to home appliances over the Internet. This design involves a 3 layered-security system: (i) SSL, (ii) iris recognition and (iii) image hashing using random numbers. A detail explanation of each of these security layers is discussed in Chapter 2, 3 and 4 respectively.
- A communication channel is constructed using C language that is secured using OpenSSL (v-0.9.8g). This channel embodies a real world client-server interaction system where commands and biometric data are exchanged.

- A novel image hashing technique is developed using MATLAB. It utilizes Mersenne twister randomization algorithm and Gram-Schmidt orthonormalization algorithm to hash biometric data and produce a hash code that is approximately 2% in size compared to that of the input biometric data. The details of its implementation are described in Chapter 4.
- A new server called the Authentication Server is proposed that can handle user authentication for multiple homes. This will reduce user responsibility for installing and maintaining the server, and the cost of implementing remote access to home appliances. Description on how it can be implemented is illustrated in Chapter 5.
- A method of using SIP has been proposed in order to exchange control messages between an authenticated user and the home gateway, by harnessing its event based notification and security features. Details are illustrated in Chapter 6.

## **1.8. Organization of the Thesis**

The entire thesis deals with the development of an architecture to implement access to home appliances over the Internet. The subsequent discussion of this thesis has been organized into the following chapters. A brief overview of the different technologies available to secure the communication channel between the client and the server is provided in **Chapter 2** along with the justification for using SSL Communication Channel in our application. **Chapter 3** elaborates about the biometric authentication used in this research. **Chapter 4** explains the scheme of

implementing image hashing for iris images. **Chapter 5** elaborates the existing server architectures available in order to access home appliances over the Internet, and also explains why we need a new architecture for our application. The various home networks available today are described in **Chapter 6** along with the reasons for selecting UPnP as the home network in our architecture. Finally, **Chapter 7** concludes the thesis with results and discussions on the work and experiments that have been carried out during this research, and provides some suggestions related to its possible future extension.

## **Chapter 2**

# **Secure Communication Channel**

### **2.1. An Overview**

Internet has revolutionized information access and communication between different computers, transforming personal and mass communication forever by incorporating digital technology in all aspects of life. On one hand, it is an information repository, a ground for collaboration between people and organization; and on the other hand, it is a channel for imparting knowledge and interaction between people and their computers, irrespective of their geographical location. The real world applications of Internet have undergone major changes through the decades. Currently, Internet is being exhaustively used for online shopping, e-commerce, banking, stock brokering, etc. Some communication protocols like Transfer Control Protocol/Internet Protocol (TCP/IP), Hypertext Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), File Transfer Protocol (FTP), etc. [36] have become standards for transferring diverse kind of information in this cyber world. But all the above protocols exchange information in plaintext, and hence, transit data is easily readable. As a result, confidential information (like credit card numbers, bank account numbers, passwords, etc.



required for e-commerce) while in transit via Internet can be easily intercepted and read if no security measures are implemented in these protocols. This gave birth to the notion of security for data in transit on the web, and paved the way for Information Security.

An Internet protocol with security features should enable verification of both client and server before exchange of any actual data and also be able to protect the confidentiality of information exchanged between them. A number of protocols have been developed over the years with this aim, most of which used public key technology to encrypt and decrypt data in transit, like in Secure Hypertext Transfer Protocol (SHTTP), Secure Sockets Layer (SSL), Transport Layer Security (TLS), IP Security (IPSec), Layer 2 Tunnel Protocol (L2TP) and Point to Point Tunneling Protocol (PPTP).

Among these, the most notable are SSL/TLS and SHTTP protocols, all being used for TCP/IP communications on the Web, which we discuss in detail in *Section 2.1.1*. A Virtual Private Network (VPN) can be used to implement this secure communication and it is already being used by many companies to let their employees access company's network securely. In *Section 2.1.2*, we discuss more about VPN and the disadvantages of using this in our application. Later in *Section 2.2*, we discuss in detail the Secure Sockets Layer (SSL) that we use to secure the communication channel between the user and the server.

### **2.1.1. Secure HTTP**

#### ***a. Overview:***

Secure HTTP also known as S-HTTP, is a secure message-based communication protocol designed by Eric Rescorla and A. Schiffman for use in the World Wide Web in order to secure the HTTP connections [37], [38]. The main objective of developing this protocol was to provide confidentiality, authenticity and integrity to the messages exchanged over HTTP. This is a very flexible protocol as it is not tied to any particular cryptographic system or key infrastructure. Rather, it provides a variety of security options to HTTP by encapsulating each message and performing encryption and digital signature on it, inserting Message Authenticity Check (MAC), and corresponding headers so as to transfer the message efficiently. It is very customizable in nature and provides multiple key management schemes, operational modes, trust models, cryptographic algorithms and encapsulation formats [32], predominantly utilizing RSA and Diffie-Hellman public key encryption systems to exchange session keys. It is compatible with normal HTTP connections and therefore, S-HTTP clients can successfully communicate with a normal HTTP server and vice versa, but such transactions would miss the security features of S-HTTP. S-HTTP operates on symmetric key-only operation mode and hence does not require any client side public key certificates [32]. This means unsolicited confidential transactions that would ultimately help in avoiding time consuming communications for establishing a public session key. This protocol works in a very similar way to SSL but with minor differences discussed later. For setting up of the communication channel, at the very beginning of the communication, the client

requests for a page to the server using SHTTP, and along with it, sends a session key after encrypting it with the server's public key. If the server is a valid one, it will be able to decrypt the message and the session key using its private key. It then sends the document requested by the client, after encrypting it with the session key sent by the client. This simple mechanism helps in securing the HTTP messages from unwanted alterations and helps in establishing confidentiality of the message as nobody other than the actual server will possess the proper private key and hence, will not be able to decipher the encryption key sent by the client.

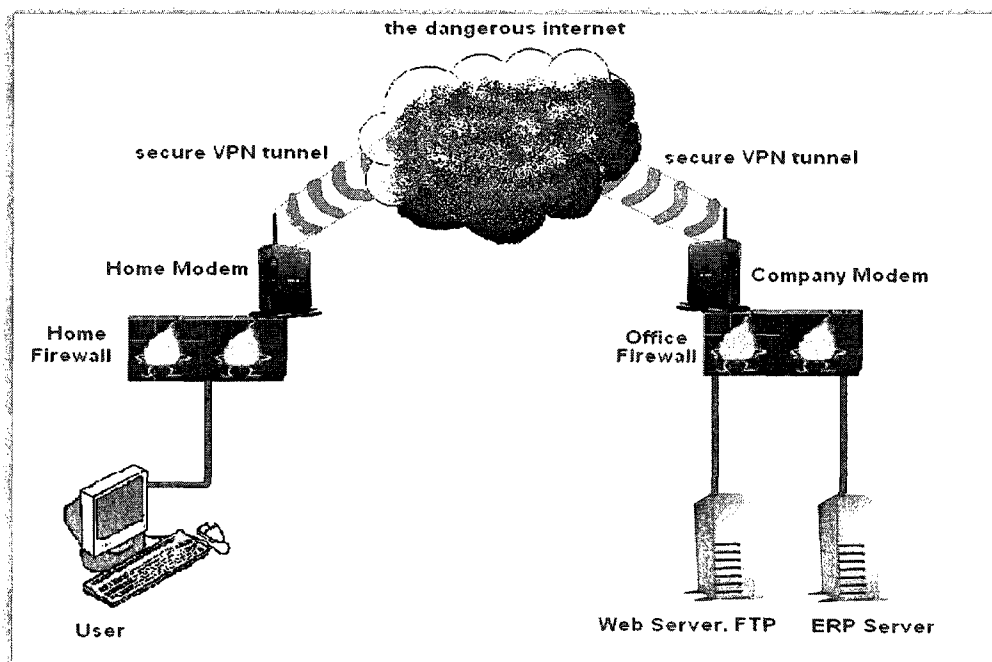
***b. Associated Problems:***

A major problem with S-HTTP being used in this context is that, every S-HTTP request from the client and every response from the server is encrypted separately and then transmitted over an insecure connection. No attempt is made to secure the communication channel as a whole; just the messages are encrypted to provide the security in the HTTP messages. This increases the overhead of performing complex encryption procedure for each message and is not suited for situations where there is a long chain of data to be exchanged between terminals. SSL on the other hand first establishes a single secure connection between the client and the server over which command messages and data are continuously exchanged. Here, the whole communication channel is secured and not just individual messages being transmitted over it.

## 2.1.2. Virtual Private Network

### a. Overview:

As technological innovation improved and simplified access to markets, the need for better and stronger ways to ensure confidentiality, integrity, authenticity and availability of data on the Internet was felt for both Business-to-Business (B2B) and Business-to-Consumer (B2C) applications. A Virtual Private Network (VPN) can be used to securely access the networked home appliances the same way it is implemented by organizations to enable its employees secure access to office network from their home, using the public Internet as shown in Figure 2.1. It is actually a private communication channel between two or more devices, using the public network like Internet. In other words this network uses the common public network (like the Internet) to transfer its data to other remote sites or networks or



**Figure 2.1:** Virtual Private Network (VPN) operating over the Internet

users, thus sharing information as if it were in the same network, though physically separated. Thus, the VPN data that actually travels over the public network can be prone to sniffing or tampering attacks. But this data is protected using very strong encryption techniques, which is why it is secure. Moreover, VPN's monitor the traffic flowing between two VPN enabled devices using sophisticated techniques, which ensures that no packet is manipulated while in transit [39], [40].

***b. Associated Problems:***

VPN has some major problems in addition to being very expensive to implement [41]. Firstly, a VPN connection is difficult to implement by a user as it needs specialized or professional knowledge to implement it. Without this specialized knowledge, a third party help will be needed by users to use this technology. Secondly, VPN connections are slower than normal connectivity because of which, authenticating a user and sending commands to the home appliances becomes more time consuming than regular connections. A third major problem with VPN connection is that, any security breach in the user's computer from which he/she is accessing the home appliances can threaten the security of the home gateway and hence the home network. So extra care needs to be taken by the users in order to secure the terminal that is used to access this web application. Fourthly, VPN products have multiple vendors which have been mostly found to be incompatible with each other [42], [43]. As a result, all the devices used in accessing home network must be from the same vendor, which imposes an unsatisfactory restriction on the consumers. Lastly, in VPN connectivity, firewalls can be

implemented in two ways, one in front of the VPN server and the other, behind the VPN server [43], [44]. Both these implementations are very complex, which makes specialized knowledge customary for implementing firewall in VPN. Compared to this, SSL makes working with firewalls much easier by setting up the packet filtering firewall in a way that it receives all SSL connections on a particular port and all packets through that port is allowed to pass through unrestricted. This way, SSL connections can pass through firewalls unrestricted and hence, gateways implementing such firewalls will not hamper the SSL connections.

Considering all the above reasons, the need for a superior solution other than SHTTP and VPN was felt in order to access home appliances via Internet. The new architecture needed to be easy to implement, vendor independent, cost effective and secure, which not only enables accurate verification of a user, but also helps in providing maximum security to the data in transit viable in this architecture.

## **2.2. Secure Sockets Layer**

### ***a. Overview:***

Secure Sockets Layer (SSL), is a protocol designed and implemented by Netscape Communications in order to secure data in transit in the Internet. SSL is a lower layer protocol, between the Transport and Application layer, and can be used with all other protocols. It runs below the higher-layer protocols like TELNET, HTTP or FTP and above TCP/IP. It mainly deals with verification of client and server, message encryption and authentication. It ensures privacy in communication through symmetric encryption and integrity through message authenticity check

(MACs) codes [45]. SSL certificate helps to reinstate the authenticity of the server and if needed the client also [11], [12]. Each SSL certificate is verified and signed by a trusted third party called Certificate Authority, which confirms that the certificate holder is really the one it claims to be. This certificate contains unique authenticated information about the certificate owner (verified by the CA), and this is primarily what makes SSL very secure. After both sides participating in communication have verified each other's certificates as part of the handshaking protocol, SSL establishes a session key valid for that particular session only, which is used to encrypt all the data that is exchanged between the client and the server. The client can now perform transaction with the server being assured that it is talking to the genuine server and not with any fraudulent one. However, SSL is prone to man-in-the-middle attack, which can be thwarted effectively by using mandatory certificate verification for both client and server. As a result, we have used mandatory certificate checking of both client and server in our system. If the client connection does not find a valid server certificate, it terminates its connection and looks for the valid server. If the server on the other hand, does not find a valid client certificate, it blocks the client from accessing its resources, and ultimately, terminates the connection.

***b. Architecture:***

SSL is entrusted with performing the following responsibilities on the messages in transit:

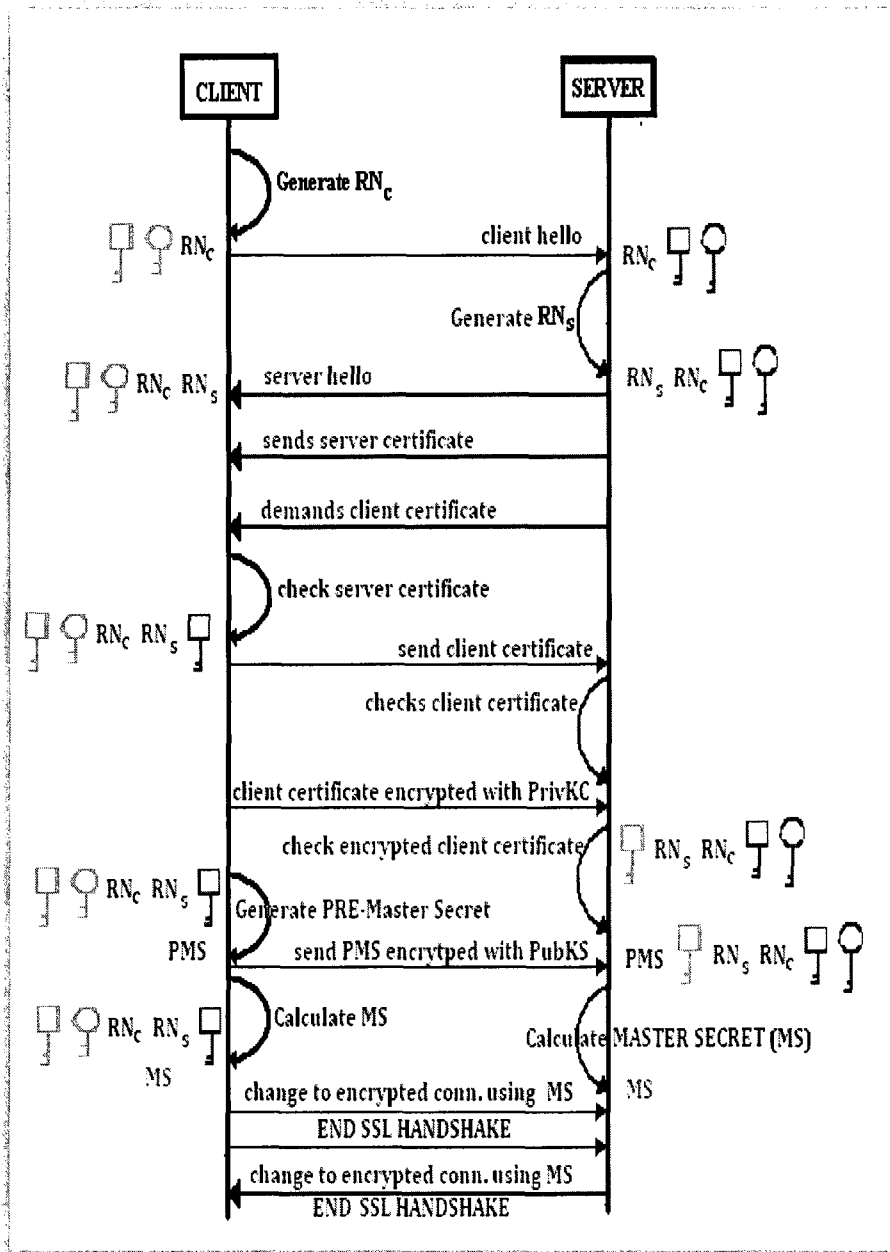
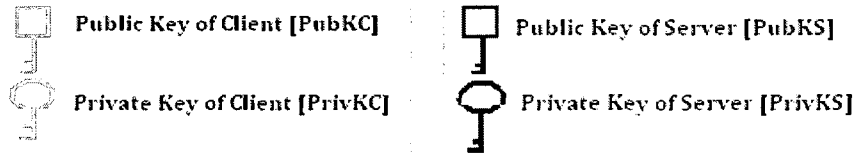
- Authenticate the server by checking server certificate and keys

- Authenticate the client using client certificate and keys
- Provide data confidentiality using exchange of data in encrypted format
- Provide integrity using message authentication codes

At numerous occasions, Secure Sockets Layer/Transport Layer Security (SSL/TLS) protocol is used interchangeably to mean SSL protocol. Here, in this research, we use SSL v.3 in particular, as there exists a minor difference between SSL and TLS, which contributes majorly towards the security of the communication channel. TLS uses the Keyed-Hashing for Message Authentication Code (HMAC) algorithm for producing the integrity checks value, whereas SSL uses the MAC algorithm, which is more secure than the HMAC algorithm.

We have depicted the procedure of SSL communication in the following Figure 2.2 and explained it in details following the figure. Let us assume that the user requesting a 'https' page be called a 'client', the terminal sending the requested page to the client be called a 'server' and the medium used to exchange messages between them be called the 'communication channel'.





**Figure 2.2:** SSL handshaking procedure

The following steps are executed to implement a SSL communication between a client and server [11], [12]:

- i. A browser, i.e. a client requests for a page securely (<https://...>)
- ii. The web server sends its certificate to the client (that has been signed by a recognized Certificate Authority and contains the server's public key)
- iii. The client browser checks whether the certificate is rightly signed by a trusted CA, valid, and really belongs to the server contacted or not
- iv. The client browser then chooses a random number as the symmetric encryption key (also called the session key) and encrypts it with server's public key before sending it over to the server. Along with this, it also sends the URL requested and other data encrypting it using the same session key
- v. The server proves its identity by using its private key (corresponding to the particular public key) to decrypt the encrypted session key (encrypted with server's public key). After being able to successfully extract the session key, it decrypts the requested URL and other data using this extracted session key
- vi. The server then sends back the page requested all encrypted using the session key

Thus we see that with certificate checking of both client and server, we can actually make sure that both the server and client are valid. Thus, due to the implementation of mandatory certificate checking in our architecture, the server will be unable to validate the user (during the commencement of SSL connection) on absence of this

certificate in the client terminal, and hence, fail to establish any connection with that terminal.

**c. Advantages:**

SSL provides numerous benefits to the users implementing it in their network. Few of the advantages are listed as follows:

- i. **Strong authentication mechanism, message privacy and integrity** – SSL is mainly responsible for securing transmitted data using encryption. It also provides server and client (optional) authentication based on certificates. Addition to this, it checks integrity of data through an integrity check value. It can be successfully used to thwart various kinds of attacks on digital data like man-in-the-middle attack, bucket brigade attack, rollback attack, replay attack, etc.
- ii. **Interoperability** – SSL is compatible with most browsers including Microsoft Internet Explorer, Netscape Navigator, Google Chrome, etc. and on most widely used operating systems.
- iii. **Algorithm flexibility** – there are multiple options for the SSL authentication mechanisms, choice of encryption algorithms and hashing algorithms that can be used during a secure session.
- iv. **Ease of deployment** – SSL can be used transparently by many applications and is easy to deploy.
- v. **Ease of use** – since SSL operates below the application layer, it is mostly invisible to the user and still be protected from attacks.

## 2.3. Secure Bookmark

From the above discussion, we can conclude that safety through certificate necessitates every user to obtain a user certificate from the Certificate Server (that can be the same server as the authentication server that releases this certificate to a limited registered user's community) and install it in one's web browser before actually using this web application. Things become difficult when a user attempts to use a public workstation to access this web application, for which, he/she first needs to install the client certificate in the IE of each of the computers used and then, carefully remove it after use (in order to maintain the application's security). Thus, because of its inconveniences, we propose avoiding client certificates and instead, use a unique, identifying bookmark [46] to perform the pre-screening of users. This bookmark can be communicated to every user of this application through registered e-mails to always log into the website. This user-friendly and simple mechanism effectively permits preliminary user verification (as a non-registered user will not receive this e-mail with the bookmark link) and consequently, prevents MITM and phishing attacks successfully. After this phase, SSL establishes a session key that is used to encrypt all the data exchanged between the client and server, similar to symmetric encryption. We implemented SSL communication channel in a Linux machine with standard configuration and using OpenSSL version 0.9.8g and C language. We developed two programs that represented a client and a server exchanging messages between them. The details of this Client and Server program along with the steps performed to set up the environment are discussed in detail in the results and discussion section in Chapter 7 *Section 7.2*.

## **Chapter 3**

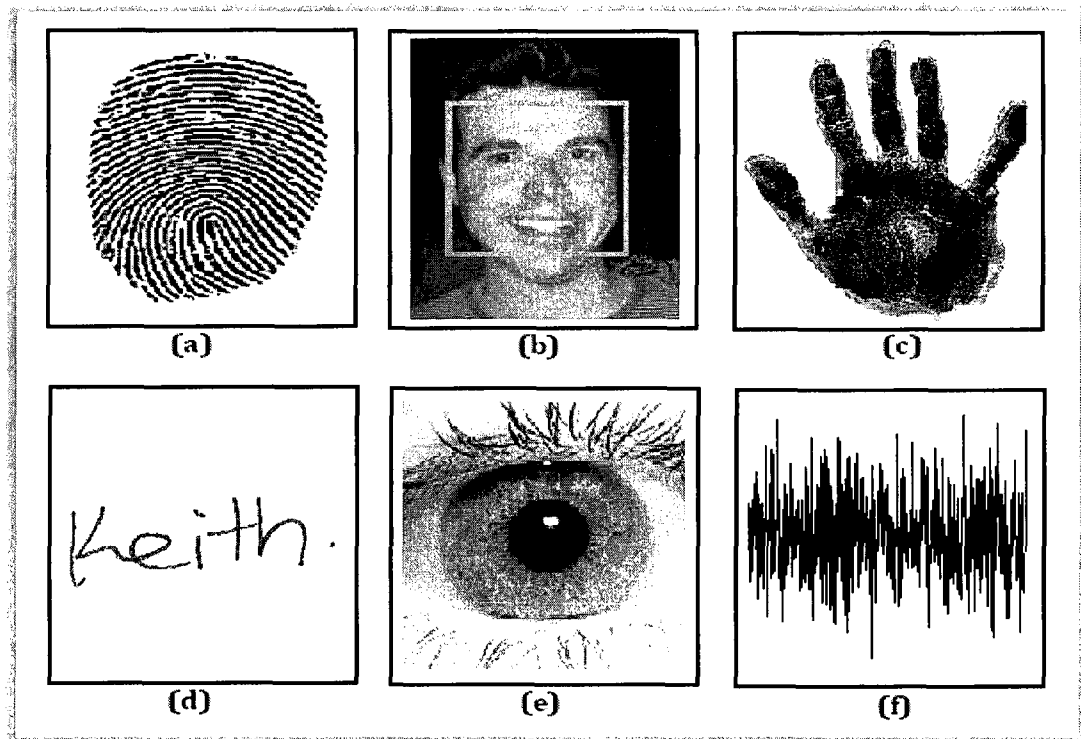
# **Biometric Authentication**

### **3.1. An Overview**

For decades now, biometric authentication has gained a lot of momentum in order to provide security in various areas like in-place security, web security, wireless security, etc. mainly due to its various advantages and accuracy. Another reason for its popularity is the gradual reduction in price of hardware and software components, which helped design complex biometric systems without escalating the cost. By using biometric authentication, it is possible to identify a person on the basis of who he/she is, rather than what he/she possesses (ID cards) or remembers (password) [13], [19], [20]. The main features that are important to consider while developing a biometric system are: it should be easy to capture, induce minimum discomfort to the user while being judged, it must not change over one's lifetime, and that it should uniquely identify a person without any additional data. Considering these criteria, most of the biometric identification systems can be grouped based on the main physical characteristics on which they are founded. These include fingerprint identification, voice recognition, iris recognition, signature, palm print, hand geometry, palm vein analysis, retina scan, face

recognition, gait analysis, ear lobe analysis, body odor identification systems, etc. [15], [18], [21], [24], [47], [48].

Biometrics is used to identify a person both using his/her physical (relatively stable) and behavioral characteristics [15], [18], relying mainly on calibrating a feature of the person, rather than validating an individual based on possession of a card or secret information. Among all the different biometric traits available, blood samples, DNA analysis, or even tissue samples act as perfect features for identifying a person in forensic applications and investigations. But for online, live applications, features like iris, retina, face, fingerprint, voice, signature, etc. are the most popular and accurate method of identification [15], [18], [21], [47], as shown in Figure 3.1.

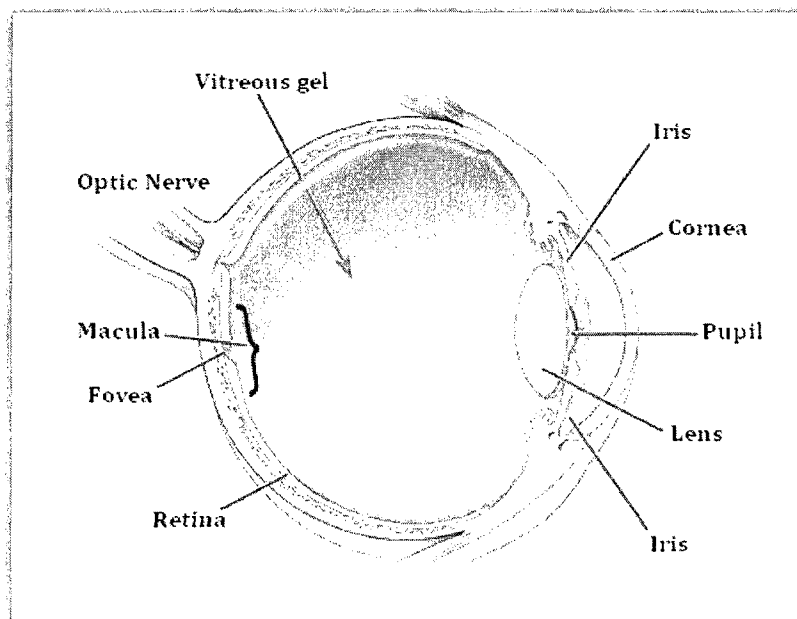


**Figure 3.1:** Sample biometric traits: (a) fingerprint, (b) face, (c) palmprint, (d) signature, (e) iris and (f) voice

Among these features, iris is the most eligible trait that we can use for our application because of its universality yet distinctiveness, constancy and ease of measurement [47].

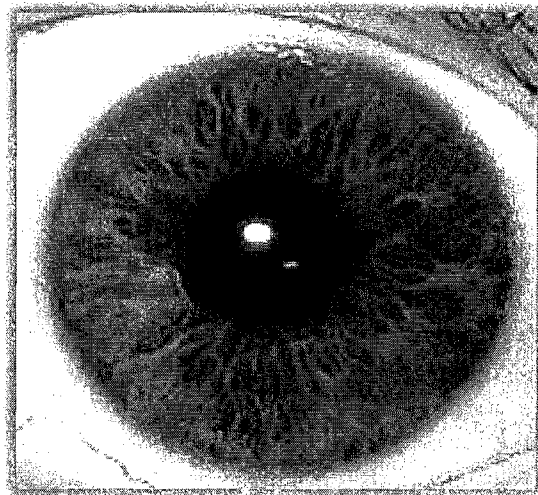
### 3.1.1. Iris for Biometric Recognition

With recent improvements in security measures in almost all applications, personal identification systems founded on biometrics have also undergone an enormous growth in technology and awareness [15], [18], [21]. Iris recognition is considered to be one of the most reliable biometric features that can be used to identify a person effortlessly in present biometric systems. It is an internal organ of the human body, which is well visible from outside and can be measured without trouble from short distances. It is the pigmented part of the eye (contains the pigment called melanin), with a round opening in the centre called the pupil (see



**Figure 3.2:** Structure of an eye

Figure 3.2 and Figure 3.3). Iris contains tiny muscles, called dilator and sphincter muscles, that dilates (widens) or constricts (narrows) the pupil and thus controls the amount of light reaching the retina [49]. It is a unique structure made up of fibrous tissue that contains minute freckles, coronas, zigzag collarette area, etc. which forms the basis to calibrate this feature [21]. It is completely formed by the eighth month of pregnancy, though the color and pigmentation continues to build till the first year of birth. After this time period, it remains permanent throughout one's lifetime, unless directly harmed or damaged due to accidents or surgery. This constancy is one of the winning factors that makes iris much more popular and



**Figure 3.3:** Sample picture of an iris

effective than all other biometric features.

Moreover, recognition systems based on iris verification are nonintrusive to users, which is of great importance for real-time applications [50]. Due to the above reasons, iris has found popular ground in various surveillance systems, like in



airports, passport offices, security areas, wireless authentication, at the bank's cash machines or for online bank transactions, while getting driving licenses, forensics, etc. As we can see in Figure 3.3, pupil is the central transparent area showing as black. The multicolored area (greenish black) surrounding the pupil is the iris. The iris sits inside a whiter outer area known as the sclera.



**Figure 3.4:** Samples of iris images from CASIA 1 and CASIA 2 iris image databases

### 3.1.2. Issues in Iris Recognition

Though iris recognition is the best feature that we can use for our web based, thin client application, there exists a few problems with using it effectively.

1. There exists no large scale database for testing the accuracy of these iris recognition systems. Most of the databases available today are small and does not effectively represent the population even of a country.
2. The quality of the iris images also plays a big part in recognizing people and hence needs good image capturing devices for this purpose.

3. Liveness detection for iris images is still not implemented convincingly. Hence, prints of iris images can cause a fraud to enter or access such iris-based authentication systems.
4. In case of iris-based recognition systems that use templates to store iris data and later use it for verification, theft of iris data can cause major problems in regenerating these templates for the said person.

### **3.2. Related work in Iris Recognition**

In recent years, automated iris verification has been drawing a lot of attention among other biometrics owing to its advantage of non-invasive authentication and accurateness in identifying people. It has paved its way within a wide range of business applications where identity of an individual needs to be determined accurately for access, determination of privileges, security, tracking, protection of resources, quality of service, etc. Lately, a lot of research has been done resulting in the discovery of various techniques for identification. Iris identification process primarily consists of the following four steps:

- a) **Localization** – In this phase, the inner and outer boundaries of the iris image is calculated.
- b) **Normalization** – Iris image of different people may be captured in various sizes due to many reasons, like varying closeness to the camera, actual size of the iris of that particular person, variation in illumination, etc. In this phase, the input iris image is made independent of its size in polar coordinates.

- c) **Feature Extraction** – In this phase, a feature vector is constructed, which consists of the features extracted from the iris images.
- d) **Matching** – In this phase, the feature vectors are classified using various similarity measures like Hamming distance, Euclidian distance, dissimilarity function, weight vector and winner selection, etc. and the decision is made whether the input and model images are from the same individual or from different people.

The credit of proposing the concept of iris recognition for the first time goes to Flom and Safir [51], though it is not known whether they had actually developed and tested the system or not [21]. Later, work progressed in Los Alamos National Laboratories, CA [52] and later in Europe and North America [53], [54]. But the most well known scheme was developed in 1993 by John Daugman, where iris recognition was done on phase code using Gabor filters [15], [50], [55], which was later patented by IriScan Inc. In this method, the visual texture of each person's iris image is encoded into a compact sequence of multiscale Gabor wavelet coefficients. These 1024 complex phasor values, which denote the iris phase configuration at various scales, were produced after filtering the iris image with a family of filters. Then, each phasor value is quantized into one of the four quadrants in the complex number plane. This results a 2048 bit iris code that can be used uniquely to describe an iris of a person. According to this scheme, Hamming distance was used for evaluating the difference between two iris images of same or different person. Wildes et al. [56], [57] proposed the representation of iris texture using a system

based on isotropic band-pass decomposition derived from the application of Laplacian of Gaussian filters to the image, constructed with four different resolution levels, and then used normalized correlation for measuring the similarity between the input image and the model image in order to determine, whether they are from the same class or not. Wildes used the first derivative of image intensity like Daugman to find the edges corresponding to the boundary of the iris image. This system modeled the upper and lower eyelids of the iris image with parabolic arcs unlike Daugman, who omitted those portions from the image. This system produced remarkable results while identifying a person in minimum time.

Sanchez-Reillo et al. [57] employed a partial implementation of the algorithm proposed by Daugman, using Gabor filters as feature extractor and then using a statistical matcher. Boles et al. [58] used a zero-crossing representation of one-dimensional (1-D) wavelet transform at different levels of resolution of a concentric circle on an iris image for its feature extraction. Firstly, they localized and normalized the iris image using well known computer vision algorithms, then, calculated the zero-crossings of the wavelet transform at various levels of resolution, over concentric circles on iris. This resulted into one-dimensional (1-D) signals, which was then compared with the model features using various similarity functions. This algorithm was found to be effective in handling noisy situations and immune to geometric transformations like scaling, translation and rotation of the images [57]. A similar kind of work was performed by Sanchez-Avila et al. [59], where they further developed the method used by Boles et al. in [58] and used different similarity measures for matching, like Euclidean distance and Hamming

distance. This method also showed high level of accuracy while identifying individuals using iris image.

Ma et al. [60] at the same time adopted texture analysis methods in order to capture the details of an iris image. Here, the global features of the iris were extracted using Gabor filters at various scales and orientations. They went to construct a collection of spatial filters in order to construct the local texture features of iris, suitable for the recognition procedure. Later, they developed [61] a feature extraction method based on Gaussian-Hermite moments for capturing the local intensity variations of the iris, which has been already decomposed into 1-D intensity signals, and showed it to produce minimum redundancy in data. In another work, Lim et al. [62] showed better results when dividing the iris image into four levels using 2-D Haar wavelet transform compared to Gabor transform. Afterwards, they quantized the fourth-level high-frequency information to form a 87-bit code that aptly represented the iris pattern and improved the system performance. Tisse et al. [63] used Hilbert transform to analyze the iris characteristics. Park et al. [64] decomposed an iris image into eight directional subband outputs using a directional filter bank and calculated the normalized directional energy as features. Matching of the iris images was performed using Euclidean distance between the input feature and the template feature vectors. Further work was done by Kumar et al. [65] in this area using correlation filters. Two dimensional (2-D) Fourier transforms were used on training images to design the correlation filter of each class. The decision that the input image is from an authorized subject or not is decided based on the correlation output; sharp peaks in the output denotes an authorized individual, otherwise an

imposter. Bae et al. made an important contribution in [66] where they decomposed the iris signals into a group of basis vectors that has been derived from independent analysis of components, and then, quantized the resulting decomposed coefficients as the required iris features for future matching. With all these various proposed and proven, efficient iris representation techniques, we can observe the existence of mainly four forms of approaches towards iris representation [17]: phase-based [56], [67], [68], zero-crossing [58] [59], texture analysis [61], [62], [64], [69], [70] and intensity variation [71] [72].

Among all these different methods available, we used texture analysis based method for iris normalization and its subsequent feature extraction using 1D log-Gabor filters. Details of this method are described in Chapter 4 *Section 4.4*.

## Chapter 4

# Hashing

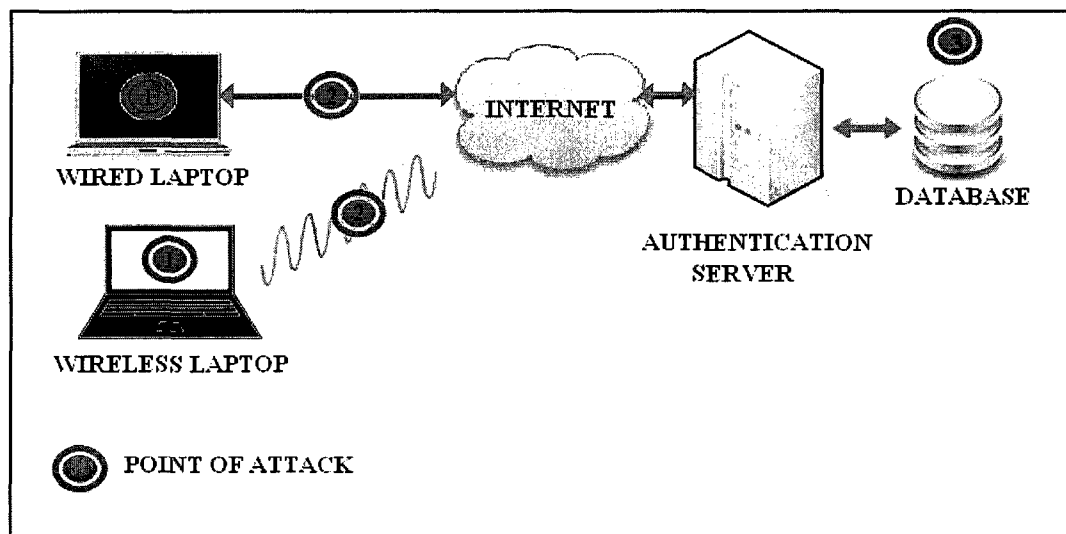
### 4.1. An Overview

With an increasing emphasis on digital security, biometric authentication systems have received a lot of attention from both cryptographic (biometric cryptography) and hashing (BioHashing, S-Iris, Image fusion, etc.) domains [25], [48], [52], [73-82]. The security pitfalls in using passwords, PINs and smart cards led to the popularity of biometric recognition systems. A biometric system is less susceptible to password or smart card based abuses and has minimum provision for human errors like forgetting or misplacing passwords, guessing, ill-use, etc.; however, it does suffer from few intrinsic biometric-specific threats [71] described in *Section 4.2*. These threats can be resolved using both hashing and cryptographic techniques depending on the requirement of the system. In the next few sections, we describe the major threats that our application is prone to and how we resolve it using BioHashing technique. Reviewing from the currently available research work on BioHashing, we can conclude that a unique token is required for every user in order to attain 100% recognition accuracy. This necessitated every user to carry a lengthy unique token in a safe device like USB, smart card, etc. This eventually

reduced the security of the application from another front: the user-possessed key can be prone to various attacks like sharing, stealing, copying, etc. Moreover, this requirement reduced the accessibility of the whole system. In order to eliminate this security problem and increase usability of this application, we developed a novel architecture using BioHashing and a random token, which secures this application's data and control messages simultaneously, without asking the user to submit any user specific token in the system during authentication.

## 4.2. Potential threats to biometric data

Biometric authentication provides better security over traditional authentication systems, which is prone to several attacks like password guessing, stealing tokens, eavesdropping, trojan horse attack [83], brute force dictionary attack [84], sniffing attack, key logger program attack (a program that can be









**Figure 4.1:** Points of attack in a typical web application that requires user authentication



secretly installed in the target computer to record all keys typed in by the user), virus attack sent through social engineering, etc. Biometric authentication system is less susceptible to such abuses and has minimum provision for human errors [85] like forgetting or misplacing passwords, guessing, ill-use, etc., but it does have its own limitations and suffers from some security threats when accuracy of the recognition system is low or when it is integrated with an unsupervised web based application. The plausible threats are depicted in Figure 4.1 and described briefly in Table 4.1 below:

Table 4.1: Different types of attack on Application data in Client-Server architecture

TYPE OF ATTACK	NATURE	POINT OF ATTACK	REMEDY	SEVERITY
<b>Present fake biometrics to the sensor</b>	Adversary presents a fake fingerprint, palmprint, face mask, etc. to the sensor	Client terminal or sensor 	Detect liveliness of source data (iris, face, etc.)	Low ★ ★ ☆ ☆ ☆ ☆
<b>Brute Force Attack</b>	Adversary tries various combination of different passwords, biometric patterns or templates	Client terminal or Server terminal 	Block user after certain predefined number of authentication attempts	Low ★ ★ ☆ ☆ ☆ ☆ (for web application)

<b>Replay / resubmission on attack</b>	Intercept biometric data and resend it later to the authentication server to masquerade as a valid user	Communication channel  	Randomize biometric data, encode-decode, hashing, etc.	Very High  ★★★★★
<b>Identity theft</b>	Intercept biometric data and later use it to pose as a valid user	Communication channel  	Randomize biometric data, encode-decode, hashing, etc.	Very High  ★★★★★
<b>Channel attack</b>	Listen to or alter data flowing through the communication channel	Communication channel  	Encrypt data flowing through the channel, use SHTTP, SSL, etc.	Very High  ★★★★★
<b>Tamper stored iris data</b>	Tamper iris data stored in the server database	Biometric image database  	Secure database server by encrypting sensitive columns in the database, take encrypted backups, use antivirus, etc.	Very High  ★★★★★

### 4.3. Methods to safeguard attack points

From Table 4.1, we can see that most of the serious threats to an online biometric authentication system emerge from the communication channel between the client and the server. Thus, neutralizing these attacks on the communication channel will allow us to design a highly secured application that protects both application data and control messages (including biometric information). Traditional cryptographic techniques like AES (used in Advanced Access Content System, AACS for encrypting data in Blu-ray discs), DES, etc. consist of an algorithm for encryption and corresponding encryption keys. This key is user dependent and therefore varies for each person, which he/she needs to remember or store safely for future use. Simple keys are easy to remember, but can be easily cracked; complex lengthy keys on the other hand are difficult to crack, but simultaneously, it is difficult to remember, and hence, needs to be stored somewhere where it again runs the risk of being stolen, hacked or even lost [86]. This added task of safekeeping of the cryptographic keys paved the way for biometric cryptography, in which, biometric features serve as the encryption key for encrypting a sensitive message. There are already a few biometric cryptographic algorithms available based on fingerprint [78], [81], [87-89], iris [74], [76], [90], face [75], signature [90], palmprints [84] and voice [72]. In [86], we find an implementation of this biometric cryptography using iris features. Here, a feature vector of 256 bits is extracted from the iris image after preprocessing it, and then, simple mathematical operations like addition, subtraction and Reed Solomon error correcting code is employed to encrypt and decrypt the data.

Hashing is a message digest function  $H$  in which an input string of characters and numbers ( $m$ ) is transformed into a shorter length string ( $s$ ) that represents the same input string  $m$ , always under the same conditions (keeping the hashing algorithm and other variables unaltered), and denoted by  $s = H(m)$ . According to the properties of a hash function, it is relatively easy to produce the hash code  $s$ , but computationally infeasible to extract  $m$  from  $s$ . These functions are called message digest functions because of its utility in producing message digests used for checking the integrity of a transmitted message. Commonly used hashing functions, like MD1, MD4, MD5, SHA-1 have been successfully used in a number of applications like compiler and database search operations, cryptography and verification of digital content in case of tampering or transformations. These functions are designed in such a way that a slightest change in the input string will completely change the resulting hash code [25-29], and thus, are unfit for use in our application as a user will never be able to resubmit the same image for authentication even under similar conditions. Due to difference in surrounding conditions while capturing an iris image, images will differ from one sample to another. But the basic content of an image (iris in our case) will not change in the pictures as long as it belongs to the same person. This way, we use a robust hashing technique called BioHashing that works on the perceptual qualities of an image and is not affected by changes in illumination, darkness, sharpness, position of the eye, etc. This technique is invariant to the variations that are visually insignificant while performing user authentication. It is a simple two-factor security mechanism (token + biometrics) that can be used to provide protection to biometric data and besides can be

incorporated into physical tokens [91] like smart cards, USB token, etc. to enable authentication of a person based on that hashed code. This method has its roots in cancellable biometrics, which was introduced to protect the privacy of biometrics in any biometric authentication system [92], according to which, intentional alterations are applied on the biometric data either in the signal domain or during feature extraction phase in a way that can be reproducible later.

From the above discussion, we can notice that there is a major difference between biometric cryptography and BioHashing: biometric cryptography safeguards other messages or data using biometrics; whereas BioHashing safeguards biometrics using random numbers or tokens. Both provide security, but the message to be secured differs in the two approaches. Moreover, there is an advantage in using BioHash in comparison to encryption because, the former generates a small hash code out of the long input string and in no way, can the original string be recovered from the corresponding hash code. In case of encryption (symmetric, asymmetric and biometric encryption), the input string being encrypted can be extracted when the encrypting key(s) becomes known. Comparatively in case of hashing, even if the hash algorithm and other parameters used (generated from an unknown data after hashing) are known, it is not possible to recover the original input string. Hence, considering the advantages in using hashing and since in our application we need to secure the biometric information in transit, we adopt BioHashing. Later, for securing the control messages being transmitted between a real user and the application, we use SSL's symmetric encryption key for encrypting all the application data.

### **4.3.1. Image Hashing**

Traditionally, hash functions have been employed for verifying integrity of data, for data retrieval using hash tables or lookups and content-based image retrieval (CBIR) [93]. As discussed in the previous section, these functions are very secure, but they lack robustness, as they are very sensitive to the slightest change in the input string. This characteristic of traditional hash functions makes it unsuitable for use in any kind of biometric authentication, as every time a user submits his/her biometric image, it will be different from the one that is stored in the database, even if both are taken under perfectly similar conditions. Thus, the need for a new hashing technique was felt that would be secure and robust at the same time.

Many secure and robust hashing techniques have been proposed [26], [94-97] in order to verify image data. They have been popularly used to determine image authenticity, content-based image retrieval and also in multimedia protection like image, video and audio watermarking [98]. Some researchers proposed a three-step framework in which, they perform feature extraction, quantization and then compression [26], [99] of the quantized code. They proposed the use of a key to randomize the hash code in any of the phases mentioned above. But sometime later, Fridrich et al. [94], [100] indicated that among the above-mentioned three steps of an image hashing algorithm, one should perform randomization in the feature extraction phase using a key to formulate the most secure hash.

There are many image hashing schemes available in literature, which state being both robust and secure. Following Fridrich et al., almost each of those endeavors introduces randomness in the feature extraction phase [97] but using

different approaches. Mihcak's scheme [98] and Venkatesan's [26] scheme introduces security by first dividing the image into rectangles and then choosing random rectangles for feature extraction. Venkatesan used wavelet decomposition in his scheme to obtain subbands and then, calculated averages or variances to quantize it to the binary form [26]. This binary string is then sent to an error-correcting decoder to generate the final hash value. The statistical properties of the wavelet transform helps in adding to the robustness of the hash code, but, it is not tightly correlated to the contents of the image and hence, insensitive to intentional or malicious tampering of the image [101].

Fridrich et al. [94], [101] introduced randomness in hash code by putting the image through a random low-pass filter. This method establishes the fact that, low frequency coefficients Discrete Cosine Transform (DCT) of an image cannot be made large, without making any perceptible variation to the image. In order to make the scheme key dependent, the DCT modes are replaced by DC-free patterns that are generated from a secret key. It results in a message digest, which can be fittingly used for watermarking, integrity verification and embedding secret information in multimedia. Swaminathan et al. [97] secured the hash by implementing weighted summations of the discrete polar Fourier transform over random subsets, making the hash code resilient to geometric and filtering operations. The cryptographic keys used while selecting the random subsets contributed to the security of the hashing technique. They also proposed two additional techniques to make the whole process of image hashing key independent. After extracting the features from the random rectangles, the data is quantized and then passed through Reed-Muller decoder for

compressing the data. In another method proposed by Mihcak [98], the low frequency wavelet subband is converted into a binary image in order to identify the geometric shapes with both significant and insignificant frequency subbands. Later, iterative filtering is performed to extract the geometry of the significant components and thus strengthen the geometrically strong parts, excluding the weaker ones. Most of the previous approaches in image hashing make use of DCT/DWT's low frequency components in order to determine the features of an image. But the performances of these techniques are not very satisfactory, because there does not exist any fitting relation between the contents of an image and the low-frequency transform domain coefficients [101].

Vishal Monga proposed an iterative method for image hashing using feature points. [28]. According to his technique, important image features are first extracted from the image using a wavelet based feature detection algorithm [102]. Then this string is converted into binary form using probabilistic quantization instead of using public key encryption techniques on the extracted feature vector [103], [104].

#### **4.3.2. BioHashing**

All the above described techniques prove to give good results for comparing images which have gone through minor changes like cropping, geometric transformation like rotation, scaling and translation, filtering operations, compression, etc. and checking integrity and authenticity of an image. But they do not perform well for determining the extent of similarity between two images, particularly biometric images. Thus was felt the need for a better image hashing



technique that can be used to decide the extent of similarity between two biometric images.

BioHashing was proposed by Jin et al. [105] as a method of fortifying biometric authentication. It has its roots in cancelable biometrics proposed by Ratha *et al.* [92], which is a method by which biometric templates can be cancelled and reissued if these stored biometric templates gets compromised or exposed by any means. According to this cancellable biometric technique, intentional and repeatable alterations in the biometric image are implanted before storing it in the database as a template for user verification. The same kind of modification is performed on the captured biometric image during verification of the user and then comparison is made between the modified reference template and test data. This removes the *problem of storing and replacing of original biometric data in the database.*

**BioHashing** is a randomly transformed feature-based cancellable biometric authentication system consisting of two-factors for authentication, an individual's biometric feature and a tokenized pseudo-random number [77], [105]. The extracted biometric feature vector is iteratively multiplied with an array of tokenized pseudo-random number, producing a compact code called BioHash code and can be even used for authenticating an individual.

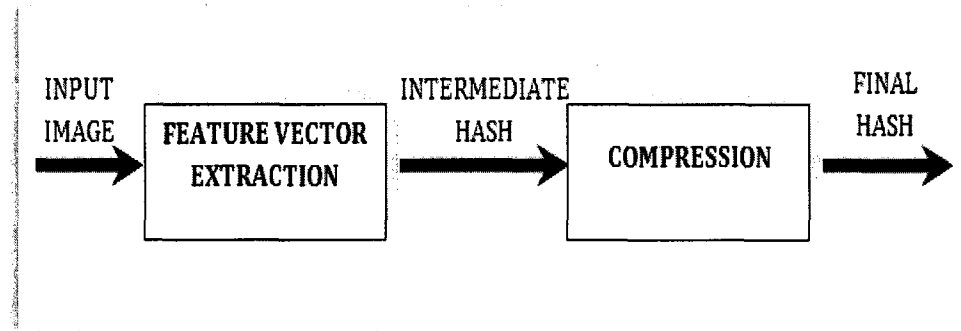
BioHashing mainly consists of two phases [73] as shown in Figure 4.1:

a) Invariant feature vector construction

Invariant and discriminative features are extracted after initial normalization or preprocessing of a biometric image.

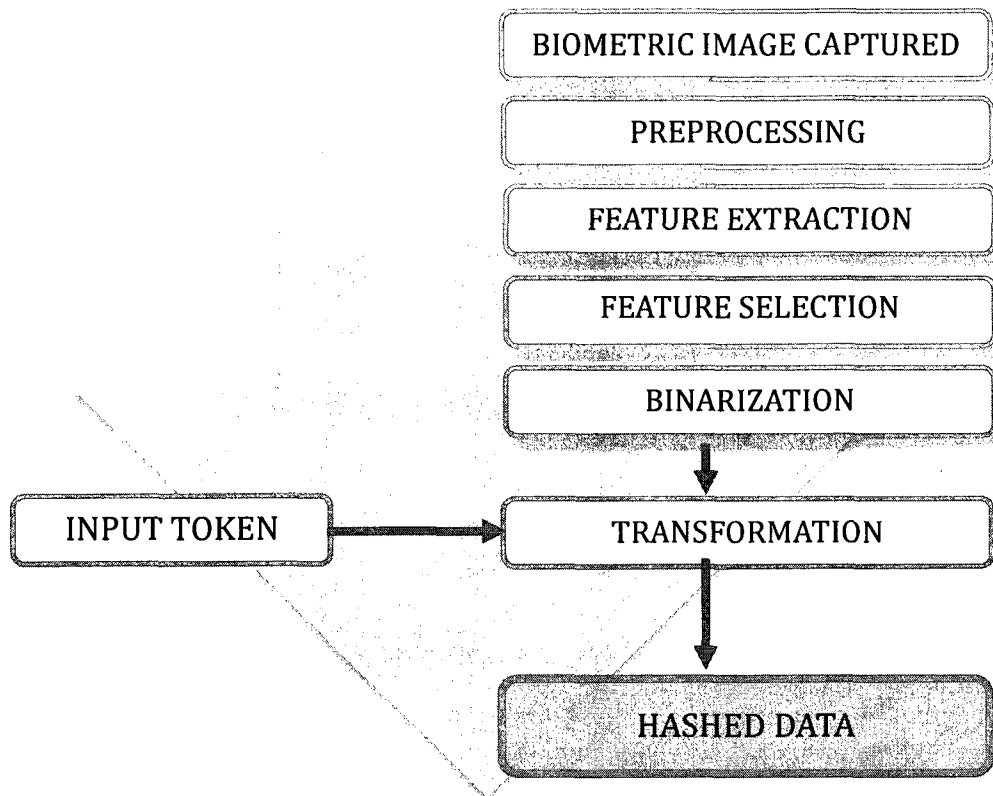
b) Randomization and discretisation of data using a tokenized random number

The biometric feature vector data is randomized and discretised in this phase using an array of tokenized pseudo-random number that also helps in compressing the data.



**Figure 4.2:** Block diagram of a two stage image hashing system

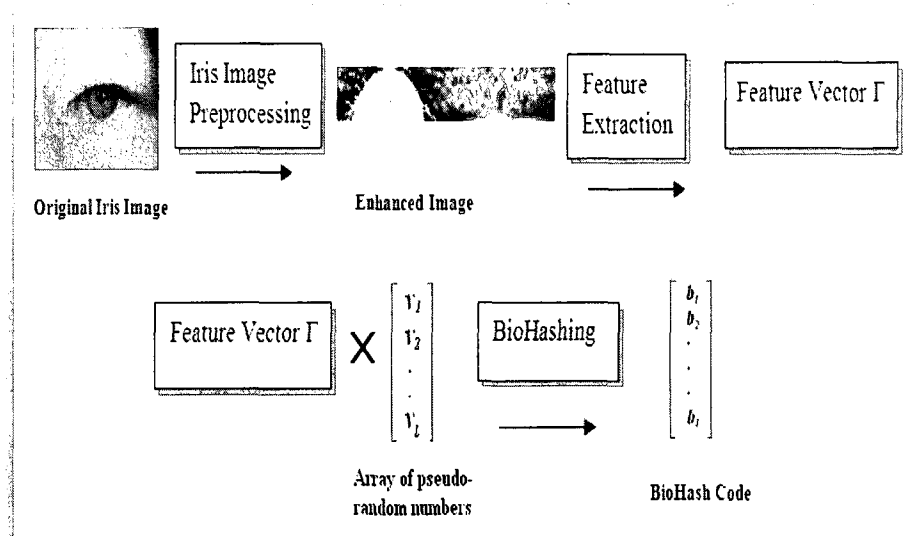
As described in Figure 4.2 and 4.3, the first phase of BioHashing is normalization, where all the irrelevant regions from the image is removed and then a feature vector is constructed from this normalized image. This feature vector is also popularly known as intermediate hash [101]. In this stage, visually alike biometric images should result in identical intermediate hash code, and images from two different persons should result in dissimilar code. In the second stage, the feature vector is iteratively multiplied with an array of random numbers to produce the final hash code.



**Figure 4.3:** A generic hashing procedure where any biometric image is binarized using a biometric hashing scheme

#### 4.3.3. S-Iris

As a variant of BioHashing, Chin et al. [106] implemented the technique of image hashing on iris images, called S-Iris, which is closely related to the problem in cancellable biometrics, where, replacing a biometric feature in the database when compromised is a very difficult issue. He proposed combining two authentication factors, the iris image feature and a pseudo-random number, to generate the cancelable binary code as shown in Figure 4.2 and 4.3. It is a method in which, the



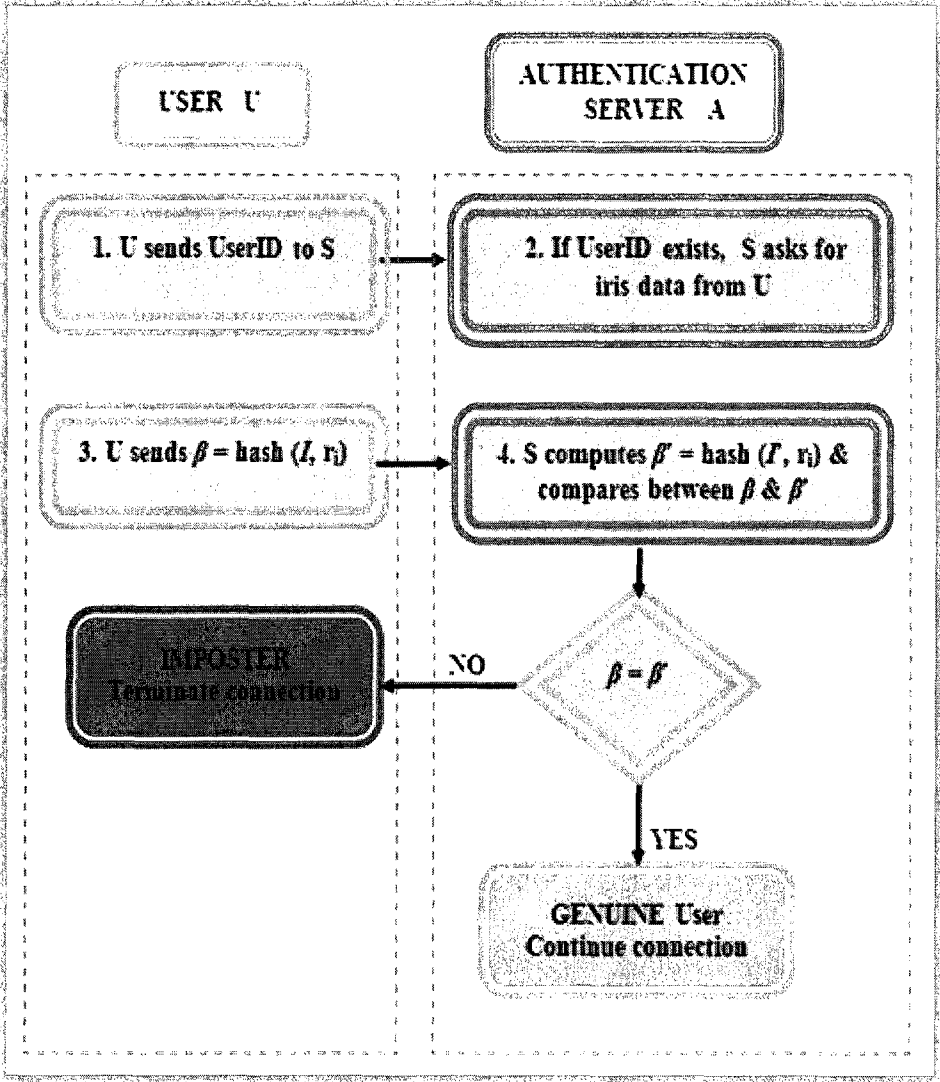
**Figure 4.4:** Procedure followed in S-Iris hashing on user submitted iris image

biometric feature while registration is distorted in a particular way and then stored in the database.

While comparing between two biometric images to verify a person, the submitted feature is distorted in the same way the stored feature is distorted and then matching is performed. Different kind of distortions can be used for different images/persons and this ensures that the actual biometric feature data is never stored in the database, thus increasing security and privacy of the whole system. Chin *et al.* used iterated inner product between the features extracted from the image and a random number and then did thresholding to produce a binary string that is used for matching. This technique used weak inner product exclusion mechanism, in order to push out the weaker values from the binary string and enhance the strong values and improve the authentication rate.

### 4.4. Proposed BioHashing Method

Following the discussions stated above, the second and third security feature in our application is BioHashing of iris features with randomization of the token used during the BioHashing procedure, as shown in Figure 4.5. Together with an improved BioHashing function and a pre-stored unique key for every user, we



**Figure 4.5:** User U and Authentication Server S interaction diagram (USID) during the user authentication process

randomize the resultant authentication code or BioHash code that renders it suitable only for a single use similar to One-Time-Password (OTP). This randomization of the biometric data without affecting the recognition accuracy has been possible only because randomization does not hamper the matching process between two images as the order of comparing two images (here iris images) is immaterial in determining their resemblance [56]. We used Mersenne twister pseudo-random number generator to generate an array of pseudo-random numbers that we later orthonormalize using Gram-Schmidt orthonormalization algorithm [26] to produce the key for BioHashing. This key, known as Tokenized Random Number (TRN) is stored in the server database and need not be carried by any user. As discussed in *Section 4.3.2*, before implementing BioHashing with iris images, we first normalize the iris image following various popular techniques and then construct the feature vector.

The steps followed in this phase are described in detail in *Section 4.4.1*. The second phase consists of generating the array of random numbers (using Mersenne twister algorithm) and its corresponding normalization (*via* Gram-Schmidt orthonormalization algorithm). The steps followed in this phase are described in detail in *Section 4.4.2*.

#### **4.4.1. Feature extraction from iris images**

Iris based recognition systems are non-invasive to users as iris is externally visible though being an internal organ. It is an annular region between the pupil and the white sclera and is enclosed by various non-relevant regions such as the pupil,

the sclera, the eyelids, and also some noise caused by the eyelashes, the eyebrows, the reflections, and the surrounding skin [16], [17]. In order to improve the recognition accuracy, we need to remove this noise from iris images. To do this, we first perform the localization of the pupil. We also detect the eyelashes and the eyelids which are the main sources of occlusion if present. Noise reduction methods are also applied to this localized iris area. The iris area is then unwrapped to form a rectangular block of fixed dimension in order to reduce the deformation of the pupil variation. The distinctive features are then extracted, and finally, hashing is performed on this feature vector to provide the required security to the feature data.

#### ***A. Iris/Pupil Localization***

Both the inner boundary (pupil) and the outer boundary (sclera) of a typical iris can be taken as approximate circles. However, the two circles are not usually concentric [16]. We use the following approach to isolate the iris and pupil boundaries from the digital eye image.

1. First, we apply a morphological operation, namely the opening to an input image to remove the noise (e.g. eyelashes).
2. The iris image is projected in the vertical and horizontal directions in order to make an approximate estimation of the center coordinates of the pupil. Generally, the pupil is darker than its surroundings; therefore, the coordinates corresponding to the minima of the two projection profiles are considered as the center coordinate values of the pupil.

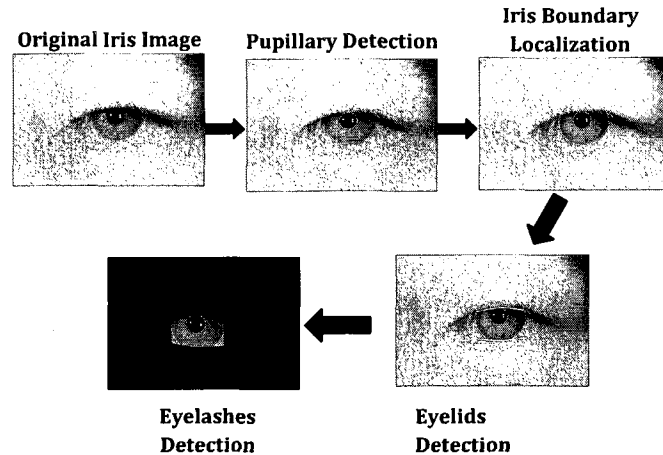
$$X_p = \underset{x}{\operatorname{argmin}}(\sum_y I(x, y)), \quad (1)$$

$$Y_p = \underset{y}{\operatorname{argmin}}(\sum_x I(x, y)) \quad (2)$$

where  $I(x, y)$  is the input image and  $(X_p, Y_p)$  denotes the center coordinates of the pupil.

3. A more accurate estimate of the center coordinate of the pupil is calculated by using a simple intensity thresholding technique to binarize the iris region centered at  $(X_p, Y_p)$ . The centroid of the resulting binary region is considered as a more accurate estimate of the pupil coordinates. We can also roughly compute the radius,  $r_p$ , of the pupil from this binarized region.
4. In this step, Canny edge detection technique is implemented on a circular region centered at  $(X_p, Y_p)$  and with  $r_p + 25$ . Afterwards, circular Hough transform is deployed in order to detect the pupil/iris boundary.
5. We repeat step 4 in order to detect the iris/sclera boundary, replacing the neighborhood region by an annulus band of width,  $R$  outside the pupil/iris boundary. The edge detector is adjusted to the vertical direction to minimize the influence of eyelids.
6. The specular highlight that typically appears in the pupil region is one source of edge pixels. These can be generally eliminated by removing the Canny edges at the pixels that have a high intensity value. Figure 4.6 denotes the localized pupil taken from the input iris image.





**Figure 4.6:** Iris Image preprocessing on CASIA 2 dataset

### ***B. Eyelids and Eyelashes Detection***

The proposed eyelids and eyelash detection technique can be summarized as follows:

1. Eyelids can be approximated as parabolic curves. Therefore, we use parabolic curves to detect the upper and the lower eyelids [107]. Generally, the eyelids occur in the top and the lower portions of the pupil center. Thus, we restrict our search to those areas only. The parametric definition of a parabola is applied to construct the parabolas of different sizes and shapes. Any abrupt change in the summation of gray scale values over the shape of a parabola is estimated for various shapes and sizes of parabolas. This results in detection of the upper and lower eyelid boundaries in the eye image. Figure 4.6 shows the detected eyelids.
2. Separable eyelashes are detected using 1D Gabor filter, since a low output value is produced by the convolution of a separable eyelash with the Gaussian smoothing function.

3. Multiple eyelashes are detected using the variance of intensity. If the values in a small window are lower than a threshold, the centre of the window is considered as a point in an eyelash as we see Figure 4.6.

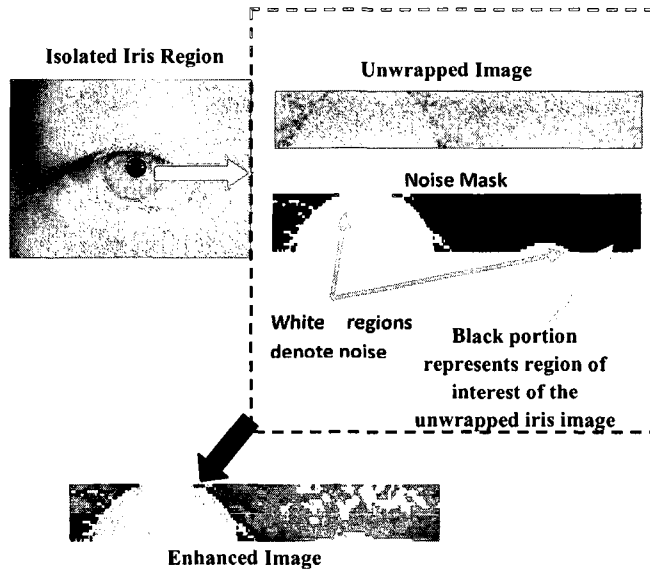
### ***C. Normalization and Iris Image Enhancement***

We use the rubber sheet model [56], [108] to normalize or unwrap the isolated collarette area. In order to compensate for the elastic deformation in iris texture, we unwrap the extracted (and localized) iris region to a normalized rectangular block of fixed size  $20 \times 240$  by converting from the Cartesian coordinates to the polar coordinates [26]. If  $I(x, y)$  is the localized image, then the polar representation of the form  $I(r, \theta)$  can be obtained as follows:

$$\sqrt{(x - x_i)^2 + (y - y_i)^2}, \quad 0 \leq r \leq r_{max} \quad (3)$$

$$\theta = \tan^{-1}\left(\frac{y - y_i}{x - x_i}\right) \quad (4)$$

where  $r$  and  $\theta$  are defined with respect to center coordinates  $(x_i, y_i)$ . The center value of the pupil is considered as the reference point. Figure 4.7 shows the unwrapping procedure. Since the normalized iris image has a relatively low contrast and may have non-uniform intensity values due to the position of the light sources, a local intensity-based histogram equalization technique is applied to enhance the quality of the contrast of the normalized iris image, thereby increasing the subsequent recognition accuracy.



**Figure 4.7:** Unwrapping and enhancement of an iris image on CASIA 2 dataset.

In our method, a local cumulative histogram is applied to the image sub-block of size 10 centered at the pixel to be converted. Figure 4.7 also shows the effect of enhancement on the normalized iris images for CASIA 2 dataset.

#### ***D. Iris Features Set Extraction and Encoding***

Gabor filters based methods have been widely used as feature extractor in computer vision, especially for the texture analysis [56], [108]. However, there is a weakness of the Gabor filter in which the even symmetric filter will have a DC component whenever the bandwidth is larger than one octave [106]. To overcome this disadvantage, a type of Gabor filter known as log-Gabor filter, which is Gaussian on a logarithmic scale, can be used to produce zero DC components for any bandwidth [106], [109]. The log-Gabor filters are obtained by multiplying the radial

and the angular components together where each even and odd symmetric pair of log-Gabor filters comprises a complex log-Gabor filter at one scale. The frequency response of a log-Gabor filter is given as

$$G(f) = \exp\left(-\left(\log\left(\frac{f}{f_0}\right)\right)^2 / 2\left(\log\left(\frac{B}{f_0}\right)\right)^2\right) \quad (5)$$

where  $f_0$  is the centre frequency, and  $B$  provides the bandwidth of the filter. In order to extract the discriminating features from the iris area, the normalized pattern is convolved with 1D log-Gabor filters. First, the 2D normalized pattern is isolated into a number of 1D signals. Then these 1D signals are convolved with 1D log-Gabor filters to extract the distinctive features.

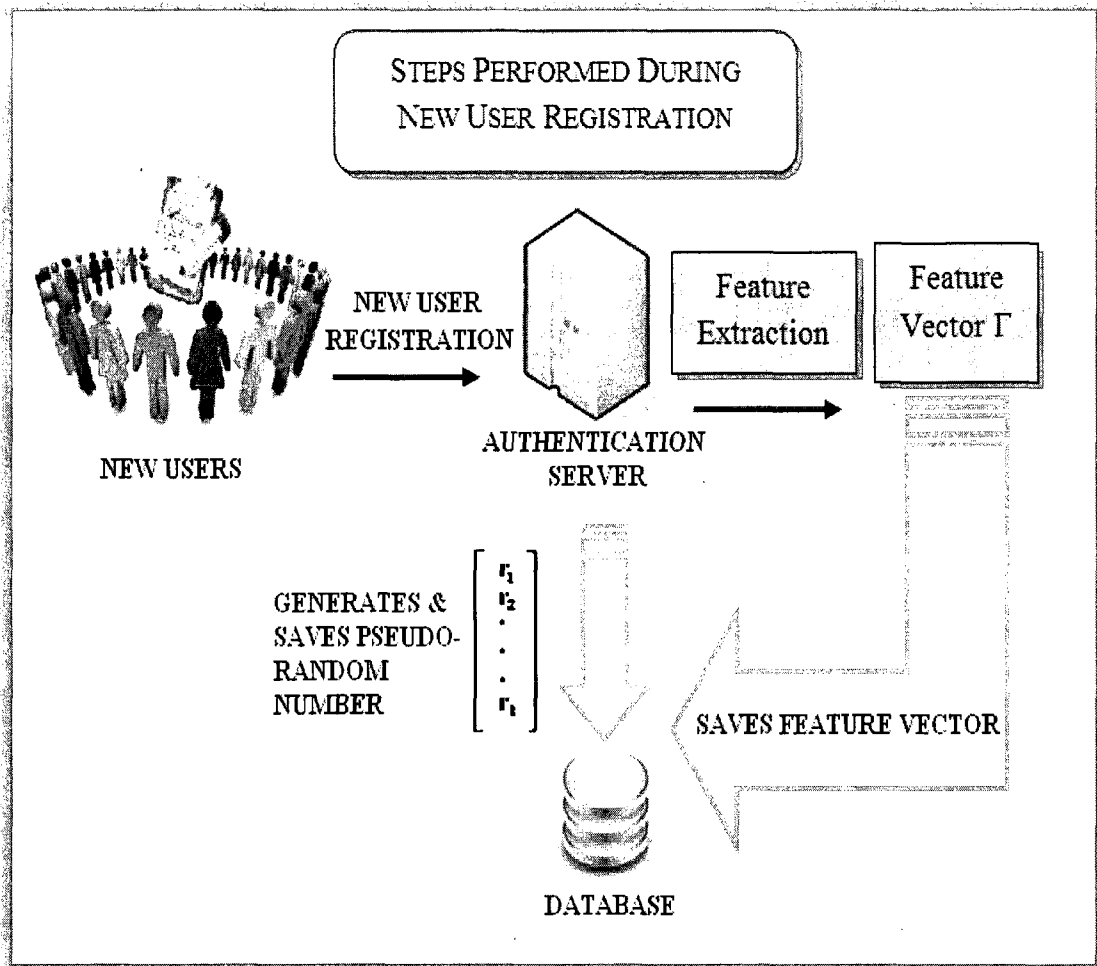
#### 4.4.2. Randomization using pseudo-random number

In previous implementations of BioHashing, it was observed that using the same random number (also known as Tokenized Random Number, TRN) for all users while generating the BioHash code gave inferior results when compared to the results obtained when a unique TRN is used for every user [110]. The authors in Refs. [73], [73], [79], [82], [105], [111-113] assumed that this TRN was the same number that was used during enrolment and also during verification, and that, different users had different TRNs, and showed excellent performance with zero equal error rate (EER). Obviously, this high performance of BioHashing was not completely based on biometric features alone, but had its lineage to this unique TRN

(distinct across users) as well. The authors suggested that this TRN be stored in a USB devices or smart cards by the users and presented to the authenticating terminal along with the biometric feature during user verification. Though this method produced a perfectly secure recognition system, it utilized TRN (possession factor) to accomplish this, which suffers from identical problem encountered while using tokens, e.g. it can be stolen, forgotten, copied or tampered. Moreover, in an ideal system, a user can use this unique TRN alone to authenticate one's own self with 100% accuracy, without necessitating any biometric matching at all. Hence, due to the above reasons, implementing BioHashing with user-owned TRNs reduces usability and weakens security, but without it, accuracy of the system falls. In this paper, we propose a user-friendly architecture, in which, we use unique TRNs for every user during BioHashing, but without the users having to remember or store it in safe devices. Instead, the server sends it to the client terminal for the client application to perform BioHashing on the user's captured iris image.

During new user registration, the server generates a TRN for every user and stores it in its database. (see Figure 4.8). When a user (username  $U$ ) tries to authenticate into the system, the server transmits the pre-stored TRN corresponding to  $U$  to the client terminal. Thus every time a user requests for authentication, the server extracts the corresponding TRN and transmits it to the client terminal, where it uses the TRN to perform BioHash of the captured image of the user and then sends it to the server for authentication (see Figure 4.9). The same TRN is used at the server's end to generate BioHash code from the stored template in the server's database.

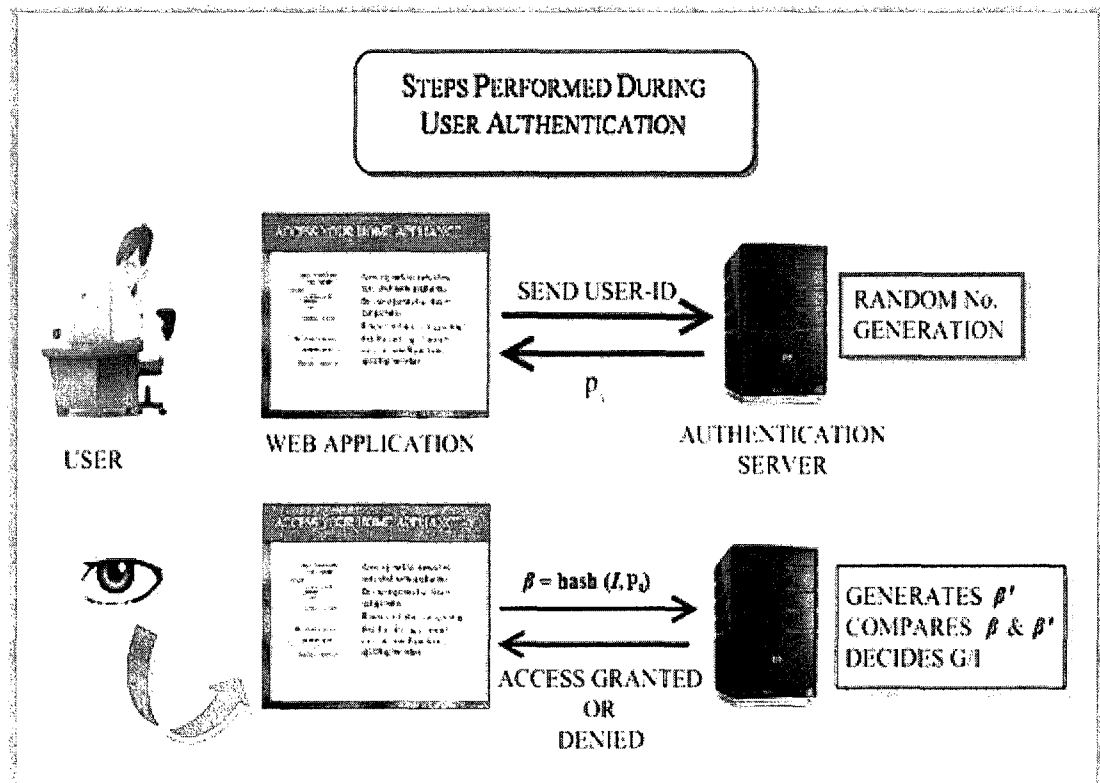
In this design, if an adversary monitors the messages being exchanged between the server and client, he/she will be able to acquire the TRN being sent to



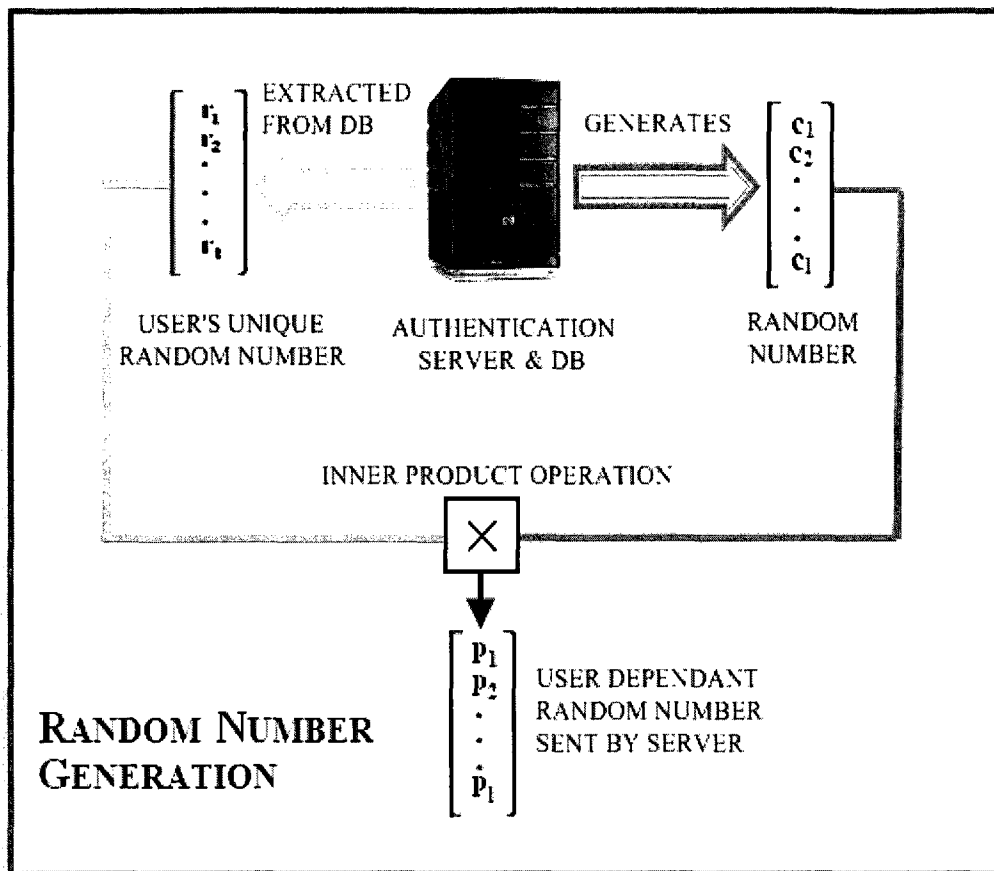
**Figure 4.8:** Procedure of new user registration in our proposed architecture

the client, and perform the same hashing procedure as being done by the client terminal, if he possesses an ill-gotten iris image of a valid user. This is possible because most of the feature extraction procedures used for feature extraction in iris images are publicly available and can be adopted or developed by an adversary with considerable time and effort.

So though a random number like TRN perfectly randomizes the BioHash code, it fails to prevent a courageous adversary from replicating this procedure or prevent phishing attacks. Hence, we randomize this TRN at the server by mixing the TRN ( $r$ ) with another random number array ( $c$ ) and then send this randomized TRN, called Random Token, RT (mathematically represented by  $p$ ) to the client terminal. The randomization is done in such a way, so that 'RT' (or  $p$ ) does not lose its individuality rendered by its constituent  $r$ , and simultaneously is random enough (due to  $c$ ) so that 'RT' (or  $p$ ) never takes the same value for a particular user  $U$  (see Figure 4.9 and Figure 4.10). This randomization is done using a simple mathematical operation like inner product with a similar sized random array  $c$ ,



**Figure 4.9:** Steps performed during user authentication using our proposed architecture



**Figure 4.10:** The process of Random Number generation during the user authentication process

which alters the TRN without severe time consumption or computational complexity, and without the adversary knowing about it, which ultimately helps in strengthening the security of the system.

Thus, we first implement feature extraction of the iris images and then transform the unique token TRN stored in the server into a random token RT, and then perform BioHashing on the extracted features with this modified RT (see Figure 4.9 and Figure 4.10). This procedure effectively reduces the size of the resulting unique iris feature data, called BioHash code that is later used to



authenticate a user. This hashing mechanism makes it unfeasible for any attacker to recover the actual iris feature data from the authentication code (see *Section 7.4* in Chapter 7) or replicate this whole procedure.

Table 4.2: Variables used in the User interaction with Server during authentication

Registered User ID = $U_v$	Imposter conjured User ID = $U_i$
Biometric data submitted by a valid user = $B_v$	Biometric data stored in the database corresponding to user $U = B_D$
Biometric data of an imposter = $B_i$	
Random Token $p_i = \text{TRN } r \times c$ , where $c$ is a small random number	

Hashing mechanism in Client and Server (using the variables defined in Table 4.2):

- 1) Server  $S$  generates  $l$  orthonormal pseudo-random array of numbers  $r$ ,  $\{r_{\perp i} \in \mathbb{R}^n \mid i = 1, \dots, l\}$  and  $l \ll n$  at the time of user registration, called Tokenized Random Number (TRN), which is unique to each user and stored in the server database corresponding to that user's record (see Figure 4.6).
- 2) At the time of user authentication, server  $S$  sends a random token (RT)  $p$  to the user  $U$ 's terminal;  $\{p = r_{\perp i} \pm c_{\perp i} \mid i = 1, \dots, l\}$ , where  $\{r_{\perp i} \in \mathbb{R}^n \mid i = 1, \dots, l\}$ ,  $l \ll n$ ,  $r$  being the orthonormal pseudo-random number corresponding to that particular user  $U$ , and  $c$  is a random seed generated by the server at that instant  $\{c_{\perp i} \in \mathbb{R}^n \mid i = 1, \dots, l\}$  (see Figure 4.9). *The random seed  $c$  ensures that each time a one-time password is generated from*

*TRN, it is random enough to prevent an adversary from executing a replay attack. Whereas the base key  $r$  ensures that each  $p$  is unique enough to help make the BioHash relatively unique for each user, though the hash key that is sent to a particular user for hashing is never same.*

- 3) Terminal  $U$  extracts biometric features ( $\Gamma$ ) from the iris image  $B_V$  which can be represented by the vector form  $\Gamma \in \mathfrak{R}^n$ , where  $n$  denotes the length of the feature,  $n \gg l$  (see Figure 4.9).
- 4)  $U$  calculates BioHash code  $\beta$ ;  $\{\beta = \langle \Gamma | p_{\perp i} \rangle | i = 1, \dots, l\}$ , where  $\langle . | . \rangle$  represents an inner product operation.
- 5)  $S$  extracts biometric features ( $\Gamma'$ ) from the iris image  $B_D$  stored in its database corresponding to the user  $U$ , which can be represented by the vector form  $\Gamma' \in \mathfrak{R}^n$ , where  $n$  denotes the length of the feature.
- 6)  $S$  computes BioHash code  $\beta'$ ;  $\{\beta' = \langle \Gamma' | p_{\perp i} \rangle | i = 1, \dots, l\}$ , where  $\langle . | . \rangle$  represents mathematical inner product operation.
- 7)  $S$  compares  $\beta$  &  $\beta'$  and decides the authenticity of the user.

If  $|\beta - \beta'| < \mu$ , where  $\mu$  is the threshold, the user is considered to be GENUINE; else, an IMPOSTER.

There can be multiple combinations of this User ID and biometric data between a user/imposter and the authenticating server. These different scenarios are listed in the next section, *Section 4.4.3*.

#### 4.4.3. User and Authentication Server Interaction Scenarios

##### ***Case 1: Valid User $\{U_V, \beta (= B_V, p_i)\}$ vs. Registered User $\{U_V, \beta' (= B_D, p_i)\}$***

User  $U_V$  first enters his/her user id in the application, which is sent to the server for checking whether this user is a registered user or not. Since we assume that user id  $U_V$  exists, the server retrieves the random number  $r_i$  stored in its database corresponding to  $U_V$  and generates  $p_i (= r_i \times c_i)$ , where  $c_i$  is a random array generated then by the server. This partly random number  $p_i$  is sent to the user terminal. The user now submits his/her iris image ( $B_V$ ) using the computer's camera after which, the application sends  $\beta = \text{hash}(B_V, p_i)$  to the server for authentication with the registered user's data,  $\beta' = \text{hash}(B_D, p_i)$ . The application compares  $\beta$  &  $\beta'$  and returns a matching score using Hausdorff distance with two possible outcomes: "true acceptance" with probability P or "false rejection" with probability (1- P).

##### ***Case 2: Imposter $\{U_I\}$ vs. Registered User $\{U_V\}$***

User  $U_I$ , who is actually an imposter, first submits his/her user id to the server, which checks whether this user is valid or not. Since we assume that user id  $U_I$  does not exist, the server terminates the connection and waits for other users requesting access.

##### ***Case 3: Valid User ID & imposter's biometrics $\{U_V, \beta (= B_I, p_i)\}$ vs. Registered User $\{U_V, \beta' (= B_D, p_i)\}$***

An imposter, who actually knows a valid user id  $U_V$ , first submits this id to the server, which confirms this as a legitimate user. As a result, the server sends the

random number  $p_i (= r_i \times c_i)$  to the user terminal for BioHashing. The imposter at this time submits his/her biometric data ( $B_i$ ) through the camera and the application sends  $\beta = \text{hash}(B_i, p_i)$  to the server for authentication with the registered user's data,  $\beta' = \text{hash}(B_D, p_i)$ . The application compares these two data and returns a matching score using Hausdorff distance with two possible outcomes: "false acceptance" with probability K or "true rejection" with probability (1- K).

***Case 4: Replay attack with a valid User ID and biometric data  $\{U_v, \beta_{rec} (= B_v, p_k)\}$  vs.  $\{U_v, \beta' (= B_v, p_i)\}$***

The imposter secretly records the user id and the biometric of a valid user during an ongoing authentication. Later, he/she submits this recorded user id  $U_v$  to the server, which finds this user id valid and sends the corresponding random number  $p_i (= r_i \times c_i)$  to the user terminal. But, the imposter submits the recorded hash data  $\beta_{rec} = \text{hash}(B_v, p_k)$  to the server. This is compared with the registered user's data,  $\beta' = \text{hash}(B_D, p_i)$ . The application compares these two data and returns a matching score using Hausdorff distance with two possible outcomes: "false acceptance" with probability P or "true rejection" with probability (1- P).

A user or an imposter can interact with a server for user authentication in these above described four ways with different accuracy of recognition in each of these scenarios, shown in Chapter 7 *Section 7.3*.

## **Chapter 5**

# **Authentication Server**

### **5.1. An Overview**

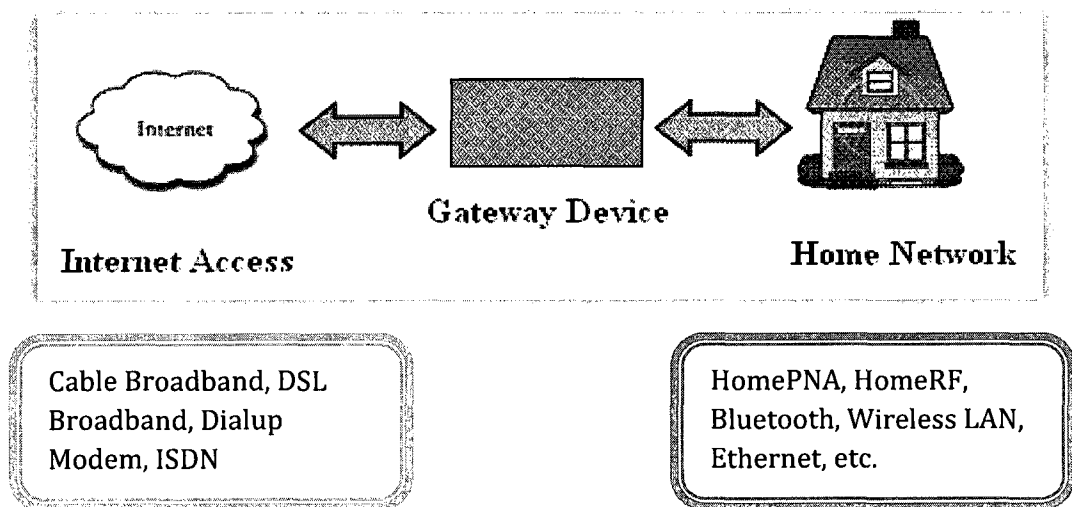
Over the last two decades, Internet is increasingly gaining significance and popularity in our daily lives and as a result, almost every household is now connected to it. Simultaneously, various home networking technologies gained great importance and approval from consumers that helped in developing interoperability, sharing of services and exchange of information between different electronic devices at home. Here, with the term interoperability, we primarily mean the ability of a single device or appliance to be able to discover and use the services or functionality of other devices or appliances that are connected to the same network as the first device. Research on this ubiquitous environment also enriched the study on remote control of home appliances [114] via Internet, or phone (including text messages and voice) [10]. In this research, we implement the architecture for remote access to networked home appliances via the widely available Internet in a very secure and user-friendly manner.

Two of the main components of the architecture for remote access to home appliances are home gateway and home server. We discuss briefly about the home gateway in this chapter in *Section 5.2*. A lot of research has already been done on the architecture and location of the home server, and many techniques have been proposed to implement it such that it helps in remote access to the home appliances. In *Section 5.3.1*, *5.3.2* and *5.3.3*, we briefly describe some of the existing schemes proposed for using this home server. In each of these sections we discuss simultaneously the problems associated with the design that leads us to devise a better framework in order to realize an efficient and easy access to the home appliances over the Internet. We illustrate our proposed architecture in *Section 5.4 (a)*, and describe the advantages of using it as compared to the previous endeavors in *Section 5.4 (b)*.

## **5.2. Home Gateway**

One of the main components in all the various designs proposed for access to home appliances remotely via Internet is the use of a gateway device, known as home gateway (HG) or Residential Gateway (RG). This device, which must be located in each home's premise, is responsible for enabling communication between the information appliances (appliances that can perform some specific user-friendly function) inside the home on one side, and the world-wide Internet on the other side (as shown in Figure 5.1). It has routing capability that assists it to communicate with the home network inside the home and is equipped with a Network Interface Card (NIC) that allows it to communicate with the Internet [8], [115]. For this

reason, an HG can successfully transfer messages from any home network inside a home to the Internet and vice versa. It thus does the bridging or routing, protocol and address translation between the external broadband network, i.e. the Internet and the internal home network . It is also responsible for maintaining the Quality of Service (QoS) in the home network depending on the user requirement. If multiple users are accessing the Internet in a particular home at the same time, the HG is



**Figure 5.1:** Access to Home Network using a Home Gateway

responsible for ensuring a smooth operation of all the different connections. If for some reason, like overload on the Internet connection, any of the service(s) has to be delayed; HG makes this decision based on the priority of the services running at that time. Thus, if a teleconference call is set as a high priority service with the HG and broadband TV as a low priority service, in situations of bandwidth contention when both the services are running, the HG will try to provide highest quality of service to the conference call, while compromising on broadband TV with lower

priority. Thus essentially, a HG provides the necessary connectivity features that enable users to exploit the advantages of a networked home with simultaneous connection to the Internet.

In recent years, HG comes with auto-configuration, multiple interfaces, abundant features, powerful functions and more user friendly interfaces [114], [116]. There are various home gateways available in the market from numerous vendors such as SOHO router home gateway, Cable Router Home gateway, Digital STB Home Gateway, etc. from which the users can chose the one they need for their remote home access implementation.

### **5.3. Popular Home Servers**

A home gateway alone is not powerful enough to perform all the desired operations like user authentication, storing home appliance information and user preferences, storing user privileges, maintaining access logs, etc. Hence, a server is needed that would provide these required services like user authentication, recognized access to home appliances, and store customization information concerning the way users want to access their home appliances according to their preferences and situation. Many attempts were made in the past to implement this server for remote access to home appliances, most of which had severe shortfalls due to which they were impracticable. The very first attempt was made using a dedicated home server placed inside the premises of every home [2]. A personal home server was also proposed in 2003 to provide a personalized experience to every home user while accessing the home appliances. Later, Internet Home Server



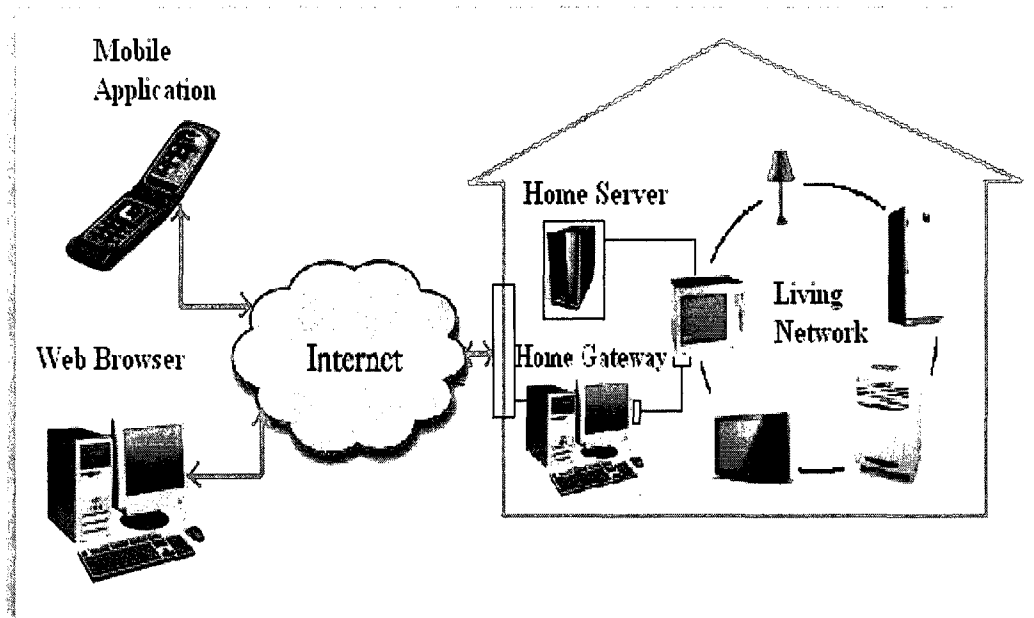
(IHS) made a promising implementation [2] in the similar milieu. But all these designs have unresolved problems that stalled their popularity and made them unfit to be used for large scale commercial use. Later in 2007, Remote Management Server (RMS) was proposed, which promised to solve the problems faced in earlier attempts though being proposed for accessing a home's network security system [6]. In the next sections, we discuss in detail the design highlights and flaws of each of these servers and discuss the reasons for the design of a better server in order to facilitate access to home network (and ultimately the home appliances) over the Internet. Our proposed approach can be extremely helpful in building future remote controlled, secure access to home network with minimum knowledge requirement for setting up and maintaining the server, minimal responsibility on the user(s), least cost and increased usability.

### **5.3.1. Home Server**

#### ***a. Overview:***

A key element that directs appliance control requests from user(s) over the Internet to the target appliance(s) in a language comprehensible to that particular appliance is the Home Server (HS). This server is responsible for connecting and managing all the information appliances in the home [3], [9], [117]. It can not only be used to configure the user experience according to the necessities of the different members in the house, but can also be used to store valuable information about the appliances like services available, if any appliance is already in use, etc. which makes it easier to control the appliances remotely. This way, users can obtain a variety of

information about the appliances connected to a particular home network with their current status [2]. Along with this, the home server offers a graphical user interface



**Figure 5.2:** A traditional home network including both a home server and a home gateway

(GUI) very similar to that of the actual appliances, in order to enable the user to control an appliance in the same manner in which he/she would actually control it in person.

In Figure 5.2, we see a traditional and generic home network consisting of a number of information appliances, like lamp, microwave, fridge, etc. along with a home server (HS) and a home gateway (HG) connected to the network as discussed earlier. User(s) access, controls and/or monitors their appliances through this HS. At the very beginning, the user's access request received via a web browser reaches the HG (after travelling over the Internet) and then it is directed to the HS of the requested home (by the Authentication Server to be discussed in details in *Section*

5.4). The HS first determines the target appliance from the user-sent message and then, it easily forwards the user command to the appropriate appliance (since it is connected to all the home appliances through the home network).

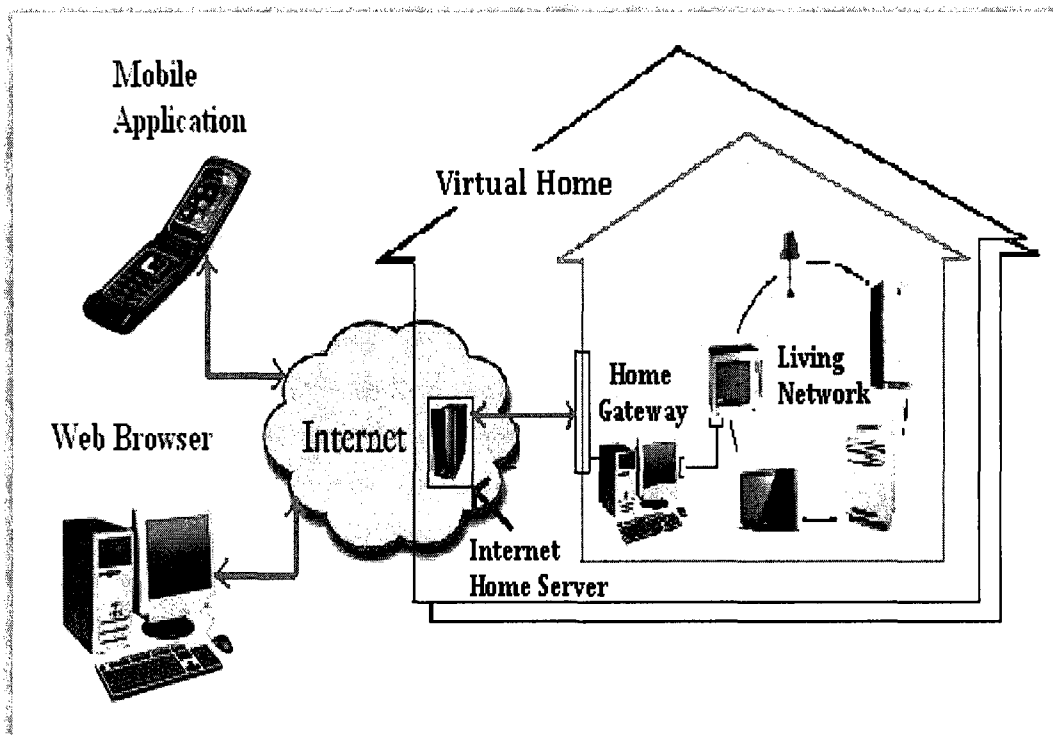
***b. Associated Problems:***

The first problem faced while using HS is that, Internet Service Providers (ISPs) usually allocates Internet Protocol (IP) addresses dynamically to the home gateway of a home from their free address pool. This IP address usually changes from one connection to the other [2], [6] depending on the addresses that are free with the ISPs at that point in time. Thus, every time the HG of a particular home boots up or connects to the Internet, it receives a new IP address. This dynamic IP address of HG makes it difficult to identify that particular home (reachable only through it's HG) from the Internet. Buying a fixed IP address for the HG from the service provider can resolve this problem, but this again leads to an increased cost for the user and renders this design less practical. Secondly, a home gateway may have firewall or proxies installed. This causes problems in accessing it from the World Wide Web as it becomes difficult for the firewall to determine which connections to block and which one to allow into the network. Thirdly, the user needs to be knowledgeable enough to be able to configure, add, update, delete user accounts, install the server and take proper backups, etc. which in turn increases user responsibility and burden. These problems decreased the acceptability of this design and hence, gave rise to the need of a better server design discussed in the next section, *Section 5.4*.

### 5.3.2. Internet Home Server

#### a. Overview:

Internet Home Server (IHS) was proposed in 2002 as an improvement to home server discussed in the previous section [4]. This server was implemented to control a LG washing machine via Internet [4, 10]. In this design, the server is placed outside each home in contrast to HS discussed previously, where the server was placed inside each home premise (as shown in Figure 5.3). When a user tries to



**Figure 5.3:** A virtual home using Internet Home Server (IHS)

access any home appliance, the command messages are first sent to the IHS that is directly connected to the Internet. Later, the IHS transfers these messages to the HG for the target home over the Internet. In this design, there still existed the problem of dynamic allocation of free IP addresses, where the network service providers

assign different IP addresses to new connections in HG (as faced in previous HS implementation). The authors of IHS devised a way to eradicate this problem by preserving the connection between the HG and the IHS at all times. This way, the live connection would ensure that the IP address once obtained by the HG never changes and hence the major problem of varying IP is solved.

***b. Associated Problems:***

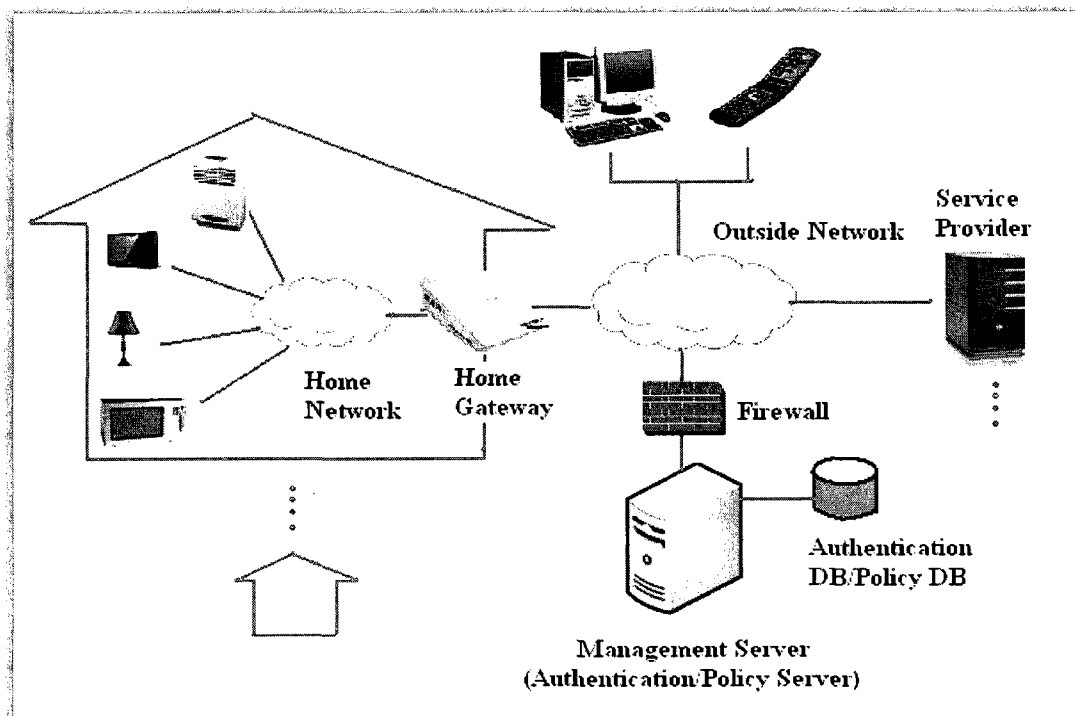
The above design of IHS fails to be practical for many reasons. Firstly, like HS, it becomes incredibly expensive for every home to deploy a dedicated server, not to mention the additional components the users need to buy to complete this initiative. Secondly, if we implement user authentication functionality in IHS, a database would be required to store user specific information and thus, it would further necessitate periodical maintenance of this database by the users. Special care would be needed to keep this database secure, as it will store sensitive user data. Along with this, any add, update or delete to this database would be the home's users' direct responsibility and any breach in its safekeeping will ruin all the other measures implemented for security. Thus, the security of this IHS and the whole application will be entrusted in the hands of the users using it, which is unrealistic. Thirdly, a live connection between the IHS and HG for maintaining a single IP address for HG is difficult to maintain over a long period of time. If this connection fails at any point of time due to reasons like accidental shutdown, device crash, power failure, any kind of malfunctioning or accidental reboot of either one of the devices, then it will again become difficult and complicated to uniquely identify the

HG and thus the home from the Internet. Fourthly, like the previous approach, specialized knowledge from the user is necessary to configure and maintain this application architecture. Hence, for all these reasons, a better design for a home server was felt.

### 5.3.3. Remote Management Server

#### *a. Overview:*

As more and more researchers started taking interest in the domain of home servers, better architectures were developed that gradually improved the designs created previously. Very recently in 2007, a new server, called the Remote Management Server (RMS) was proposed for accessing home security system



**Figure 5.4:** Remote authentication using Remote Management Server (RMS)

remotely [10] as shown in Figure 5.4. However, the design of this remote management server seemed to solve many of the problems faced in the above two efforts relating to remote access of home appliances. This RMS is used to authenticate users when they want to connect to their home security system installed inside their home. The server database stores user data and authentication policies that are used during the user authentication procedure.

In this design, the authentication and policy server is placed outside the house and shared among multiple houses or a housing society. This helps in minimizing the cost for using this application as, it can now be shared among the multiple apartments using its service. It also centralized the whole user authentication mechanism which greatly reduced the burden on users to maintain the database and security of this server. According to the design of this system, the web browser first accesses a home gateway with the user id and login password, which in turn directs this information to the RMS for authentication of the user. When the user is successfully authenticated, the information devices inside the home network are allowed to be accessed by the user.

***b. Associated Problems:***

This design though solved many problems that surfaced in the previous servers, suffered from many problems of its own. The major problem with this design is that, it failed to resolve the problem of dynamic IP, which still exists because since there are multiple home gateways being catered by a single IHS through the Internet, every home gateway may get a different IP address from the

through the Internet, every home gateway may get a different IP address from the ISPs whenever it connects to the Internet. In fact, according to this architecture, users will face difficulty in uniquely identifying the home gateway of the target home from the Internet at the very beginning of the interaction. The web browser needs to know the IP of the home gateway to access it over the Internet, which as explained earlier is difficult and not feasible when the IP address changes for HG. Moreover, there is more number of interactions between the home gateway and the RMS consuming more bandwidth. This is because, the HG first receives the user request and then transfers it to the RMS. The RMS performs the necessary operations and returns the result to the HG telling it whether the user is valid or not. The HG now allows or rejects the user depending on the verdict it got from the RMS. Thus, here, the HG talks to the RMS for a longer period of time while authenticating only one user. This way, other users of that house will not be able to access the HG during this time, thus making it unavailable. Hence, because of the two above problems, this method becomes unfit for real life implementation of remote access to the home network, though it successfully solves other issues found in HS and IHS designs like lowering of user responsibility, easy installation and maintenance, and minimizing implementation cost.



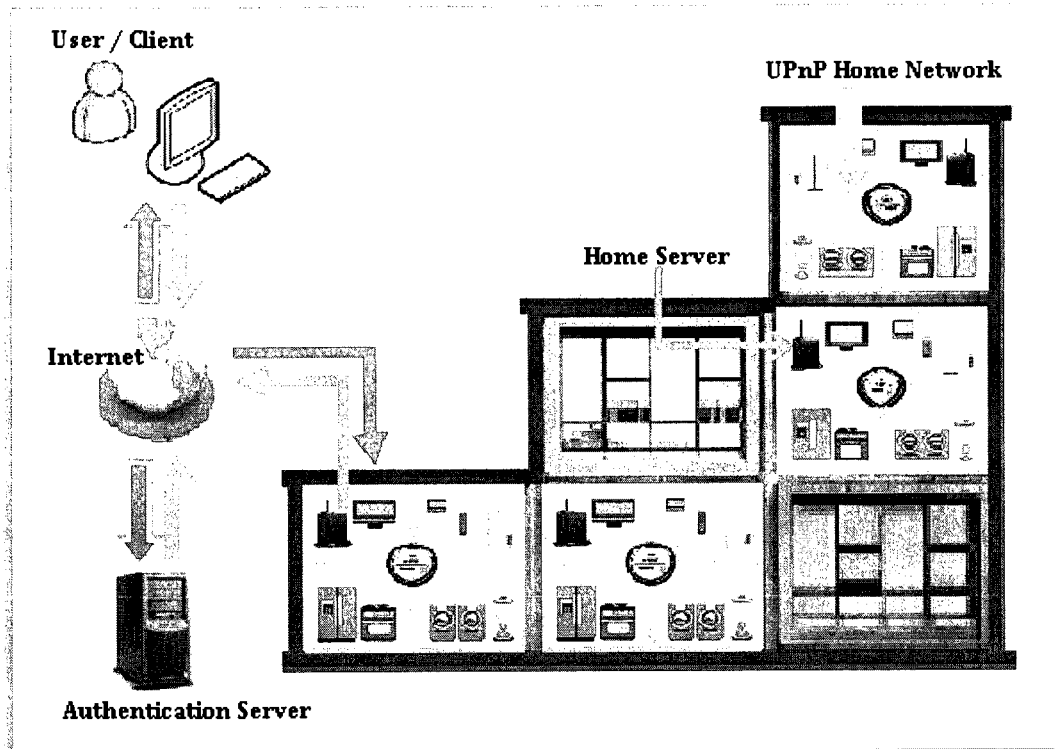
## 5.4. Proposed Authentication Server

### *a. Overview:*

Before proposing a new design for a new practicable home server, it is important to determine the most significant attributes that the server should realize. The following attributes lists few of the desirable properties that should be present in a perfect home server, but not just limited to them:

- a) Minimum user inconvenience while using the server,
- b) Minimum knowledge required from the user,
- c) Modest establishment, operational and maintenance cost,
- d) Minimum time and energy required to maintain the hardware and software components associated with the system,
- e) Stable and reliable,
- f) Simple and capable of communicating using standard protocols with backward compatibility.

Integrating most of the above qualities in a single home server, we propose a new design, called the 'Authentication Server' (AS) [118], [119], which authenticates users, provides access control information, user preferences and quality of service for controlling home appliances in multiple residences similar to a housing complex, all being managed together as shown in Figure 5.5. This will work together with Session Initiation Protocol (SIP) and provide a user friendly, accessible, stable, and low cost implementation of the home server. According to our proposed

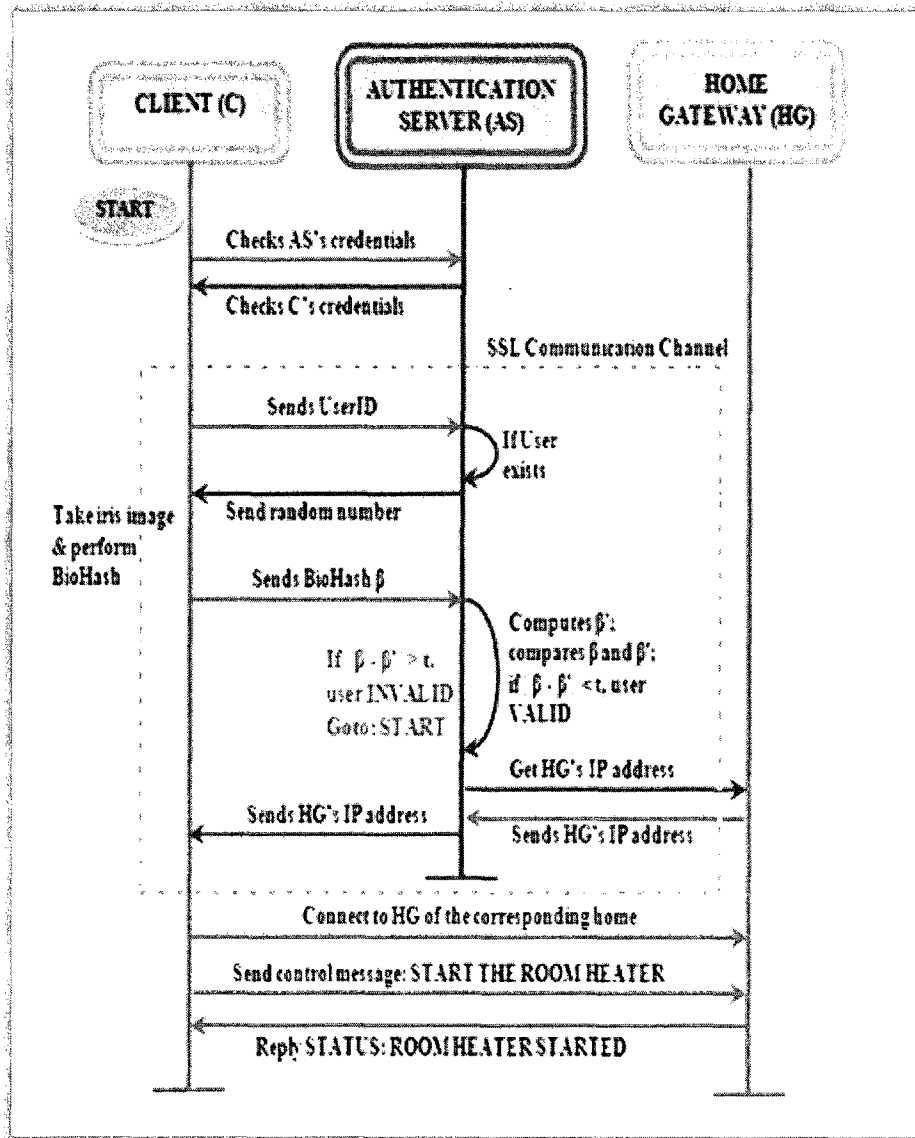


**Figure 5.5:** Authentication Server (AS) shared among multiple residences with Home Gateway (HG) in each residence

architecture, the client first accesses the Authentication Server (AS) that is easily visible and accessible over the Internet through an SSL connection (using trusted bookmarks). When the server verifies the existence of the user id sent by the client, it asks for the biometric data for user verification. On successful completion of user verification, the AS retrieves the IP address of the corresponding HG using SIP (as shown in Figure 5.6), and then transmits this data to the client application. The client application now closes the connection with the AS, and directly connects to the HG whose current IP address it has received from the AS. This frees the AS to respond to new requests from other users for the other apartments and also reduces the time and bandwidth consumed to take the unnecessary longer route via

the AS after user authentication. After successfully connecting with the HG, it starts transmitting command messages through small, encrypted packets using SIP and thus harnesses the advantages of SIP being an event driven protocol.

The Residential Gateway that is installed in each home acts both as a SIP user agent as well as a control point in UPnP network (discussed in details in Chapter 6);



**Figure 5.6:** Client - Authentication Server interaction diagram and finally, access to the Home Gateway (HG)

accepting SIP messages from the Internet (client) and unfolding them to reveal the UPnP messages to send it to the UPnP home network. The UPnP control point then delivers the message to the rightful target using its knowledge on the status and capabilities (i.e. services) of each device in the network, and asks that service to execute the necessary actions [120] according to the client request. The service in question then lets the control point know the results of the action(s) it took as to whether it was executed smoothly or met with an error through proper error messages and this status is then relayed back to the client application (in a user comprehensible language) after wrapping this message with SIP headers and encrypting it. The client application then allows the client to know the outcome of the request he made and thus determine his next course of action.

Since this AS will house the user-specific data for a large number of apartments, the size of the database will be large and a professional can be hired to maintain it and perform the necessary maintenance operations during its operational lifetime. All the apartments using it can effectively share the initial cost of buying, installing and maintaining it, relieving each user from spending huge amount of time and energy on this. Additionally, the database in AS (containing sensitive and personal user information) if stored in human readable format can cause identity theft and similar problems. Hence, it is best to secure the database using database encryption [31], but whether the whole database or specific sensitive columns should be encrypted is a design concern, beyond the scope of this research. Incorporating database encryption will hamper the performance of the system, and so, a tradeoff needs to be achieved about whether the whole database requires encryption or only

the sensitive columns [31]. This database is very critical to the access of home appliances remotely and should also be well protected from viruses, equipment failure, disaster, and accidental data loss or data corruption. It should have a firewall and an antivirus running in order to save it from harmful Trojans and viruses and regulate flow of traffic to the home networks. Proper encrypted database backups need to be taken at regular intervals manually or by scheduling jobs. Necessary add, update or delete operations on the user accounts can be performed in a centralized manner only when followed with valid requests from users requesting it.

***b. Advantages:***

Our proposed design has several advantages and it solves most of the problems encountered in the previous endeavors of constructing a home sever. We explain the major advantages in the following listing:

- a. This design requires just a single server to cater to the needs of multiple homes. Thus, several users from multiple homes can register with a single instance of AS and access their home appliances over the Internet.
- b. This design requires minimum knowledge of computers and no expert knowledge and hence can be very suitable for a wider and general consumer market.
- c. This architecture uses standard protocols like SIP and SSL to implement this complete design, both of which are backward compatible and have been popularly used for many years now. Hence this design can be easily

implemented at a large scale without much standardization problem and can be easily accessible to everyone.

- d. This design can reduce the cost of installation and maintenance of the authentication server since it can be shared among numerous users and the total cost can be shared accordingly.
- e. Since in this design, we have all the user information being stored at one place, maintaining and securing this database becomes centralized and easy.

## **5.5. Session Initiation Protocol**

Session Initiation Protocol (SIP) is a signaling protocol [121], [122] that is independent of the Transport Layer protocol being used (can be used with UDP, TCP, SCTP, etc.) and also does not depend on the type of session established. It was developed to set up, modify and tear down multimedia communication sessions such as voice and video calls over the IP [121], [122]. The protocol can be used for video conferencing, instant messaging and for playing multiplayer online games. The protocol can be used for setting up of two-party (unicast session) or multi-party sessions (multicast session). SIP, developed by the IETF in 1997 and standardized in 2002, contains elements of both Hyper Text Transport Protocol (HTTP) that is used for browsing the web and Simple Mail Transport Protocol (SMTP) that is mainly used for e-mails. It borrowed the client-server architecture and the use of Uniform Resource Identifier (URI) from HTTP and from SMTP, it mainly acquired the header style and text-encoding scheme.

Usually a SIP-enabled electronic device, called a user agent (UA), can create or receive SIP messages and thereby manage a SIP session. This UA can act as a server (called User Agent Server - UAS) and also as a client (called User Agent Client - UAC) or operate as both. Each resource in a SIP network is uniquely identifiable by a URI, which actually means that a single address is not tied up to a device in the network. Hence, this supports mobility of a device in the network as it can still be identified by its URI and not affected by changing of IP address. When the IP address of a device changes, registration procedure is used to allow this information to be automatically updated in the SIP network [121], [122].

A user can both use certificates or simple HTTP Digest [6] in order to authenticate a proxy server or a user agent [47] using SIP. The data and header information in each SIP message is encrypted using strong encryption algorithm in a hop-by-hop basis or end-to-end basis. The encryption of the header information makes it more difficult for the eavesdropper to figure out the location and/or commands sent by the user. Usually, TLS is used to encrypt messages in a hop-by-hop basis and S/MIME is used for an end-to-end encryption [123]. Both of these encryption techniques checks whether the message body has been modified while in transit between them.

We chose SIP to carry the user messages directly, from the user terminal to the Home Gateway (HG), because this way, the user's command messages will be able to harness the security of SIP and establish a secure connection to the home network. The messages from the HG are later directed to the appropriate appliance using UPnP messages inside the home network. HTTP is also a promising contender

protocol that can be used here instead of SIP mainly because of its simplicity and text based character [47]. However, it lacks some of the biggest advantages of SIP like event based notification and name-address resolution scheme like e-mail for which SIP supports users even in mobile environments. Moreover, SIP transmits messages using small encrypted packets that are better than HTTP, where message is transferred in plain text. Hence, we propose the use of SIP instead of HTTP between the user terminal and the home gateway after user successful authentication.

We propose using SIP and not SSL as the later uses a long handshaking protocol to just establish a connection between two terminals and will not be effective to transfer short command messages. Moreover, since SSL uses a dedicated connection for each user request to the HG, it will be difficult for multiple users to connect to a single HG at the same time. SIP on the other hand does not set up any connection for communication. It uses small encrypted packets to transfer data that in turn helps in reducing the load on the HG. Therefore, in this case, the HG simply needs to decipher each SIP packet and direct it to the target appliance without any connection setup. This is why, multiple users after being authenticated by the AS can send command messages to the HG and control home appliances simultaneously. For all these reasons, we conclude that SIP is a better choice over HTTP and SSL for sending command messages to a home gateway of a particular home from an authentication user via Internet.



## **Chapter 6**

# **Home Networks**

### **6.1. An Overview**

Computer networks have existed for more than thirty years, but only recently have they become popular in homes. Millions of homes all over the world have by now adopted home network in their homes due to the several advantages that it provides and the simplicity with which these can be set-up. A home network is basically a collection of networked home appliances such as computers, consumer audio-visual (A/V) devices, online gaming terminals, etc. that is connected to each other by a cable so that they can share a common interface and communicate between themselves. This network allows the connected devices to share information and computer resources like files and documents, printers and scanners, music systems, gaming systems, internet connection, various services from different devices, etc. and enable online communication. Network file sharing

gives users a lot of flexibility in storing data other than floppy drives or Zip drives. With this, sharing of files, pictures, music among various terminals becomes uncomplicated and taking backups gets trouble-free. Sharing of a printer between a computer, a laptop and a digital camera that are all connected in a home network also becomes child's play. Similarly, sharing a single Internet connection between multiple terminals is possible without having to pay the ISP for multiple accounts. A home with small number of connected devices can have a single network while for a home with numerous devices, there can be several clusters, such as one cluster per floor or possibly one per room.

The method by which the different electronic devices and computers are connected to each other to form a network differs from one system to another. Wireless transmission or a cable can be used as a transmission media for these home networks. An adapter, popularly known as network interface card (NIC) is also needed that will connect these appliances to the network medium. A home network can be constructed with electronic intelligent appliances (devices with its own computing capability), smart objects (appliances that can be controlled remotely), and telecommunications (the products and systems that present content and information to the consumer), as well as the whole-home and subsystem controls (security, heating/cooling, lighting, and energy management) that are traditionally associated with home automation [4].

Currently, there are numerous wired and wireless home networking technologies available in the market. These networking technologies fall mainly into two groups: those involving physical interconnection and those involving a service

or application architecture [4], [7], [8], [120], [124], [125]. The former include technologies like X-10, HomePNA, HomeRF, Gigabit Ethernet, IEEE 1394, Bluetooth, etc. The latter group includes Home Audio/Video Interoperability (HAVi), Jini, Open Services Gateway Initiative (OSGi), Vesa Home Network and Universal Plug and Play (UPnP) [126]. Some standards use communication and control wiring, some embed signals in the powerline, some use radio frequency (RF) signals, and some use a combination of several methods. Some of the most important factors that drive the selection of these home networks are its ease of use (i.e. its complexity), reliability, its scalability, compliance to industry standards, any new wiring required or not, types of devices supported, etc. Since requiring completely new wiring for setting up a home network is a very unrealistic demand in a consumer market where other better options for the same function is available, we do not encourage selecting any network that requires new wiring. Thus we focus on technologies like Jini, HAVi, OSGi and UPnP [126-133], which requires service architecture. In this chapter, we discuss about the three most popular home networking technologies – Jini, HAVi and UPnP in *Section 6.1.1*, *6.1.2* and *6.2* respectively and later explain the advantages for selecting UPnP as our choice of home network in *Section 6.2 (c)*.

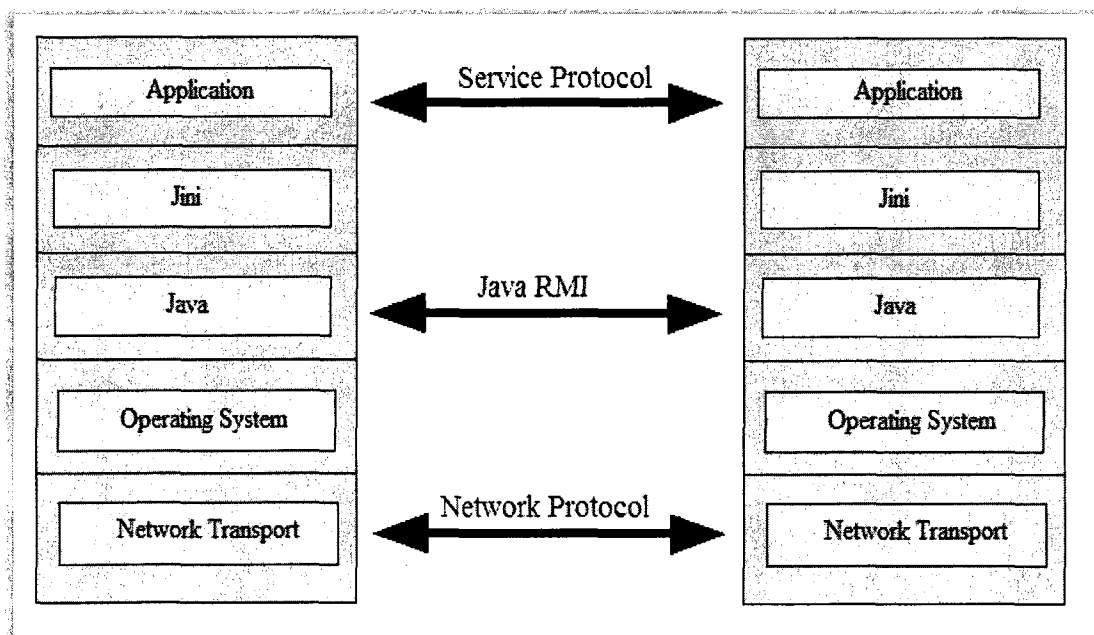
### **6.1.1. Jini**

#### **a. Overview:**

Jini was an initiative of Sun Microsystems [32] as a solution to easily usable home networking software. It is developed in Java that allows devices to plug directly into an existing home network without pre-installing drivers and

configuring operating systems. In this model, each device in the network provides services that can be used by other devices in the network. Jini provides the platform for clients or devices to locate the different services available in the network. Any user can use the Jini lookup service and request information about any other device. A look up service is used by this technology that allows it to first discover and then register the new device into the network. The interoperability of the devices in the network is maintained using Java Remote Method Invocation (Java RMI) [4], [133] (see Figure 6.1). RMI is a set of protocols that allows Java objects to communicate remotely with other Java objects.

When a new device joins the network, it goes through a discovery and join-in protocol. It first locates the lookup service through discovery and then uploads an object that implements all of its services' interfaces using the join protocol [4]. In order to consume a service, the device first locates the service using the lookup service and copies the object back to the requesting device that plans to use it. Thus, the lookup service acts as an intermediary that connects between the device



**Figure 6.1:** Architecture of Jini

requesting some service and the actual device providing the service.

*Jini uses the following key concepts to support a home network:*

- a. **Communities** – a group of services on a home network that are available for itself and other devices requesting those services.
- b. **Discovery** – this service can be used by devices to discover the presence of a particular service by some device on the network. The lookup service is used to discover the service wanted or keep track of shared services.
- c. **Lookup** – in order to consume a shared service, the application must first discover the presence of the service in the network. This is done by using the lookup service.
- d. **Leasing** – this method enables the Jini network to maintain updated and correct information about the component services and also a consistent means to free unused or unneeded resources. Rather than granting access to a resource or service for an unlimited period of time, Jini uses the concept of leasing it, or giving it for load for a specified amount of time. After this time, the loan would expire and the resources freed if not renewed before this time limit. If a service leaves the network intentionally or unintentionally, this leasing technique helps in freeing the unused resources and keeping the network updated.
- e. **Remote events** – An event is an object that contains information about some state change that happened in the network which might be useful to some other software component. It helps in updating a service already

present in the network about a new service that might in some way help it in better functioning.

- f. **Transactions** – various devices in the Jini network regularly communicate between each other for computational purposes. Transactions are a way to cluster a sequence of related operations that can have only two possible outcomes: either all these operations succeed or all of them fail. The advantage of using this transaction is to ensure data integrity in the home network.

***b. Associated Problems:***

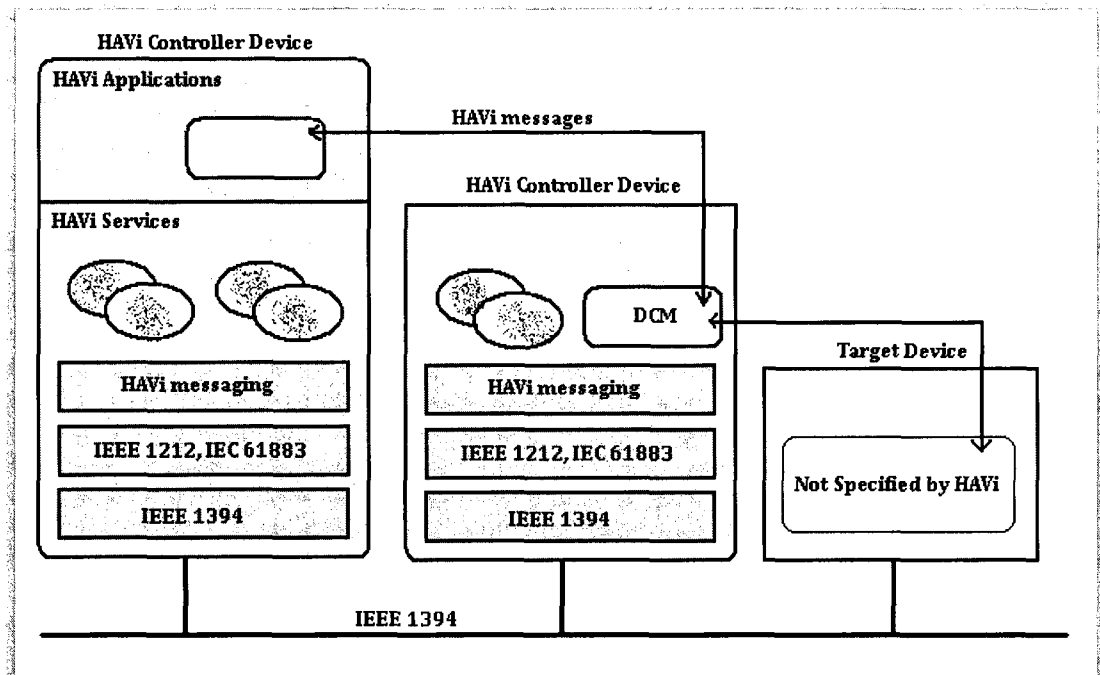
The main problem of Jini is that it is completely based on Java and so, consumes a lot of resources and computation power. Hence, Jini is not suitable for appliances which are low in memory and computational power.

### **6.1.2. HAVi**

***a. Overview:***

The home audio-video interoperability (HAVi) architecture specifies a set of application programming interfaces (APIs), services, and protocol that allows consumer electronics manufacturers and software engineering companies to develop applications for home networks based on different networking technologies. It facilitates seamless interoperability between multivendor consumer electronics devices and computing devices and simplifies the development of distributed applications on home networks [126], as shown in Figure 6.2. HAVi

provides a peer-to-peer environment that allows any device to detect, query, and



**Figure 6.2:** Communication in a HAVi network

control any other device, and groups of devices to cooperate thus providing improved functionalities and services. Its specification is based on Audio-Video (A/V) and has been designed and optimized to meet the demands of digital audio-video devices that requires higher bandwidth and has strict real-time constraints. The main characteristics of HAVi are [8]: (a) distributed control, (b) interoperability, (c) auto-installation and (d) upgrade capability.

It identifies each home networking service as an object that contains both data and the procedures to manipulate the data. This object is uniquely identified by an exclusive ID and name, and is easily accessible through interfaces that define them. All such objects are known to each other through a naming service called

registry that is present in HAVi. This registry is actually a distributed database that stores all information about HAVi objects, and thus, is often used by other objects to search for the presence of some particular objects.

The primary advantage of HAVi is that, unlike Jini and OSGi that are Java based technologies, HAVi is a platform independent and language independent solution. The middleware in HAVi can be implemented on a wide variety of hardware platforms together with digital products such as cable modems, integrated TVs, Internet TVs, set-top boxes, or intelligent storage devices for A/V content.

***b. Associated Problems:***

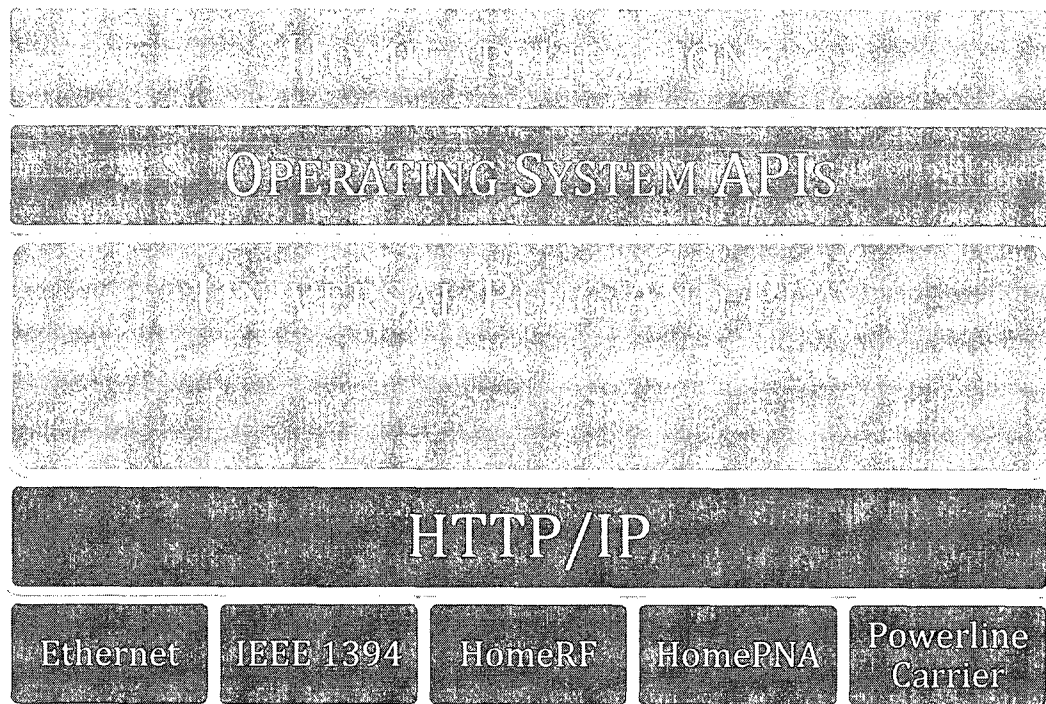
Popular home networking technologies like Jini, UPnP, OSGi, etc. controls a wide variety of home devices encompassing for example home security system, security cameras, lights, HVAC (heating, ventilation and air conditioning) systems. For these networks, streaming audio-video is not a primary design concern unlike HAVi, which is explicitly designed for home entertainment and audio-video devices. Thus, the scope of its implementation in residences diminishes as consumers will always like to interconnect not only their audio/video devices but other electronic devices also. hence, HAVi will not be the best choice of home network in such cases.

## **6.2. Universal Plug and Play**

***a. Overview:***



Universal Plug and Play (UPnP) was developed by Microsoft as their solution to home networking technologies. It uses open Internet communication standards to connect various consumer electronic devices and appliances to standard PCs. It has been developed over standard IP, HTTP and Extensible Markup Language (XML) and allows a new device to join the network dynamically, without any prior configuration and allows it to obtain an IP address, communicate its capabilities and



properties to the nearest control point, and also learn about the presence and features of other devices in that network [8], [126]. In the end, any device can leave a network smoothly and automatically without leaving behind any unwanted state information [129].

The architecture of UPnP is developed for pervasive peer-to-peer network connectivity of intelligent appliances, wireless devices and PCs enabling distributed and open communication [130], [134]. It's design objective is to bring user-friendly,

flexible, standard based connectivity to ad-hoc unmanaged networks, either in a home or public networks, or in business. Leveraging the universal IP protocol, UPnP can be used over most physical media including radio frequency, phone line, power line, coaxial, IrDA, Ethernet, and IEEE 1394 (see Figure 6.3). In other words, any medium that can be used to network devices together can enable UPnP [8]. It is compatible with existing networks, like standard 10 Base-T Ethernet and also with new networking technologies that do not require costly installation of new wiring systems in existing homes - HomePNA and HomeRF. Instead of concentrating on any one particular device type such as HAVi (focuses on audio/video devices), UPnP interconnects all types of devices in the home, including PCs, PC peripherals, new smart home appliances, gateway devices, home control systems, and Web-connectable devices. Thus implementing UPnP on an in-home network requires very little work and human intervention, which increases its usability. It is equally adaptable to both dynamic home environments and fixed configured corporate networks and hence, finds a huge scope for practical implementation. The main entity in UPnP network is the device. It offers functionality through services. Another important entity is Control Point (CP). It invokes actions on services with the required input parameters and receives the output if any and the return value. It is mainly responsible for discovering devices and invokes action on devices in order to facilitate interoperability [129].

UPnP classifies the devices in a home network into four categories:

1. **Control Point:** These are intelligent, active UPnP devices that host a set of software modules and are able to communicate and supervise a number of controlled devices. E.g. PC, laptop, PDA.
2. **Controlled Device:** These are the less intelligent, passive, UPnP devices that are capable of responding to a control point and perform an action. E.g. DVD, VCR, or automated light controller.
3. **UPnP Bridge:** These are the intelligent, multi-technology, multiprotocol UPnP devices that allow UPnP devices to communicate with legacy devices.
4. **Legacy Devices:** These are the devices that are not UPnP compliant or cannot participate in UPnP network because they do not have sufficient hardware resources or because the underlying media is unsuitable to run the TCP and HTTP protocols.

***b. Architecture:***

As stated above, UPnP operates via a set of open, standard, Internet-based protocols like TCP/IP, HTTP, etc. On top of these protocols, it performs six different activities namely addressing, discovery, description, control, eventing and presentation. With the help of these six operations, UPnP discovers a new device, knows its description and then adds it to the existing UPnP network without any human intervention for performing configuration.

When a new device is discovered by the UPnP home network, it assigns it an IP address using Dynamic Host Control Protocol (DHCP) or AutoIP. It then uses the SSDP module to advertise its services or functionalities to the control points in the

network. This advertisement carries a discovery message, which is an XML document, which contains the specification of the device including its type and ID and the Uniform Resource Locator (URL) of the device. The description also includes a list of any embedded devices or services, as well as URLs for control, eventing, and presentation. The descriptions of the services are also written in XML language and include a list of the commands, their actions, parameters and arguments required for each action, and lists of variables that represent the run time state of each service (e.g., data type, range, event characteristics). The UPnP control points use this device description document of the particular device to interact with it. When this device or its service document change, the control points need to be informed about the changes. For this reason, the service publishes updates to the network by the operation called eventing. It contains the names and the state of the variables and their current values, with the list of actions and responses. This way, all the control points receive the updates about a device and access it accordingly. In addition, the control points use URL of the devices to present its capabilities, so that, the user may control the device or view the status easily. This way, a UPnP network enables addition of new device into an existing network without any difficulty and prior configuration.

***c. Advantages:***

One of the biggest advantages of UPnP is its independence of the physical network in the house and as a result, does not require costly installation of new wiring for setting up this network. Unlike HAVi, UPnP does not focus on any

particular electronic device during networking and connects all types of PCs, networked appliances, gateway devices, audio-video devices, gaming systems, home control and home security devices smoothly. Moreover, since it is based on common internet protocols, it is compatible with a broad range of devices, which can seamlessly work together.

Secondly, another advantage of UPnP is that, it works well for both small and big networks. Thus, it is easy for a home with limited number of smart appliances to implement this, and later scale it to a larger network as they add new devices into the network. Thirdly, as its name suggests, it is a plug-n-play protocol, where adding a new device to an existing network does not require any prior configuration or installation of any drivers, mainly because it is based on a simple innovative mechanism of discovery and connectivity of devices. Thus, implementing this home network in any home requires very little user intervention and configuration which increases its usability and hence its acceptability among consumers.

One of the major constraints while developing a home network for different non-PC devices, i.e. for appliances, security cameras, lights, etc. is the availability of system resources. These consumer electronic appliances have radically less system resources than regular PCs and hence, they can support only those protocols that have very small footprint. UPnP is a perfect protocol to implement in these situations as it requires very small amount of system resources. Moreover, since it is platform and language independent, any type of operating system can be used to implement UPnP enabled devices [130]. A fifth very important advantage of UPnP network is that, since it is based on a peer-to-peer network architecture, the home

network can function flawlessly even without a personal computer, though its presence is advantageous, but not mandatory.

A sixth advantage of UPnP is that, it enables the control of a vendor over the user interface and interaction via a web browser. For all the above advantages, we adopted UPnP to control the home appliances for both small sized networks (apartment with fewer appliances) and giant ones (apartment with numerous appliances) in our architecture.

## **Chapter 7**

# **Experimental Results & Discussions**

### **7.1. An Overview**

Our architecture for remote access to home appliances over the Internet consists of four major components, the SSL communication channel, iris recognition with its consequent BioHashing and UPnP home network. Evaluating the performance of such a composite system can be done by evaluating the functioning of each of the component domains. We start our evaluation by discussing the principle present behind the client and server programs that we constructed to replicate a real world user and authentication server. We discuss the steps taken to implement client and server certificate checking in the program and demonstrate the code that actually did the certificate checking. We realize that the security and performance of the whole application depends mostly on the recognition accuracy of the BioHashing phase after which users are authenticated into the system. Therefore, we represent the performance of the whole system by evaluating the accuracy of recognition of the BioHashing phase only. We perform this evaluation in

two phases: first using support vector machines (SVMs) and then using Hausdorff distance in order to determine the extent of similarity between two iris images. While using Hausdorff distance for image comparison, we present the first results using different random numbers for different users during the BioHashing phase, that corresponds to Cases 1, 2 and 4 discussed in Chapter 4, *Section 4.4.2*. Following this, we present the results for Case 3 in *Section 4.4.2* Chapter 4, where we hash images of all users using a single array of random number.

## **7.2. Client Server Program**

In today's world, we need security in almost every web application on the World Wide Web and cryptography is one of the primary tools that have been popularly used to provide this security. The primary objective of cryptography is to provide confidentiality, integrity of data, authentication and non-repudiation [12] and can be used to thwart numerous types of network-based attacks like IP spoofing, eavesdropping, tampering, connection hijacking, etc. [12]. OpenSSL is a cryptographic library that provides implementation of the industry's best algorithms including 3DES (Tripple DES), AES, RSA, message digest algorithms, etc. [11], [12].

Implementing cryptographic algorithms is a very difficult task as it is not possible for a single developer to anticipate all the various kinds of attacks possible on the system and resist them accordingly. Even if the protocol algorithms are well developed and proven, implementation errors fails to ensure integrity of data. The main purpose of Secure Sockets Layer (SSL) or Transport Security Layer (TLS) is to



provide the common security features to any TCP based network connection that implements it, such that the need for cryptographic expertise is minimized in order to develop a secure connection. Thus SSL makes it easy to secure a network connection and does not even require any deep understanding of how the algorithm works. It just needs the developer to understand how to apply the algorithm properly.

The main purpose of OpenSSL library is to implement SSL and TLS protocols. It is a free implementation of SSL/TLS based on E. Young's SSLeay package [11]. We implemented this OpenSSL library version 0.9.8g and used it in two programs coded in C language, the client and server programs. In the next two sections, we discuss in detail the functioning of the client-server programs with few glimpses on how the

```
common.h |
#include <string.h>
#include <openssl/bio.h>
#include <openssl/err.h>
#include <openssl/rand.h>
#include <openssl/ssl.h>
#include <openssl/x509v3.h>
#include "reentrant.h"

#ifdef WIN32
#include <pthread.h>
#define THREAD_CC
#define THREAD_TYPE pthread_t
#define THREAD_CREATE(tid, entry, arg) pthread_create(&(tid), NULL, \
                                                    (entry), (arg))
```

**Figure 7.1:** Common.h program – a header file

code is arranged. Along with these two separate client-server programs, we used

two common programs called `common.h` and `common.c`. *Common.h* is basically a C language header file, which includes the required headers from OpenSSL.

The function *common.c* contains the error handlers which defines how an error should be handled if one occurs. It also defines the common initialization functions of OpenSSL like initialization of libraries, etc.

```
#include "common.h"

void handle_error(const char *file, int lineno, const char *msg)
{
    fprintf(stderr, "*** %s:%i %s\n", file, lineno, msg);
    ERR_print_errors_fp(stderr);
    exit(-1);
}

void init_OpenSSL(void)
{
    if (!THREAD_setup() || !SSL_library_init())
    {
        fprintf(stderr, "*** OpenSSL initialization failed!\n");
        exit(-1);
    }
    SSL_load_error_strings();
}
```

**Figure 7.2:** *Common.c* program – contains common functions of the client and server program

### 7.2.1. Client program (*SSLClient.c*)

We developed *SSLClient.c* to represent a client program or the user end in our application. This program first sets up an SSL connection with the server and then allows users to communicate with the server for exchanging user information for authentication. In most cases, SSL only authenticates the server, i.e. checks

whether the server possesses a valid certificate. But for reasons explained in Chapter 2 *Section 2.2*, we use both client and server certificate checks in order to

```
SSLClient.c
#include <string.h>
#include <openssl/sha.h>
#include "common.h"
#include <openssl/evp.h>

#define CIPHER_LIST "ALL:!ADH:!LOW:!EXP:!MD5:@STRENGTH"
#define CAFILE "rootcert.pem"
#define CADIR NULL
#define CERTFILE "client.pem"

SSL_CTX *setup_client_ctx(void)
{
    SSL_CTX *ctx;

    ctx = SSL_CTX_new(SSLv23_method( ));
    if (SSL_CTX_load_verify_locations(ctx, CAFILE, CADIR) != 1)
        int_error("Error loading CA file and/or directory");
    if (SSL_CTX_set_default_verify_paths(ctx) != 1)
        int_error("Error loading default CA file and/or directory");
    if (SSL_CTX_use_certificate_chain_file(ctx, CERTFILE) != 1)
```

**Figure 7.3:** A snapshot of SSLClient.c program

verify both the client and the server.

Below is our code for performing the client server authentication. The client first says hello to the server, which verifies if the user (represented by the User ID) exists or not. On successful verification, the client sends the biometric data of the user to the server on server's request.

Table 7.1: Interaction between the Client and Server program exchanging UserID

### SSLClient.c: Client Authentication Function

```

for (;;)
{
    if (!fgets(buf, sizeof(buf), stdin))
        continue;

    for (nwritten = 0; nwritten < sizeof(buf); nwritten += err)
    {
        if(isBioWritten == 0)
        {
            bufsslread[0] = '\0';
            err = SSL_write(ssl, buf + nwritten, sizeof(buf) -
nwritten);
        }

        nread = SSL_read(ssl, bufsslread, sizeof(bufsslread));
        if(nread > 0)
        {
            printf("%s \n", bufsslread);
            if(strstr(bufsslread,searchstr)!=NULL)
            {
                printf("Entering Biometric data ---- \n");
                if (!fgets(buf, sizeof(buf), stdin))
                    break;

                printf("\nPrinting from Buffer!: %i", buf);
                err = SSL_write(ssl, buf, sizeof(buf));
                isBioWritten = 1;

                if (err <= 0)
                    return 0;
            }

            if(strstr(bufsslread,searchnot)!=NULL) // searchnot =
"not"
            {
                goto end;
            }

            /*TO ENABLE THE CLIENT TO SEND MORE DATA AFTER WELCOME MESSAGE */
            if(strstr(bufsslread,welcome)!=NULL)
            {
                printf("\nEnter Application data ---- \n");
                if (!fgets(buf, sizeof(buf), stdin))
                    break;

                err = SSL_write(ssl, buf, sizeof(buf));
                isBioWritten = 1;
            }
        }
    }
}

```

```

        if (err <= 0)
        {
            printf("Error occurred!");
            return 0;
        }
    }
}

/** TO ENABLE THE CLIENT TO SEND MORE DATA AFTER WELCOME MESSAGE **/

    }
}
}

```

SSL APIs have provided several ways to check server certificates. We use the function `setup_client_ctx()` to check the validity of a server certificate. The following code represents the function.

Table 7.2: Checking the Server Certificate in the Client program (SSLClient.c)

### SSLClient.c: SSL Server Certificate Check Function

```

SSL_CTX *setup_client_ctx(void)
{
    SSL_CTX *ctx;
    ctx = SSL_CTX_new(SSLv23_method( ));
    if (SSL_CTX_load_verify_locations(ctx, CAFILE, CADIR) != 1)
        int_error("Error loading CA file and/or directory");
    if (SSL_CTX_set_default_verify_paths(ctx) != 1)
        int_error("Error loading default CA file and/or directory");
    if (SSL_CTX_use_certificate_chain_file(ctx, CERTFILE) != 1)
        int_error("Error loading certificate from file");
    if (SSL_CTX_use_PrivateKey_file(ctx, CERTFILE, SSL_FILETYPE_PEM) != 1)
        int_error("Error loading private key from file");
    SSL_CTX_set_verify(ctx, SSL_VERIFY_PEER, verify_callback);
    SSL_CTX_set_verify_depth(ctx, 4);
    SSL_CTX_set_options(ctx, SSL_OP_ALL|SSL_OP_NO_SSLv2);
    if (SSL_CTX_set_cipher_list(ctx, CIPHER_LIST) != 1)
        int_error("Error setting cipher list (no valid ciphers)");
    return ctx;
}

```

### 7.2.2. Server program (SSLServer.c)

The server program is similar to the client program for the most part. It opens an SSL connection and then waits for connectivity requests from users. When it receives a request, it first checks to see whether the user exists or not in its database. If it exists, it asks the user to send his/her biometric data. When it receives the biometric data, it sends this data to MATLAB to compare the image submitted by the user and the stored image corresponding to that user to decide whether the user is a valid one or not.

```
#include "common.h"
#include "math.h"

DH *dh512 = NULL;
DH *dh2048 = NULL;

# define SIZE_ARRAY 10000

void init_dhparams(void)
{
    BIO *bio;

    bio = BIO_new_file("dh512.pem", "r");
    if (!bio)
        int_error("Error opening file dh512.pem");
    dh512 = PEM_read_bio_DHparams(bio, NULL, NULL, NULL);
```

**Figure 7.4:** A snapshot of SSLServer.c program

The following code performs this interaction between the client and server.

Table 7.3: Client authentication in the Server program (SSLServer.c)

### SSLServer.c: Client Authentication Function

```

/***** START: To check for UserName in Data *****/
if(strstr(buf,searchstr)!=NULL)
{ isBioOK = 1;
  printf("%s \n", writedata); // "Hi Arpita. Enter BiometricData";
  wrote = SSL_write(ssl, writedata, strlen(writedata));
}
if(strstr(buf,applicationdata)!= NULL) //applicationdata =
"Application"
{ printf("%s \n", "Displaying Application Data from Client");
}
else if(isBioOK == 0 && isApplicationData!=1)
{
printf("%s \n", writedataFalse); // writedataFalse = "Enter Proper
User ID";
wrote = SSL_write(ssl, writedataFalse, sizeof(writedataFalse));
}
fph = fopen ("hausdorff.txt", "r" );

if (fph==NULL)
{
printf ("File error");
exit (1);
}
hv = fread(hvalue, sizeof(char), readsize, fph) ;
if (hv != readsize)
{
printf ("Reading error");
exit (3);
}
printf("Hausdorff Filedata: %02x \n", hvalue);
fclose (fph) ;
fpbd = fopen ( "bd.txt", "r" ) ;
bdv = fread(bdvalue, sizeof(char), readsize, fpbd) ;
bdvalue[bdv]='\0';
printf("BD Filedata: %f \n",bdvalue);
fclose (fpbd) ;
if (hvalue > 0)
{SSL_write(ssl, userverified, sizeof(userverified)); //User Verified.
printf("\n\n%s \n", userverified);
isApplicationData = 1;
goto applicationdata;
return 1;
}else {
printf("%s \n\n", usernotverified);
SSL_write(ssl, usernotverified,
sizeof(usernotverified)); //usernotverified
return 0;
}
}

```

The following code performs the client certificate check (assuming that the client already has a valid client-certificate):

Table 7.4: Checking the Client Certificate in the Server program (SSLServer.c)

### SSLServer.c: SSL Server Certificate Check Function

```
SSL_CTX *setup_server_ctx(void)
{
    SSL_CTX *ctx;
    ctx = SSL_CTX_new(SSLv23_method( ));
    if (SSL_CTX_load_verify_locations(ctx, CAFILE, CADIR) != 1)
        int_error("Error loading CA file and/or directory");
    if (SSL_CTX_set_default_verify_paths(ctx) != 1)
        int_error("Error loading default CA file and/or directory");
    if (SSL_CTX_use_certificate_chain_file(ctx, CERTFILE) != 1)
        int_error("Error loading certificate from file");
    if (SSL_CTX_use_PrivateKey_file(ctx, CERTFILE, SSL_FILETYPE_PEM) !=
1)
        int_error("Error loading private key from file");
    SSL_CTX_set_verify(ctx,
SSL_VERIFY_PEER|SSL_VERIFY_FAIL_IF_NO_PEER_CERT,
        verify_callback);
    SSL_CTX_set_verify_depth(ctx, 4);
    SSL_CTX_set_options(ctx, SSL_OP_ALL | SSL_OP_NO_SSLv2 |
        SSL_OP_SINGLE_DH_USE);
    SSL_CTX_set_tmp_dh_callback(ctx, tmp_dh_callback);
    if (SSL_CTX_set_cipher_list(ctx, CIPHER_LIST) != 1)
        int_error("Error setting cipher list (no valid ciphers)");
    return ctx;
}
```

### 7.2.3. Server and Client Certificate

When a client browser sends a request for an SSL page, the server replies back with its SSL certificate which has the public key of the server. The public key is actually a random number generated by the server as part of a key pair. The other pair of this key is the private key, which as the name suggests is known to all. But the private key is only possessed to the server. After the certificate verification of the client and server as described earlier in *Section 7.2.1* and *7.2.2*, the client



generates a master key or session key (as it is valid only for that session) and encrypts it with the server's public key which it possesses. This key is sent to the server, which it must decrypt using its corresponding private key. This way, both the sides have a shared key that is only known to them, and this is used to encrypt all the data that is exchanged between the two. The certificate contains credentials that the other side may use to determine the authenticity of each other and whether it can be trusted. Now, we will discuss about the certificate generation procedure. There are mainly three steps to generate this certificate and be able to use it:

- a) Generate the server's key pair (public and private keys)
- b) Generate a certificate signing request from the server and send it to a Certificate Authority (CA)
- c) Install the certificate that the CA sent

In our implementation, we created a self-signed root CA and signed the client certificates directly with this CA. This root CA represents the Certificate Authority for a company. We first created a certificate signing request using the command line utility [12] as shown in Figure 7.5.

```
-----BEGIN CERTIFICATE REQUEST-----  
MIIBhDCB7gIBADBFBMQswCQYDVQQGEwJBVTETMBEGA1UECBMKU29tZS1TdGF0ZTEh  
MB8GA1UEChMYSW50ZXJuZXQgV2lkZ210c3QtYQdHkgTHRkMIGfMA0GCSqGSIb3DQEB  
AQUAAAGNADCBiQKBgQct1DKptuqtKyOYJIVXMq2OYME4updenyWrtkIoL+Gafr52  
f5iMYAENO/aVcCXmAiM8zagWSvWSpZQN5PEj1FatRWgdK4pOLGIwJtxUSuJ4QLiE  
I7ax376KNO+Ez6D1yZN32Jq1FjfnYNR/Rpptx6yZTpW1FiLunGUILGFP+tNbYQID  
AQABoAAwDQYJKoZIhvcNAQEFBQADgYEAAo+SsE+07kKCENJAdZyJVUanezxLAYWk  
b/5rWFOpiUGd/tUvILb7QWSzmxuvE+HaBJhmjVCqzOnacOwx7kaILgnzy8sW0Jz8  
47KXTYjzVcX7RVcRjtRxzzMvi7AO6UHxk4H1F1VQkCkNwMa63XDwOtIutj59XA+o  
O3a68eUwhaA=  
-----END CERTIFICATE REQUEST-----
```

**Figure 7.5:** Certificate Request generated

All information like common name, validity period, location, etc. are included in the certificate request. The certificate request thus generated is then signed by the root CA to generate the server certificate as shown in Figure 7.7. Similarly, the client certificate is also generated with the same root CA. Later, during the SSL handshaking phase, these certificates are loaded from their respective locations and then checked to see whether they are valid and signed by a trusted CA, which here is our root CA.

```

-----BEGIN CERTIFICATE-----
MIICvTCCAiaGAWIBAgIJAj4c1X1DDBaVMAOGCSqGSIb3DQEBAQUAMEUxCzAJBgNV
BAYTAkFVMRMwEQYDVQQIEwpTb211LVNOYXR1MSEwHwYDVQQKEWhJbnRlcm5ldCBX
aWRnaXRzIFB0eSBMdGQwHhcNMDgwNzI1MDMxNTQwWhcNMDgwODIOMDMxNTQwWjBF
MQswCQYDVQQGEwJBVETETMBEGA1UECBMKU29tZS1TdGF0ZTEhMB8GA1UEChMYSW50
ZXJvZXQgV2lkZ210cyBQdHkgTHRkMIGfMAOGCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQDFeSKAACoyCU6z5gO+u53UmC5247+dmacPbItOkKqQF9NVfEJKziWq6YDjO/e7
pJB3vEmLqxsBwJsUyBX7PH+IAu5jHdVfdEetNanMWpJQ22Mr7ekvYvGrNRE5jXct
ZpEULQ75cybtUEcuApxx1NC6QzB44YfcKiFY8ssOmrAaQIDAQABo4GOMIGxMBOG
A1UdDgQWBBTQa1YW3QkfqX1pCOMnxO66hfkuuzB1BgNVHSMEbjBsgBTQa1YW3Qkf
qX1pCOMnxO66hfkuuzB1BgNVHSMGAgEwRTElMAkGA1UEBhMCQVUxEzARBgNVBAGTC1NvbWUu
U3RhdGUxITAfBgNVBAoTGE1udGVybWV0IFdpZGpdHMgUHR5IEExOZIIJAj4c1X1D
dBaVMAwGA1UdEwQFMAMBaf8wCwYDVROPAQDAgEGMAOGCSqGSIb3DQEBAQUAA4GB
AGPB3rlwgbrX5ygyBfTAMZSff8XKGXmavswB+vRMhY+1tSyPNerNOCG19i/U2ez/
aOn5hKE2fmj1gXDE1y8HSGNL9911/XIXOxyPbtcf4SnVddZ6C1PgC08+LzwyHypr
9N8kiOY2OjO7+NUBM6dFJWYfehUXYV+3H9+2c3vzUYe
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4, ENCRYPTED
DEK-Info: DES-EDE3-CBC,FEA8A1FEAB7DA142

eOuF5BLlq3WSGtdskQoo61+OIPBjJriIj4H5u0JULtG9Rxy7hA0vkG1SO+Dfa3AE
EE/sF9cYztf01TORNWnxkRPMoirPB2MnQE+uIvV2NMHa41AanyXicCO+7SmQFCAU
izgFTRPjttS53e2fIe1H6Kc9iOj185PxcjC4Vv/VbZ6YmzFZMChcr9Q/ch6DJzS3
axZlg11MAJsi9QOPou+NskJhkbPBZPm3txpBrfu566ZmhAckHjUx8BjStqDYgWjt
lAFo2MeoznkCmkbwSoMXwzAOTWUmYFzD+tHg+Dt6Fu2cx5itQQQFe/nq14Ok2ssq
XVOpN+OZ61ip6hXy6Pu+NMUucMOMP1EMSXm4jLep4PeS7m7LeSi+KTqmhXpfUOF5
r5U6ury9wgfCthp9i5zNQASqEfqTLA3qLroNb/7xFYWB0xyfV/mY3i2nHnviEdnZ
ICx+ENKjpyqJ2LuCRWeCTJncpLps7GNiDQxpFDX1H2GNKcPmzm+LVONqN3tb1Ok6
OXe0sqP11TRxN9H28lqRrJLAOG07BAUWH5DAEGbxtemuqhBNxuW9uhf63fZxkjr3
b/VgzDyhNzQgL9thVWDPKgRnPIcbUOFYAwBJOadqpEXqZIkTyn03kCnVX78ec1XB
g6q5dR+z79zLXPmoQPzqCYKj9YLDKKxwN9ao0x5E1JJsh80+Mf5GJZQarKSf9sde
T6g4zvmgKNVqABqOBUJ4p8cbioakHpv+MMW+wr6wdacvSF3+mR9F5QuddFvDeJMJ
TFFe5kPpXCr5VD1PMe6mfFSiPSzjq7sS/PVrs+GtoOVaUryguojRsw==
-----END RSA PRIVATE KEY-----

```

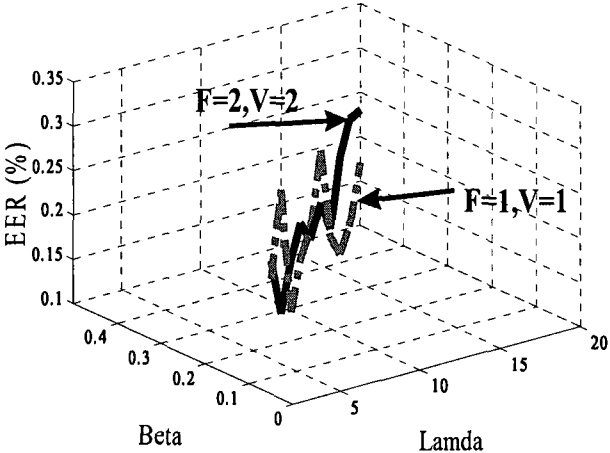
Figure 7.6: Server Certificate with both the private key and public key

## **7.3. Performance Evaluation**

### **7.3.1. Support Vector Machine (SVM)**

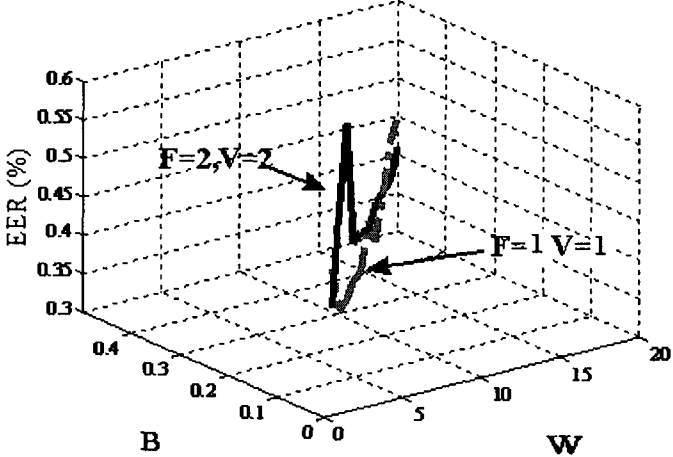
We conducted experimentation on the dataset of Chinese Academy of Sciences - Institute of Automation (CASIA), namely CASIA 1 and CASIA 2 [135]. CASIA 1 iris image database contains 756 grey scale eye images from 108 eyes (hence 108 different classes, 1 class each from 108 different users). Each iris image class consists of 7 samples taken in two sessions, three samples in the first session and four in the second session with an interval of one month between the two sessions, which is a real world application level simulation and allows a realistic test of our application. Each of the iris images are 320 x 280 pixels gray scale that has been captured by a digital optical sensor designed by National Laboratory of Pattern Recognition (NLPR). We also perform our experimentation on CASIA 2 iris image dataset, which comprises of 1200 grey scale iris images corresponding to 60 different individuals. Each of these classes or unique eye consists of 20 different samples. We conducted the experiments in two phases: in the first phase, we evaluated the performance of our system using support vector machines (SVMs). We also made a comparative analysis of our method with some existing methods with respect to recognition accuracy. In the second phase, we performed experimentation using Hausdorff distance in order to evaluate the similarity between two images and determine the recognition accuracy. We first show the accuracy of the system using a three dimensional representation of the log-Gabor values and then verify the results using a Receiver Operator Characteristic curve (ROC). We vary the value of the threshold in order to obtain the corresponding

values of False Accept Rate (FAR) and False Reject Rate (FRR) in order to replicate the different security tolerance levels demanded by various situations.



**Figure 7.7:** Selection of optimal values of 1D log-Gabor parameters in CASIA 1 image dataset

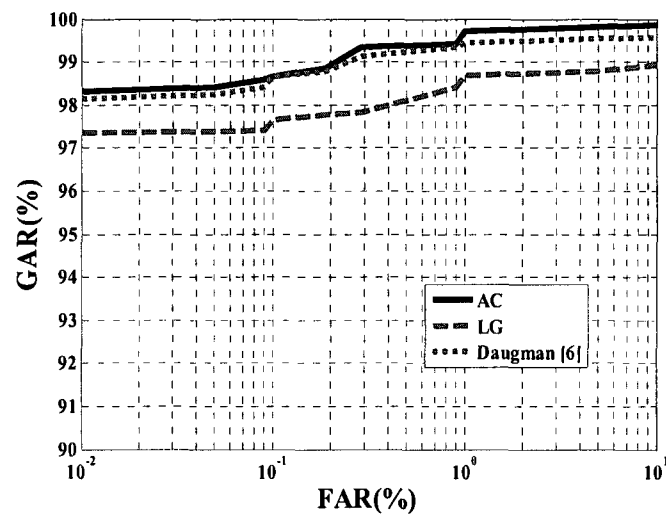
Leave-One-Out cross validation is used for CASIA 1 dataset, and for CASIA 2 dataset, we use 3-fold cross-validation to obtain the training accuracy for AASVM.



**Figure 7.8:** Selection of optimal values of 1D log-Gabor parameters in CASIA 2 image dataset

We need to tune a number of parameters required to process feature extraction using 1D log-Gabor filters, in order to assure a higher matching accuracy. These parameters include number of filters  $F$ , base wavelength  $W$ , filter bandwidth  $B$  and multiplicative factor between center wavelengths of successive filters  $V$ . In order to avoid the redundancy of the filters, the output of each filter should be independent so that there is no correlation in the encoded iris pattern. To achieve maximum independence, the bandwidths of each filter must not overlap in the frequency domain, and also the center frequency should be spread out [16].

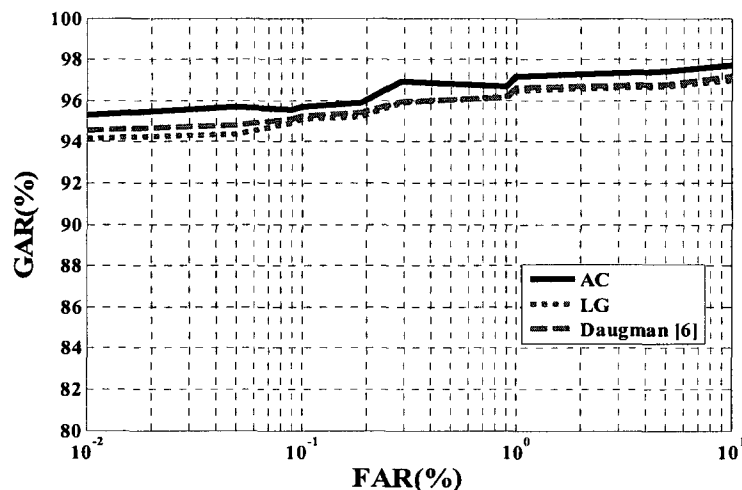
The template size with radial resolution of 20 pixels and angular resolution of 240 pixels is selected, and an iris template with 9600 bits of information has been



**Figure 7.9:** ROC curve shows the comparison between GAR and FAR on CASIA 1 iris image database

generated using these parameter values. We conduct several experiments on both the datasets, and the optimal parameters of the 1D log-Gabor filters are tuned to attain a reasonable accuracy (See Figure 7.7 and 7.8). From Figure 7.7, we can find

that lowest Equal Error Rate (EER) of 0.12% is obtained when the bandwidth  $B$  is 0.45 and the center wavelength  $W$  is 18 pixels on the CASIA 1 dataset. Figure 7.8 exhibits that the lowest EER of 0.41% is achieved when the bandwidth  $B$  is 0.50 and the center wavelength  $W$  is 16 pixels on the CASIA 2 dataset. For both of the datasets, we attain the higher accuracies when the number of filter,  $F$  and the multiplicative factor,  $V$  are set to 1. The performance of a verification system is evaluated using ROC curve; see Figure 7.10 and 7.11, which demonstrates how the *Genuine Acceptance Rate (GAR)* ( $1 - \text{FRR}$ ) changes with the variation in *False Accept Rate (FAR)* for the proposed approach, and a comparison is given between the randomized template of Authentication Code (AC) and the original template of



**Figure 7.10:** ROC curve shows the comparison between GAR and FAR on CASIA 2 iris image database

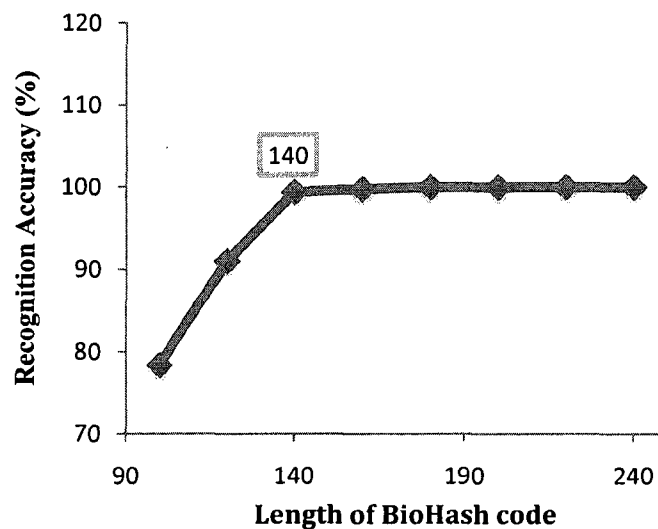
extracted features using 1D log-Gabor (LG) filters. We also compare our scheme with the Daugman's iris code [56]. It is observable from this figure that the proposed approach with AC performs better than the approach with LG on both of the

datasets. Also, the proposed iris recognition scheme with AC shows better accuracy than the Daugman's method; however, the proposed approach with LG provides lower accuracy than the Daugman's method for both of the datasets.

### 7.3.2. Hausdorff Distance

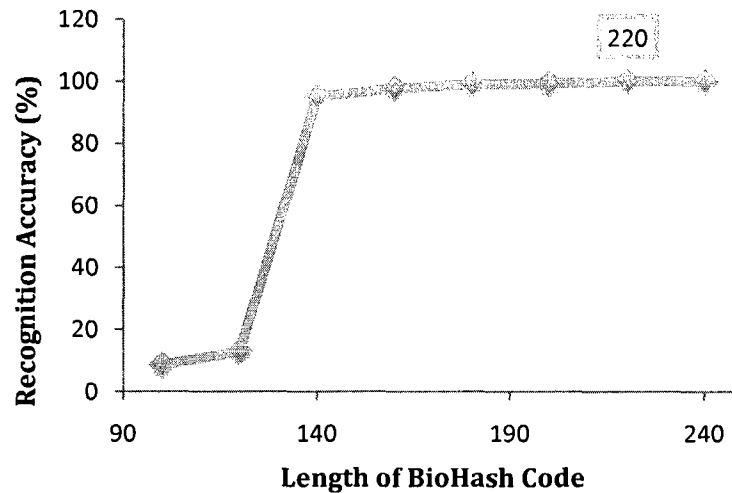
#### a. Cases 1, 2 and 4: (as discussed in Section 4.4.2 in Chapter 4)

The main objective while implementing the iris recognition system along with BioHashing is to be able to maximize the inter class variation between two different classes and simultaneously minimize the intra-class variation for attaining higher accuracy during user classification. In the second phase of performance evaluation, user classification is performed using Hausdorff distance for the user authentication module after which, only valid users are permitted access to their corresponding home network to control a particular networked appliance.



**Figure 7.11:** Selection of optimal length of BioHash code in CASIA 1 iris image dataset

We performed numerous tests on CASIA 1 and CASIA 2 datasets to first determine the variation in recognition accuracy when the length of the BioHash code is varied (as seen in Figure 7.11 and Figure 7.12). In Figure 7.11 we observe



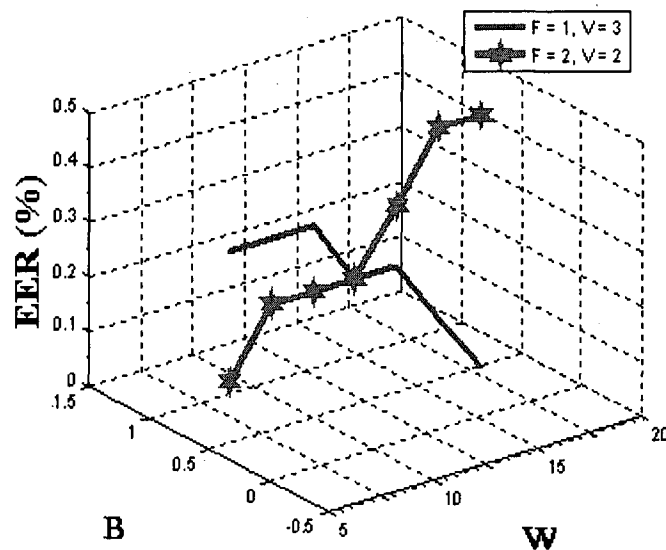
**Figure 7.12:** Selection of optimal length of BioHash code in CASIA 2 image dataset

that the recognition accuracy reaches 100% when the BioHash code length is just 140 in CASIA 1 dataset. From Figure 7.12, we can conclude that in CASIA 2 dataset, the optimum BioHash code length is 220 at which, recognition accuracy attains 100% resembling a perfect recognition system. We perform the rest of the experiments with BioHash code length of 140 and 220 for CASIA 1 and CASIA 2 respectively.

A biometric system usually has two possible outcomes, called error rates: FAR (False Accept Rate) and FRR (False Reject Rate). The interdependency between these two rates is regulated by another important parameter, the threshold value  $\mu$ .



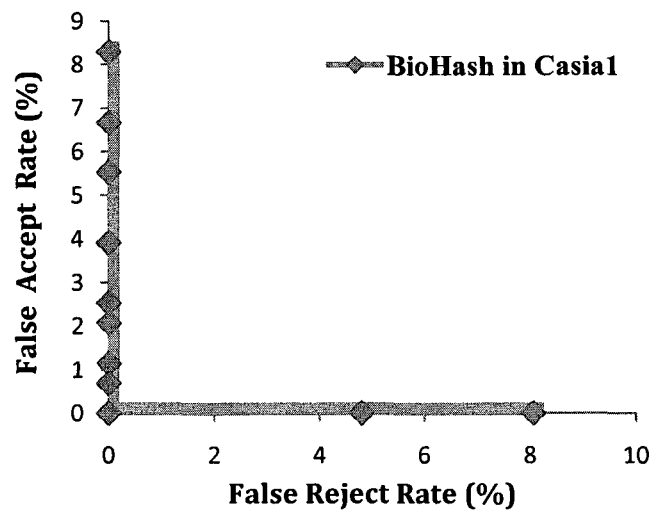
As  $\mu$  increases, the value of FAR decreases and that of FRR increases. In the verification mode also, the performance of our iris recognition system is described by the Receiver Operating Characteristic (ROC) curve and Equal Error Rate (EER). The ROC curve is developed by comparing FAR and FRR, while varying the normalized threshold value  $\mu$  in the range of 8-11. The large value of  $\mu$  can be contributed to the pseudo-random numbers that raises the base threshold value to a higher numerical space on being multiplied with  $\mu$ . EER is the point where FAR and FRR assumes equal value, calculated by  $(FAR+FRR)/2$ . The smaller the value of EER, the better is the performance of the system. In CASIA 1, for FRR test, all the seven images of each eye are matched against each other avoiding symmetric matches, thus generating  $(108 \times 7 \times 6)/2 = 2268$  genuine match scores. Additionally, for the FAR test, the first image of the eye of each user is matched with the first image of all



**Figure 7.13:** Selection of optimal values of 1D log-Gabor parameters in CASIA 1 image dataset

the other users avoiding symmetric matches, thus generating  $(108 \times 107 \times 7)/2 = 40,446$  imposter matching scores.

Similar to the first phase, we adjust few parameters in order to process feature extraction using 1D log-Gabor filters to assure a higher matching accuracy. The template size with radial resolution of 20 pixels and angular resolution of 240



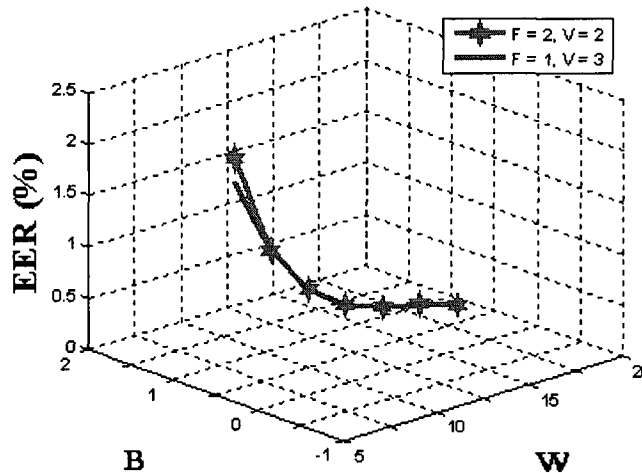
**Figure 7.14:** ROC curve for BioHash on CASIA 1 image dataset

pixels is selected for both CASIA 1 and 2 datasets, and an iris template with 9600 bits of information is generated using these parameter values.

After conducting numerous experimentations on the CASIA 1 dataset, and the optimal parameters of the 1D log-Gabor filters tuned to attain a reasonable accuracy (see Figure 7.13). From Figure 7.13 we observe that the lowest equal error rate (EER) of 0.0% is obtained when the bandwidth  $B$  takes the value 0.5, the centre wavelength  $W$  is 18 pixels, the number of filters  $F$  is 1 and  $V$  is 3. In the same database EER attains the lowest value of 0.11% when  $B$  is 0.5,  $W$  is 18 pixels and

both  $F = V = 2$ . From the ROC curve in Figure 7.14, we can verify that the best result is obtained, where Equal Error Rate (EER) attains the lowest value of 0.0% when the bandwidth  $B$  is 0.5, center wavelength  $W$  is 18 pixels, and when the threshold  $\mu$  is equal to 10. As the definition of EER goes, at this point, the value of both FAR and FRR attains 0%.

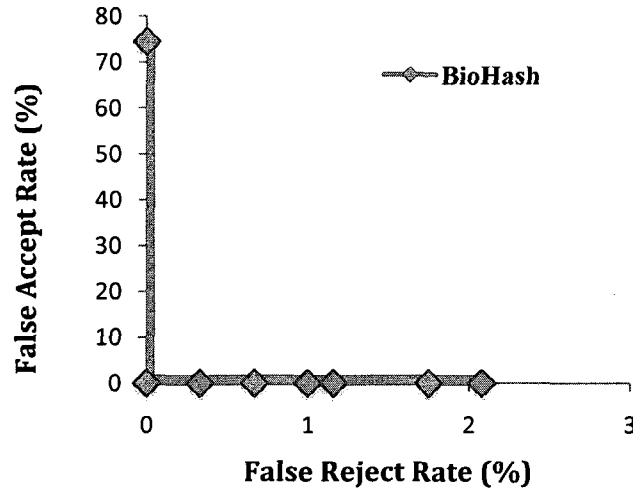
Similar to CASIA 1 experiments, in CASIA 2 dataset we first calculate the FRR by comparing all 20 images of a single eye against each other avoiding symmetric match in order to determining all possible combinations of genuine attempts, thus generating  $(60 \times 20 \times 19)/2 = 11,400$  genuine match scores. For the FAR test, the first image of each iris is matched against the first image of all other irises and the



**Figure 7.15:** Selection of optimal values of 1D log-Gabor parameters in CASIA 2 image dataset

same process is repeated for subsequent images, resulting to  $(60 \times 59 \times 20)/2 = 35,400$  imposter attempts.

We alter few parameters of the 1D log-Gabor filters in order to attain the best matching accuracy (see Figure 7.15) in CASIA 2 database. From Figure 7.15, we

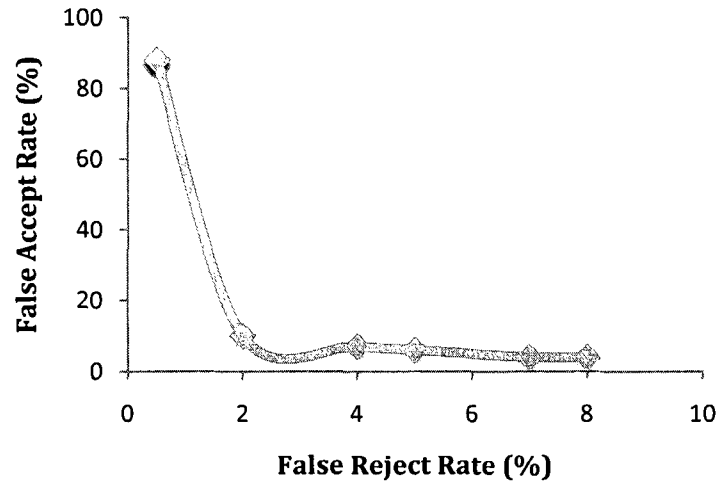


**Figure 7.16:** ROC curve for BioHash on CASIA 2 image dataset

notice that EER reaches 0.0% when the bandwidth  $B$  takes the value 0.5, the centre wavelength  $W$  is 18 pixels and the number of filters  $F = V = 2$ . We can also observe that for  $F = V = 2$  and  $F = 1, V = 3$ , both the curves take very similar shape, which shows that the system behaves almost the same for the above values of  $F$  and  $V$ . Observing the ROC curve in Figure 7.16 we observe that the lowest equal error rate (EER) of 0.0% is obtained when the bandwidth  $B$  takes the value 0.5, the centre wavelength  $W$  is 18 pixels, and the threshold  $\mu$  is equal to 10.

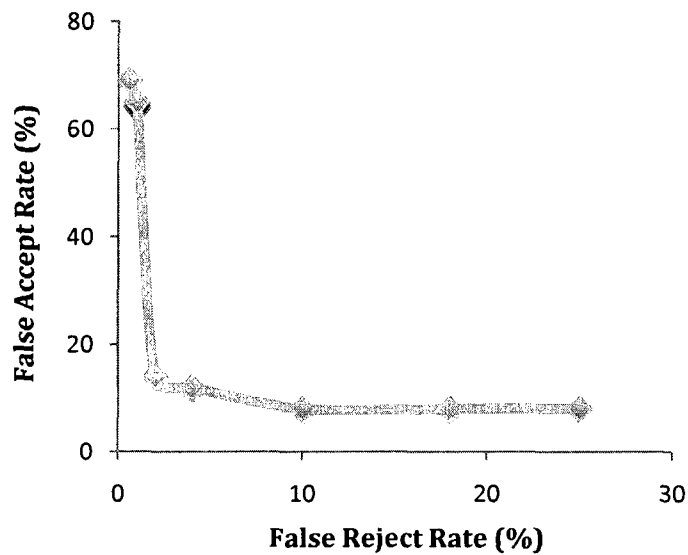
***b. Case 3: (as discussed in Section 4.4.2 in Chapter 4)***

The ROC curves in Figure 7.17 and Figure 7.18 are constructed using a single array of random number for all comparisons. The number of imposter attempts in both the databases remains the same. In this case, we assume that the biometric



**Figure 7.17:** ROC curve for BioHash on CASIA 1 image dataset

data of all the imposters are treated with a genuine random number; this corresponds to the case when an imposter submits his/her biometric data along with a valid user ID. The server on receiving a valid user ID verified it to be a valid one and sent a genuine random number to the client terminal. But an imposter is



**Figure 7.18:** ROC curve for BioHash on CASIA 2 image dataset

present at the client terminal who submits his/her iris image, which is an unregistered image, and this gets hashed with a genuine random number. This corresponds to Case 3 in *Section 4.4.2* in Chapter 4. Generating the ROC curve from the values of FAR and FRR obtained, we find that the lowest value of EER attained is higher in both CASIA 1 and CASIA 2 databases as compared to 0% in all the previous 3 cases. In this case, the lowest value of EER is 6% in CASIA 1 database and 7.5% in CASIA 2 database, which shows that the accuracy of the system in this case is less than when using different random numbers for different users. But since, there is no direct link between a user's user ID and the random number sent to him, if an imposter somehow gets a valid user ID, he will be able to access our proposed system with a recognition accuracy of about 93% (average of both CASIA 1 and CASIA 2 database results).

Though this reduced recognition accuracy of 93% (average) is permissible in home access scenarios, additional measure needs to be taken if we want maximum security in sensitive applications like accessing the nuclear reactors or missiles, etc. In these situations, we suggest using a simple user-known metric to be submitted by the user during submission of user id. The user can be asked to submit his/her user id and a simple value like the number of smart appliances connected to the network (home or any private). This question is not very difficult to remember and at the same time is confidential enough to be accurately known by the user himself. Hence this way, we can increase the overall security of the application even if the false acceptance rate is high when a valid user id is used with an invalid iris image.

## 7.4. Security Analysis

Here we discuss the strength of security in BioHash mechanism implemented in our application. Following the design of our system, biometric data of the user is sent from the user terminal to the server after hashing it using a key that is delivered to the user terminal by the server, which stores it in the server database (a separate unique key corresponding to each user). We need to verify that if the hashed biometric data (BioHash) gets intercepted or recorded by an adversary, it is computationally infeasible for the person to be able to recover the original feature data, even if he/she knows the random number being used by the genuine user. This mechanism reinstates the fact that only true combination of the biometric feature data and the random token can contribute to positive authentication. The previously described Equation (5) in Chapter 4 *Section 4.4.1.D* can be represented [48] in the matrix format as

$$s = \text{Sig}(\Sigma\beta - \mu) \quad (6)$$

where  $\text{Sig}(\cdot)$  is defined as the signum function,  $\mu$  is the preset threshold value of the system and  $\beta$  is the inner product between the feature vector and pseudo-random number called "BioHash code". Alternatively, we can also represent it as

$$s = \text{Sig}(V\Gamma - \mu) \quad (7)$$

where  $\Gamma$  is the feature vector ( $\Gamma \in \mathbb{R}^n$ , where  $n$  denotes the length of the feature),  $V$  is the  $l \times n$  orthonormal matrix of random numbers ( $V \in \mathbb{R}^{l \times n} \mid i = 1, \dots, l$ ) where  $l < n$ , and  $V\Gamma$  is the random projection [33].

From Equation (7), we get  $v = V\Gamma$  such that, with this value we can successfully distinguish a genuine user from a potential imposter [136]. Vector  $v$  can be regarded as the set of undetermined systems of linear equations, which has more unknowns than equations. Hence, it is impossible to determine the exact values of all the elements present in  $\Gamma$  just by solving this undetermined linear equation system in  $v = V\Gamma$  if  $l < n$ , which is firmly founded on the hypothesis that infinite number of solutions are possible in this framework. Additionally, from the exhaustive formal proofs discussed in [48], [136], we can infer that even if the pseudo-random token  $V$  is known to the adversary, it is unfeasible for him/her to calculate the exact values of the all the elements in the feature vector  $\Gamma$  for a user. Thus, we prove that no one can derive the actual biometric information of a user even if he/she has complete knowledge of the random numbers used to generate the BioHash code.

## 7.5. Conclusions

In this thesis, we present a novel architecture to implement secure remote access to home appliances over the Internet using SSL, iris recognition and BioHashing via random numbers. The proposed approach consists of four domains, from which we implemented and tested the first two domains while exhaustively researched on the later two areas and proposed the most suitable methods to use and a scheme to implement them.



We divided the complete architecture into four component areas: Secure Sockets Layer communication channel, BioHashing, Authentication Server and Home Network. We developed a secure client and server communication channel that uses SSL v3.0 using OpenSSL v-0.9.8g that is secure towards most of the attacks on digital data. We also developed a novel scheme to perform BioHashing of iris image data such that, users do not have to remember or carry a long password, but still be able to transfer their biometric data to the server for authentication in a very secured way with 100% and 93% (average) recognition accuracy (in two separate situations). When a user registers with the server for using this application, the server generates a random but unique key for each user and stores it in its database. Later, when the user tries to use this application on the Internet to access his/her home appliances, he needs to perform a verification procedure in order to rightly identify himself and access the right home. During this verification procedure, the user submits only his user ID to the application, which is first verified by the server to check whether this user is a registered user or not. On successful verification of the user ID, the server sends the stored random number to the client terminal after further randomizing it with a small random number generated at that time. The user terminal now sends the user's iris image to the server after performing BioHash on it (using the random number that the server sent). The server now performs a comparison between the BioHash of the user submitted image and the same on the stored image of the same user. Depending on the outcome, the user is considered as a valid user or an invalid one. If the user is a genuine user, he/she is allowed access to the corresponding home network. This is done using Session Initiation Protocol

(SIP) which determines the IP address of the home gateway and passes this to the client connection. The client connection now connects to the home gateway directly using SIP. We discussed in brief the reasons for selecting SIP among other similar protocols and how it fits in our architecture. On the other hand, the home gateway maintains all the information of the networked appliances inside the home (which are preferably networked using Universal Plug and Play home network). So when it receives command messages from a user for a particular home appliance, it redirects it to the target appliance and returns back the status on the outcome of the command. In this phase, we briefly discussed the various home networking technologies that are applicable in our architecture and why UPnP is the preferred technology to use.

We evaluate our system based on the accuracy of recognition of BioHashing performed on two databases, CASIA 1 and CASIA 2 in two phases; first using different random numbers for different users, later using the same random number for the different users. Also, we first use SVM in order to evaluate the recognition accuracy in our system and compare it with two other systems, and then use Hausdorff distance to find the extent of similarity between two user iris images.

## **7.6. Contributions to the Knowledge**

In this thesis, we made a number of contributions to develop a new architecture to implement a three-layered secured remote access to home appliances via Internet. Firstly, we designed a novel architecture for remote access to home appliances providing three layers of security to the user and application

data in the system. Previous designs either focused on the home server or on the home network and the interaction between the different networked appliances. But until now, there is no architecture for a secured remote access, and which discusses the complete design of the endeavor. Our design provides three tiers of security to the whole application and is very secure against the most common attacks on web applications like sniffing attacks, bucket brigade attack, replay attack, dictionary attack, eavesdropping, tampering, connection hijacking, etc. Our proposed scheme not only provides very high security but also considers user's convenience, effort and cost issues while implementing the architecture. We also developed a new design for the home server that will be able to cater to the needs of multiple homes and handle all the users for accessing their home appliances over the Internet. Using our architecture, multiple apartments will be able to provide access their home appliances (networked inside the home to form a home network) in a trouble-free manner, without requiring any specialized knowledge for using this system. This will not only help in managing the user records, but will also decrease the expense of each user and the time and energy spent by them to install and maintain such a server. This will in a way help in the commercial viability of such a system as it will be able to satisfy user needs in a simple and cost effective manner. As a third contribution, we developed this system architecture without inventing any new protocol or language, but using already available protocols like SSL, SIP and UPnP to build this design. All these components are backward compatible and is widely accessible without any problem. The performance of our design exhibits encouraging results, where we attained 100% and 93% accuracy in two different

scenarios during iris recognition after BioHashing. This proves that this scheme can be used successfully for developing a very secure method to access one's home appliances in a cost effective and simple way.

## **7.7. Future Research**

In this thesis, our proposed architecture performs comparatively well for most of the situations except one, where it fails to be 100% accurate in determining the authenticity of a user. In order to increase the accuracy of this procedure, better feature extraction algorithms can be used or experimented with. Moreover, as a future work, a complete system can be developed following this architecture in order to prove all the claims made in this research. In addition to this, since this proposed application is not a monitored system and users will access this application at any time of the day, they would perform unsupervised biometric authentication. As a result, stricter rules need to be set as to how many attempts should a user be allowed in order to prove his/her identity. In this regard, the level of security desired needs to be determined first while formulating such rules. Moreover, such situations demands liveness detection so as to prevent attacks by an adversary entering a stored iris image (in case of using JPEG iris image as input to the application) or a printed copy of an iris image (in case of directly taking the picture of the eye) of a user to gain entry to the application and control home appliances.

Another important section where future work needs to be done is in testing a real life implementation of the recognition accuracy after BioHashing on images

taken with a laptop's or computer's camera. Since no iris image database is available that uses such cameras, we based our experiments on the popularly available CASIA databases which has been developed with cameras that have better resolution than present day laptop cameras. As a result, modifications in the feature extraction algorithm may be required to actually make it fit for use with less accurate laptop cameras. Lastly, the computations like normalization of user submitted iris image and its corresponding feature extraction and hashing presently requires MATLAB software for operating in the client terminal. Further work needs to be done to convert these MATLAB specific files to C dlls, so that, the client side computations can be done without requiring this software.

## References

- [1] D. Cook, and S. Das, *Smart environments: Technology, protocols and applications*. Wiley-Interscience, 2004.
- [2] J. Lee, Y. Kim, and K. Kim, "Development of internet home server to control information appliances remotely," *Int. Conf. Information Networking, Wireless Comm. Tech. and Network Appl.*, Springer Lecture Note Series in Computer Science (LNCS), 2343, pp. 581-588, 2002.
- [3] C. S. Bae, J. H. Yoo, K. C. Kang, Y. S. Choe, and J. W. Lee, "Home server for home digital service environments," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 1129-1135, Nov.2003.
- [4] G. O'Driscoll, *The Essential Guide to Home Networking*. Prentice Hall PTR Upper Saddle River, NJ, USA, 2000.
- [5] I. Chong, *Wired Communications and Management*. Springer, 2002.
- [6] Y. G. Jang, H. I. Choi, and C. K. Park, "Implementation of Home Network Security System based on Remote Management Server," *IJCSNS*, vol. 7, no. 2, pp. 267, 2007.
- [7] A. Leventis, T. Antonakopoulos, C. Stavroulopoulos, T. Luckenbach, and V. Makios, "Intelligent devices for appliances control in home networks," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 2, pp. 328-336, 2003.
- [8] T. B. Zahariadis, *Home networking technologies and standards*. Artech House, 2003.
- [9] C. Bae, J. Lee, and C. Kim, "State of the art and the development direction of home server technology," *Korea Information Processing Society Review*, vol. 8, no. 1, pp. 28-41, 2001.
- [10] M. Nikolova, F. Meijs, and P. Voorwinden, "Remote mobile control of home appliances," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 1, pp. 123-127, 2003.

- [11] E. A. Young, and T. J. Hudson, "OpenSSL," *World Wide Web*, <http://www.openssl.org/>, vol. 9 2001.
- [12] J. Viega, M. Messier, and P. Chandra, *Network security with OpenSSL*. O'Reilly Media, Inc., 2002.
- [13] R. Oppliger, R. Hauser, and D. Basin, "SSL/TLS session-aware user authentication: Or how to effectively thwart the man-in-the-middle," *Computer Communications*, vol. 29, no. 12, pp. 2238-2246, 2006.
- [14] R. Oppliger, R. Hauser, and D. Basin, "SSL/TLS session-aware user authentication," *Computer*, vol. 41, no. 3, pp. 59-65, 2008.
- [15] A. K. Jain, R. Bolle, and S. Pankanti, *Biometrics: personal identification in networked society*. Kluwer Academic Publishers, 1999.
- [16] L. Ma, T. Tan, Y. Wang, and D. Zhang, "Personal identification based on iris texture analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1519-1533, December 2003.
- [17] L. Ma, T. Tan, Y. Wang, and D. Zhang, "Efficient iris recognition by characterizing key local variations," *IEEE Transactions on Image Processing*, vol. 13, no. 6, pp. 739-750, 2004.
- [18] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4-20, 2004.
- [19] G. O. Williams, "Iris recognition technology," *IEEE Aerospace and Electronic Systems Magazine*, vol. 12, no. 4, pp. 23-29, 1997.
- [20] N. Poh, and S. Bengio, "Compensating User-Specific Information with User-Independent Information in Biometric Authentication Tasks," *Research Report*, pp. 5-44.
- [21] R. P. Wildes, "Iris recognition: an emerging biometric technology," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1348-1363, 1997.
- [22] J. Daugman, "How iris recognition works," *Handbook of Image and Video Processing*, pp. 1251, 2005.
- [23] J. M. H. Ali, and A. E. Hassanien, "An iris recognition system to enhance e-security environment based on wavelet theory," *AMO-Advanced Modeling and Optimization*, vol. 5, no. 2, pp. 93-104, 2003.
- [24] S. Prabhakar, S. Pankanti, and A. K. Jain, "Biometric recognition: Security and privacy concerns," *IEEE Security & Privacy*, vol. 1, no. 2, pp. 33-42, 2003.

- [25] Y. Sutcu, H. T. Sencar, and N. Memon, "A secure biometric authentication scheme based on robust hashing," *Proceedings of the 7th Workshop on Multimedia and Security*, ACM New York, NY, USA, pp. 111-116, 2005.
- [26] R. Venkatesan, S. M. Koon, M. H. Jakubowski, and P. Moulin, "Robust image hashing," *Proceedings of the IEEE International Conference on Image Processing (ICIPÆ00)*, Citeseer, 2000.
- [27] A. B. J. Teoh, and T. S. Ong, "Secure biometric template protection via randomized dynamic quantization transformation," *International Symposium on Biometrics and Security Technologies (ISBAST 2008)*, pp. 1-6, 2008.
- [28] V. Monga, and B. L. Evans, "Perceptual image hashing via feature points: performance evaluation and tradeoffs," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3452, 2006.
- [29] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. CRC press, 1997.
- [30] V. Monga, A. Banerjee, and B. L. Evans, "Clustering algorithms for perceptual image hashing," *Proc. IEEE Digital Sig. Processing Workshop*, pp. 283-287, 2004.
- [31] J. A. Trinanes, "Database security in high risk environments," Technical report, governmentsecurity.org, 2005.
- [32] J. Turnbull, and S. Garrett, *Broadband applications and the digital home*. IEE, London, 2003.
- [33] H. Sinnreich, and A. B. Johnston, *Internet communications using SIP: delivering VoIP and multimedia services with Session Initiation Protocol*. Wiley, 2006.
- [34] A. B. Johnston, *SIP: understanding the session initiation protocol*. Artech House, 2004.
- [35] D. Sisalem, J. Kuthan, and G. M. D. Fokus, "Understanding SIP," *Mobile Integrated Services, GMD Fokus*.
- [36] D. Goldschlag, M. Reed, and P. Syverson, "Onion routing," *Communications of the ACM*, vol. 42, no. 2, pp. 39-41, 1999.
- [37] E. Rescorla, and A. Schiffman, "The secure hypertext transfer protocol," RFC 2660, August 1999.
- [38] A. Lingnau, O. Drobnik, and P. Domel, "An HTTP-based infrastructure for mobile agents," *Fourth International World Wide Web Conference Proceedings*, Citeseer, pp. 461-471, 1995.



- [39] R. Cohen, "On the establishment of an access VPN in broadband access networks," *IEEE Communications Magazine*, vol. 41, no. 2, pp. 156-163, 2003.
- [40] J. Tyson, "How virtual private networks work," *How Stuff Works*, 2002.
- [41] J. De Clercq, and O. Paridaens, "Scalability implications of virtual private networks," *IEEE Communications Magazine*, vol. 40, no. 5, pp. 151-157, 2002.
- [42] C. Scott, P. Wolfe, and M. Erwin, *Virtual private networks*. O'Reilly Media, Inc., 1999.
- [43] M. Van der Haak, A. C. Wolff, R. Brandner, P. Drings, M. Wannemacher, and T. Wetter, "Data security and protection in cross-institutional electronic patient records," *International journal of medical informatics*, vol. 70, no. 2-3, pp. 117-130, 2003.
- [44] T. Markham, and C. Payne, "Security at the network edge: A distributed firewall architecture," *Proceedings of the 2nd DARPA Information Survivability Conference and Exposition*, vol. 1, pp. 279-286, 2001.
- [45] G. Kambourakis, A. Rouskas, and S. Gritzalis, "Using SSL/TLS in authentication and key agreement procedures of future mobile networks," *4th International Workshop on Mobile and Wireless Communications Network*, pp. 152-156, 2002.
- [46] A. Herzberg, "Why Johnny can't surf (safely)? Attacks and defenses for web users," *Computers & Security*, 2008.
- [47] M. Rahman, and P. Bhattacharya, "Remote access and networked appliance control using biometrics features," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 2, pp. 348-353, May 2003.
- [48] A. B. J. Teoh, Y. W. Kuan, and S. Lee, "Cancellable biometrics and annotations on BioHash," *Pattern Recognition*, vol. 41, no. 6, pp. 2034-2044, 2008.
- [49] J. Kim, S. Cho, D. Kim, and S. Chung, "Iris Recognition Using a Low Level of Details," *Lecture Notes in Computer Science*, vol. 4292, pp. 196, 2006.
- [50] T. Mansfield, G. Kelly, D. Chandler, and J. Kane, "Biometric product testing," *Final Report*, 2001.
- [51] L. Flom, and A. Safir, "Iris recognition system," U.S. Patent No. 4641394, Feb.3, 1987.
- [52] A. J. Mansfield, and J. L. Wayman, "Best practices in testing and reporting performance of biometric devices," Centre for Mathematics and Scientific Computing, National Physical Laboratory, 2002.

- [53] F. H. Adler, W. F. Norris, and G. E. de Schweinitz, "Physiology of the Eye. Clinical Application," *Southern Medical Journal*, vol. 44, no. 7, pp. 669, 1951.
- [54] H. Davson, *The eye* Academic Press, 1969.
- [55] D. Zhang, *Automated biometrics: Technologies and systems*. Kluwer Academic Publishers, 2000.
- [56] J. G. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1148-1161, 1993.
- [57] R. Sanchez-Reillo, and C. Sanchez-Avila, "Iris recognition with low template size," *Lecture Notes in Computer Science*, pp. 324-329, 2001.
- [58] W. W. Boles, and B. Boashash, "A human identification technique using images of the iris and wavelet transform," *IEEE transactions on signal processing*, vol. 46, no. 4, pp. 1185-1188, 1998.
- [59] C. Sanchez-Avila, R. Sanchez-Reillo, and D. de Martin-Roche, "Iris-based biometric recognition using dyadic wavelet transform," *IEEE Aerospace and Electronic Systems Magazine*, vol. 17, no. 10, pp. 3-6, 2002.
- [60] L. Ma, Y. Wang, and T. Tan, "Iris recognition based on multichannel Gabor filtering," *Proceedings of ACCV*, vol. 1, pp. 279-283, 2002.
- [61] L. Ma, Y. Wang, and T. Tan, "Iris recognition using circular symmetric filters," *Proceedings of the 16th International Conference on Pattern Recognition*, Los Alamitos, CA, USA: IEEE Computer Society, vol. 2, pp. 414-417, 2002.
- [62] S. Lim, K. Lee, O. Byeon, and T. Kim, "Efficient iris recognition through improvement of feature vector and classifier," *ETRI journal*, vol. 23, no. 2, pp. 61-70, 2001.
- [63] C. Tisse, L. Martin, L. Torres, and M. Robert, "Person identification technique using human iris recognition," *Proc. Vision Interface*, pp. 294-299, 2002.
- [64] C. Park, J. Lee, M. Smith, and K. Park, "Iris-based personal authentication using a normalized directional energy feature," *Proc. 4th Int. Conf. Audio and Video-Based Biometric Person Authentication*, Springer, pp. 224-232, 2003.
- [65] B. Kumar, C. Xie, and J. Thornton, "Iris verification using correlation filters," *Lecture Notes in Computer Science*, pp. 697-705, 2003.
- [66] K. Bae, S. Noh, and J. Kim, "Iris feature extraction using independent component analysis," *Lecture Notes in Computer Science*, pp. 838-844, 2003.

- [67] J. Daugman, "Statistical richness of visual phase information: update on recognizing persons by iris patterns," *International Journal of Computer Vision*, vol. 45, no. 1, pp. 25-38, 2001.
- [68] J. Daugman, "Demodulation by complex-valued wavelets for stochastic pattern recognition," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 1, no. 1, pp. 1-17, 2003.
- [69] R. P. Wildes, J. C. Asmuth, G. L. Green, S. C. Hsu, R. J. Kolczynski, J. R. Matey, and S. E. McBride, "A machine-vision system for iris recognition," *Machine Vision and Applications*, vol. 9, no. 1, pp. 1-8, 1996.
- [70] L. Ma, "Personal identification based on iris recognition," Ph.D dissertation, Inst. Automation, Chinese Academy of Sciences, Beijing, China, June 2003.
- [71] R. M. Bolle, J. H. Connell, and N. K. Ratha, "Biometric perils and patches," *Pattern Recognition*, vol. 35, no. 12, pp. 2727-2738, 2002.
- [72] F. Monroe, M. K. Reiter, Q. Li, and S. Wetzel, "Cryptographic key generation from voice," *2001 IEEE Symposium on Security and Privacy, 2001.S&P 2001.Proceedings*, pp. 202-213, 2001.
- [73] T. Connie, A. Teoh, M. Goh, and D. Ngo, "Palmhashing: a novel approach for cancelable biometrics," *Information Processing Letters*, vol. 93, no. 1, pp. 1-5, 2005.
- [74] G.I.Davida, Y.Frankel, B.J.Matt, and R.Peralta, "On the Relation of Error Correction and Cryptography to an Off Line Biometrics Based Identification Scheme," *Proc.Workshop Coding and Cryptography*, pp. 129-138, 1999.
- [75] A. Goh, D. C. and L. Ngo, "Computation of cryptographic keys from face biometrics," *Lecture Notes in Computer Science*, pp. 1-13, 2003.
- [76] F. Hao, R. Anderson, and J. Daugman, "Combining Crypto with Biometrics Effectively," *IEEE Transactions On Computers*, vol. 55, no. 9, pp. 1081, 2006.
- [77] A. T. B. Jin, and T. Connie, "Remarks on BioHashing based cancelable biometrics in verification system," *Neurocomputing*, vol. 69, no. 16-18, pp. 2461-2464, 2006.
- [78] A. Lumini, and L. Nanni, "An advanced multi-modal method for human authentication featuring biometrics data and tokenised random numbers," *Neurocomputing*, vol. 69, no. 13-15, pp. 1706-1710, 2006.
- [79] Y. H. Pang, A. B. J. Teoh, and D. C. L. Ngo, "Palmprint based cancelable biometric authentication system," *International Journal of Signal Processing*, vol. 1, no. 2, pp. 98-104, 2004.

- [80] A. Ross, and A. Jain, "Information fusion in biometrics," *Pattern Recognition Letters*, vol. 24, no. 13, pp. 2115-2125, 2003.
- [81] C. Soutar, D. Roberge, A. Stoianov, R. Gilroy, and B. V. Kumar, "Biometric encryption using image processing," *Proceedings of SPIE*, vol. 3314, pp. 178, 1998.
- [82] A. B. J. Teoh, and D. C. L. Ngo, "Cancellable biometrics featuring with tokenized random number," *Pattern Recognition Letters*, vol. 26, no. 10, pp. 1454-1460, 2009.
- [83] National Institutes of Science and Technology, "Secure Hash Standard," *NIST FIPS Publication 180-1*, vol. 17, 1995.
- [84] X. Wu, D. Zhang, and K. Wang, "A Palmprint Cryptosystem," *Lecture Notes in Computer Science*, vol. 4642, pp. 1035, 2007.
- [85] H. Feng, and C. C. Wah, "Private key generation from on-line handwritten signatures," *Information Management and Computer Security*, vol. 10, no. 4, pp. 159-164, 2002.
- [86] X. Li, X. Wu, N. Qi, and K. Wang, "A Novel Cryptographic Algorithm Based on Iris Feature," *International Conference on Computational Intelligence and Security (CIS'08)*, vol. 2, 2008.
- [87] U. Uludag, S. Pankanti, and A. K. Jain, "Fuzzy vault for fingerprints," *Lecture Notes in Computer Science*, vol. 3546, pp. 310, 2005.
- [88] S. Yang, and I. Verbauwhede, "Automatic secure fingerprint verification system based on fuzzy vault scheme," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05)*, vol. 5, 2005.
- [89] T. C. Clancy, N. Kiyavash, and D. J. Lin, "Secure smartcardbased fingerprint authentication," *Proceedings of the 2003 ACM SIGMM workshop on Biometrics methods and applications*, ACM New York, NY, USA, pp. 45-52, 2003.
- [90] G.I.Davida, Y.Frankel, and B.J.Matt, "On enabling secure applications through on-line biometric identification," *IEEE Symposium on Privacy and Security*, pp. 148-157, 1998.
- [91] A. Lumini, and L. Nanni, "An improved BioHashing for human authentication," *Pattern Recognition*, vol. 40, no. 3, pp. 1057-1065, 2007.
- [92] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM systems journal*, vol. 40, no. 3, pp. 614-634, 2001.

- [93] S. Lin, M. T. Ozsü, V. Oria, and R. Ng, "An extendible hash for multi-precision similarity querying of image databases," *Proceedings Of The International Conference On Very Large Data Bases*, Citeseer, pp. 221-230, 2001.
- [94] J. Fridrich, and M. Goljan, "Robust hash functions for digital watermarking," *Proceedings. International Conference on Information Technology: Coding and Computing*, pp. 178-183, 2000.
- [95] V. Monga, "Perceptually based methods for robust image hashing," 2005.
- [96] A. Swaminathan, Y. Mao, and M. Wu, "Image hashing resilient to geometric and filtering operations," *IEEE Workshop on Multimedia Signal Processing*, 2004.
- [97] A. Swaminathan, Y. Mao, and M. Wu, "Security of feature extraction in image hashing," *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP'05)*, vol. 2, 2005.
- [98] M. K. Mihcak, and R. Venkatesan, "New iterative geometric methods for robust perceptual image hashing," *Proc. of ACM Workshop on Security and Privacy in Digital Rights Management*, Springer.
- [99] M. Johnson, and K. Ramchandran, "Dither-based secure image hashing using distributed coding," *Proceedings of International Conference on Image Processing (ICIP'03)*, vol. 2, 2003.
- [100] J. Fridrich, "Image watermarking for tamper detection," *Proceedings of International Conference on Image Processing (ICIP'98)*, vol. 2, 1998.
- [101] S. Z. Wang, and X. P. Zhang, "Recent Development of Perceptual Image Hashing," *Journal of Shanghai University (English Edition)*, vol. 11, no. 4, pp. 323-331, 2007.
- [102] D. H. Hubel, and T. N. Wiesel, "Receptive fields and functional architecture in two nonstriate visual areas of the cat," *J. Neurophysiology*, pp. 229-289, 1965.
- [103] J. Dittmann, A. Steinmetz, and R. Steinmetz, "Content-based digital signature for motion pictures authentication and content-fragile watermarking," *IEEE International Conference on Multimedia Computing and Systems, 1999*, vol. 2, 1999.
- [104] S. Bhattacharjee, and M. Kutter, "Compression tolerant image authentication," *Proceedings of International Conference on Image Processing (ICIP'98)*, vol. 1, 1998.

- [105] A. T. B. Jin, D. N. C. Ling, and A. Goh, "Biohashing: two factor authentication featuring fingerprint data and tokenised random number," *Pattern Recognition*, vol. 37, no. 11, pp. 2245-2255, 2004.
- [106] C. Siew Chin, A. T. Beng Jin, and D. N. Chek Ling, "High security Iris verification system based on random secret integration," *Computer Vision and Image Understanding*, vol. 102, no. 2, pp. 169-177, 2006.
- [107] W. K. Kong, and D. Zhang, "Detecting eyelash and reflection for accurate iris segmentation," *International journal of pattern recognition and artificial intelligence*, vol. 17, no. 6, pp. 1025-1034, 2003.
- [108] J. Daugman, "The importance of being random: statistical principles of iris recognition," *Pattern Recognition*, vol. 36, no. 2, pp. 279-291, 2003.
- [109] K. Roy, and P. Bhattacharya, "Iris recognition based on collarette region and asymmetrical support vector machines," *Int. Conf. Image Anal. and Recog. (ICIAR)*, Springer Lecture Note Series in Computer Science (LNCS), vol. 4633, pp. 854-865, 2007.
- [110] A. Kong, K. H. Cheung, D. Zhang, M. Kamel, and J. You, "An analysis of BioHashing and its variants," *Pattern Recognition*, vol. 39, no. 7, pp. 1359-1368, 2006.
- [111] D. C. L. Ngo, A. B. J. Teoh, and A. Goh, "Eigenspace-based face hashing," *Lecture Notes in Computer Science*, pp. 195-199, 2004.
- [112] A. B. J. Teoh, D. C. L. Ngo, and A. Goh, "An integrated dual factor authenticator based on the face data and tokenised random number," *Lecture Notes in Computer Science*, pp. 117-123, 2004.
- [113] A. B. J. Teoh, D. C. L. Ngo, and A. Goh, "Personalised cryptographic key generation based on FaceHashing," *Computers & Security*, vol. 23, no. 7, pp. 606-614, 2004.
- [114] M. Araki, and A. Miyajima, "Application of Ubiquitous Web Technologies to Home Information Appliances," Citeseer.
- [115] S. Gupta, "Home Gateway," *White Paper of Wipro Technologies*, 2004.
- [116] Home Gateway—A Key to Open Digital Home, <http://www.huawei.com/news/view.do?id=89&cid=43>
- [117] J. Lee, and C. Bae, "Home server platform technology," *Korea Information Science Society Review*, 2001.

- [118] A. Mondal, K. Roy, and P. Bhattacharya, "Secure and Simplified Access to Home Appliance using Iris Recognition," *IEEE Workshop on Computational Intelligence in Biometrics: Theory, Algorithms, and Applications (CIB 2009)*, Nashville, USA, pp. 23-29, 2009.
- [119] A. Mondal, K. Roy, and P. Bhattacharya, "Secure Biometric System for Accessing Home Appliances via Internet," *4th IEEE International Conference on Internet Technology and Secured Transactions (ICITST-2009)*, London, UK, 2009.
- [120] B. A. Miller, T. Nixon, C. Tai, and M. D. Wood, "Home networking with universal plug and play," *IEEE Communications Magazine*, vol. 39, no. 12, pp. 104-109, 2001.
- [121] J. Rosenberg, "A presence event package for the session initiation protocol (SIP)," RFC 3856, August 2004.
- [122] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: session initiation protocol," RFC 3261, Internet Engineering Task Force, 2002.
- [123] B. Ramsdell, "S/MIME version 3 message specification," RFC 2633, June 1999, 1999.
- [124] T. Saito, I. Tomoda, Y. Takabatake, J. Ami, and K. Teramoto, "Home gateway architecture and its implementation," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 1161-1166, Nov. 2000.
- [125] A. Dutta-Roy, "Networks for homes," *IEEE spectrum*, vol. 36, no. 12, pp. 26-33, 1999.
- [126] R. Lea, S. Gibbs, A. ra-Abrams, and E. Eytchison, "Networking home entertainment devices with HAVi," *Computer*, vol. 33, no. 9, pp. 35-43, 2000.
- [127] D. Marples, and T. T. Inc, "The Open Services Gateway Initiati An Introductory Overview," *IEEE Communications Magazine*, pp. 111, 2001.
- [128] O. S. G. Alliance, "Open Services Gateway Initiative," *Specification download* [http://www.osgi.org/resources/spec\\_download.asp](http://www.osgi.org/resources/spec_download.asp), 2004.
- [129] S. H. Rhee, S. K. Yang, S. J. Park, J. H. Chun, and J. A. Park, "UPnP Home Networking-Based IEEE1394 Digital Home Appliances Control," *Lecture Notes in Computer Science*, pp. 457-466, 2004.
- [130] D. S. Kim, J. M. Lee, W. H. Kwon, and I. K. Yuh, "Design and implementation of home network systems using UPnP middleware for networked appliances," *IEEE Transactions on Consumer Electronics*, vol. 48, no. 4, pp. 963-972, 2002.

- [131] J. Waldo, "Virtual organizations, pervasive computing, and an infrastructure for networking at the edge," *Information Systems Frontiers*, vol. 4, no. 1, pp. 9-18, Apr.2002.
- [132] R. Gupta, S. Talwar, and D. P. Agrawal, "Jini home networking: a step toward pervasive computing," *Computer*, vol. 35, no. 8, pp. 34-40, 2002.
- [133] K. Edwards, "Core Jini Prentice Hall," *Upper Saddle River, NJ*, 1999.
- [134] Y. J. Oh, H. K. Lee, J. T. Kim, E. H. Paik, and K. R. Park, "Design of an extended architecture for sharing DLNA compliant home media from outside the home," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 2, pp. 542-547, May2007.
- [135] OpenSSL, <http://www.openssl.org/>
- [136] S. Kaski, "Dimensionality reduction by random mapping: Fast similarity computation for clustering," *Proceedings of IJCNN*, Citeseer, vol. 98, pp. 413-418, 1998.



## Appendix A

The following research papers originated from the research work described in this thesis. The first paper has already been presented in the corresponding international conference and the second paper accepted for presentation.

- [1] **A. Mondal**, K. Roy and Prabir Bhattacharya, "Secure and simplified access to home appliance using iris recognition," *IEEE Workshop on Computational Intelligence in Biometrics: Theory, Algorithms, and Applications (CIB 2009)*, Nashville, Tennessee, USA, 2009. **Presented (Oral)**.
- [2] **A. Mondal**, K. Roy and Prabir Bhattacharya, "Secure biometric system for accessing home appliances via Internet," *4th IEEE International Conference on Internet Technology and Secured Transactions (ICITST-2009)*, London, UK, November 9-12 2009. **Accepted for presentation**.