A NOVEL APPROACH FOR OPTIMIZATION OF WIRELESS SECURITY PROTOCOLS BASED ON MOBILE AGENT

THESIS

submitted in fulfillment of the requirement of the degree of

DOCTOR OF PHILOSOPHY

to

J.C. BOSE UNIVERSITY OF SCIENCE AND TECHNOLOGY, YMCA

by

UMESH KUMAR

Registration No: YMCAUST/Ph.D-13/2012

Under the Supervision of

Dr. SAPNA GAMBHIR

ASSISTANT PROFESSOR

Department of Computer Engineering
Faculty of Engineering & Technology
J.C. Bose University of Science and Technology, YMCA
Sector-6, Mathura Road, Faridabad, Haryana, INDIA
JULY 2020

DECLARATION

I hereby declare that this thesis entitled **"A NOVEL APPROACH FOR OPTIMIZATION OF WIRELESS SECURITY PROTOCOLS BASED ON MOBILE AGENT"** by **UMESH KUMAR,** being submitted in fulfillment of requirement for the award of Degree of Doctor of Philosophy in the Department of Computer Engineering under Faculty of Engineering & Technology of J.C. Bose University of Science and Technology YMCA, Faridabad, during the academic year March 2013 to September 2019, is a bonafide record of my original work carried out under the guidance and supervision of **DR. SAPNA GAMBHIR, ASSISTANT PROFESSOR, DEPARTMENT OF COMPUTER ENGINEERING, J.C. BOSE UNIVERSITY OF SCIENCE AND TECHNOLOGY, YMCA, FARIDABAD** and has not been presented elsewhere.

I further declare that the thesis does not contain any part of work which has been submitted for the award of any degree either in this University or in any other University.

(UMESH KUMAR)
Registration No. YMCAUST/Ph.D-13/2012

CERTIFICATE

This is to certify that this thesis entitled "**A NOVEL APPROACH FOR OPTIMIZATION OF WIRELESS SECURITY PROTOCOLS BASED ON MOBILE AGENT**" by Umesh Kumar submitted in fulfilment of the requirement for the award of Doctor of Philosophy in **DEPARTMENT OF COMPUTER ENGINEERING**, under Faculty of Engineering and Technology of **J.C. Bose University of Science & Technology, YMCA, Faridabad**, during the academic year 2019-2020, is a bonafide record of work carried out under my guidance and supervision.

I further declare that to the best of my knowledge, thesis does not contain any part of work which has been submitted for the award of any degree either in this University or in any other University.

Sapna

Dr. Sapna Gambhir
Assistant Professor
Department of Computer Engineering
Faculty of Informatics & Computing
JC BOSE UST, YMCA, Faridabad

The Ph.D viva-voce examination of Research Scholar Umesh Kumar (YMCAUST/Ph.D-13/2012) has been successfully held on 22nd July 2020.

Sapna

(Signature of Supervisor)

RKSingla 22/07/2020

(Signature of Chairman)

(Signature of External Examiner)

ACKNOWLEDGEMENT

It is a pleasure for me to thank all the people who have helped me to realize this Ph.D. thesis. First, I want to thank Dr. Sapna Gambhir for giving me the opportunity to start a Ph.D. and for providing her knowledge and research experience for completion of this work. I strongly appreciate her support, guidance and advice during the years of my Ph.D.

I am grateful to Professor Atul Mishra, Dean R&D for the guidance and support on technical implementation of this thesis. Special thanks to Professor Komal Kumar Bhatia (Dean and Chairman) for encouraging me at every stage to do my research.

I would like to thank Dr. Ashutosh Dixit, Dr. Naresh Chauhan, Dr. C.K. Nagpal, Dr. Manjeet Singh, Dr. Neelam Duhan for being my teacher at the masters level and showing me the path in this education world.

Big thanks go to my colleagues at the University for their friendliness and help. Dr. Harish Kumar, Dr. Vedpal, Dr. Sushil Panwar, Dr. Jyoti, Sh. Mukesh Gupta for providing me support and advice during the tough times.

I also want to acknowledge Late Sh. A.K. Sharma for generating a spark of research inside me.

I also want to thank TCS, my previous organization, for providing me the opportunity to work there and imparting me the best practical and theoretical training which greatly helped me in my research work.

Last but of course not least, I want to thank my parents, my sisters and my brother Ramesh for their support and encouragement. Very special thanks to my wife Diksha for facing and supporting me in tough times. Thanks to my motivation, my son master Vibhor to whom I am a hero in this world.

UMESH KUMAR

ABSTRACT

The exponential growth of users on the internet gave birth to tremendous traffic on the internet and various security issues. Wireless networks have brought a revolution in the area of networking. For these networks to be successful security is one of the important aspect. Wireless networks are more susceptible to attacks due to the broadcast nature of these networks.

A recent study has shown that authentication plays a vital role for the utilization of network resources by an authentic user. Currently existing authentication protocols have some vulnerabilities which leads to various attacks like man in the middle, black hole attack etc. Also due to the client server based architecture, traffic generated by these protocols is very large. This research proposes a new mobile agent based framework for wireless authentication. This framework can adopt different authentication protocols like MD5, TLS and PEAP etc. Proposed model makes use of the mobile agent concept which greatly helps in reducing the traffic around the authentication server.

Key Exchange is also one of the important aspect in the wireless security domain. Extensive study about the different key exchange methods has shown that there are some issues in current key exchange protocols in terms of single point of failure, higher traffic, cost, different certificate issuing authority etc. So there was a need of a robust key exchange protocol which apart from doing key exchange can do authentication of Key Distribution Center (KDC) and participating nodes also. A novel biometric based key distribution through fingerprint based authentication has been proposed. This model helps in authentication of KDC and nodes while doing key exchange. Model makes use of the mobile agent, which greatly helps in reducing the traffic at the KDC.

While exploring authentication protocols it was observed that signature of the device can be one of the mechanism to authenticate a user or device on the network. A new method for creation of device signature is presented. This method for creation of device signature takes multiple parameters which are extracted from the device. These parameters can be static or dynamic in nature. Using these parameters MD5 fingerprint of the device can be created and can be used later on for authentication.

Signature mechanism can have application in various domains. A new mobile agent based MapReduce framework is designed for straggler detection and relocation. In this technique signature of the device can be very useful in preventing the unauthorized access or attacks like Distributed Denial of Service (DDoS). Proposed mechanism is also compared with the existing state of the art technology i.e. Hadoop. Results are compared on different parameters and proposed mechanism provides good results in terms of straggler detection.

For simulation and result analysis we have used Network Simulator (NS2), Android application development and Java Agent Development Environment (JADE). These tools are used to demonstrate the virtues of the proposed mechanisms. The analysis of generated outcomes reveals that the proposed mechanisms are superior to existing methods.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

ABBREVIATIONS

MA	Mobile Agent
EAP	Extensible Authentication Protocol
MD5	Message Digest 5
WEP	Wired Equivalent Privacy
LEAP	Lightweight Extensible Authentication Protocol
TLS	Transport Layer Security
TTLS	Tunneled TLS
PEAP	Protected Extensible Authentication Protocol
A	Authenticator
AS	Authentication Server
AP	Access Point
RADIUS	Remote Authentication Dial-In User Service
MSCHAP	Microsoft Extension to Challenge Handshake Authentication Protocol
MIMA	Man in the middle attack
CA	Certificate Authority
USIM	Universal Mobile Telecommunications System Subscriber SIM card
RFID	Radio-frequency identification
AKPS	Asymmetric Key Pre Distribution scheme
PUF	Physically Unclonable Functions
KDC	Key Distribution Center
TMC	Trust Manager Component
QR-TAN	Quick Response Transaction Authentication Numbers
RPC	Remote Procedure Call
BPNN	Backpropagation Neural Network Technique
DoS	Denial of Service
DDoS	Distributed Denial of Service
LATE	Longest Approximate Time to End
HDFS	Hadoop Distributed File System

HEAP	Hadoop Efficient Authentication Protocol
SOF	Single Point of Failure
SOV	Single Point of Vulnerability
TPM	Trusted Computing Platform
TPSS	Trusted Platform Software Stack
CS	Client-Server
MABFWA	Mobile Agent Based Framework for Wireless Authentication
FAST	Flexible Authentication via Secure Tunnelling
KDFBA	Key Distribution Through Fingerprint based Authentication using Mobile Agent
KDFBWA	Key Distribution Fingerprint based Without Authentication
AODV	Ad Hoc On-Demand Distance Vector
PDR	Packet Delivery Ratio
LCF	Local Closest First
GCF	Global Closest First
ABMR	Agent based MapReduce
JADE	Java Agent Development Framework

TABLE OF CONTENTS

CHAPTER 3

MOBILE AGENT BASED FRAMEWORK FOR WIRELESS AUTHENTICATION (MABFWA)

CHAPTER 4

KEY DISTRIBUTION THROUGH FINGERPRINT BASED AUTHENTICATION USING MOBILE AGENT (KDFBA)

4.1 Introduction

4.2 Key Distribution through Fingerprint based Authentication

4.2.1 Performance Metrics

4.2.2 Simulation Result Analysis

(a) Effect of network density i.e. number of nodes on Mobile Agent using KDFBA and KDFBWA Algorithms

(b) Effect of Simulation Time variation on Mobile Agent using KDFBA and KDFBWA Algorithms

(c) Effect of network connection variation on Mobile Agent using KDFBA and KDWA Algorithms

(d) Effect of packet size change on the Mobile Agent using KDFBA and KDFBWA Algorithms

4.3 Comparison Analysis

CHAPTER 5

DEVICE FINGERPRINT BASED AUTHENTICATION USING MOBILE AGENTS

CHAPTER 6

BIG DATA PROCESSING AND AUTHENTICATION OF DEVICES USING MOBILE AGENT

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

7.2 Contributions

7.3 Future Scope

REFERENCES

BRIEF BIODATA OF THE RESEARCH SCHOLAR

CHAPTER 2

RELATED WORK

CHAPTER 1

INTRODUCTION

APPENDICES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Wireless communication is the exchange of data between two or more points that are not joined by an electrical transmitter. The most well-known wireless technologies use electromagnetic wireless telecommunications, for example, radio frequencies or infrared waves [1]. With radio waves, distances could be either short, for example, couple of meters for TV remote control, or to the extent that thousands or even a huge number of kilometres for profound space radio communications. It includes different sorts of fixed, mobile and portable applications, including two-way radios, cell phones, individual PDAs, and wireless networking. Figure 1.1 shows an example of wireless communication. The various available wireless technologies differ in local availability, coverage range and performance, and in some circumstances, users must be able to employ multiple connection types and switch between them. Supporting technologies include Wi-Fi, Cellular data service, Mobile satellite communication, Wireless technology [2] [3].

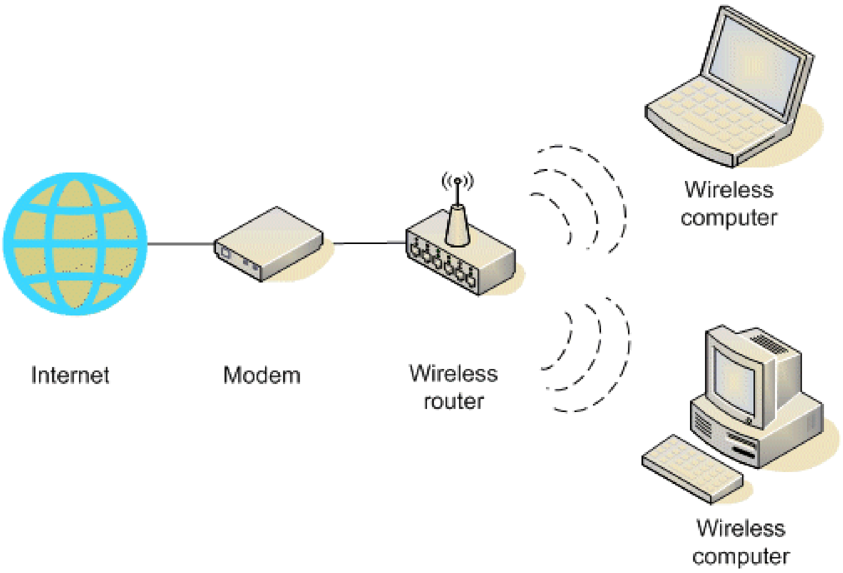


Figure 1.1 Wireless Communication

Wi-Fi is a wireless local area network that enables portable computing devices to connect easily to the internet [4]. Standardized as IEEE 802.11 a/b/g/n, Wi-Fi approaches speeds of some types of wired Ethernet. Wi-Fi has become normal standard for access in private homes, within offices, and at public hotspots.

Cellular data service offers coverage within a range of 10-15 miles from the nearest cell site. Speeds have increased as technologies have evolved, from earlier technologies such as GSM, CDMA and GPRS, to 3G networks such as W-CDMA, EDGE or CDMA2000 [5][6].

Mobile Satellite Communications may be used where other wireless connections are unavailable, such as in largely rural areas or remote locations. Satellite communications are especially important for transportation, aviation, maritime and military use [7].

Wireless technology permits services, such as long range communications, that are impossible or impractical to implement with the use of wires [8]. The term is commonly used in **telecommunication industry** to refer to telecommunications systems (e.g. radio transmitters and receivers, remote controls, computer networks, network terminals, etc.) which use some form of energy (e.g. radio frequency (RF), infrared light, laser light, visible light, acoustic energy, etc.) to transfer information without the use of wires. Information is transferred in this manner over both short and long distances [9] [10]. The following situations justify the use of wireless technology:

- To span a distance beyond the capabilities of typical cabling,
- To provide a backup communications link in case of normal network failure,
- To link portable or temporary workstations,
- To overcome situations where normal cabling is difficult or financially impractical, or
- To remotely connect mobile users or networks [9].

Wireless technology is becoming more and more popular due to so many advantages. Security is one of the most important aspect for the success of these wireless technologies. Researchers have proposed various security techniques in terms of authentication [12], secure transmission [13], encryption [14] and key exchange [15]. Mobile agent [16] for wireless security is one of the approach which is less tested. This technology is continuously evolving and generating lot of attention among researchers. Mobile agent technology has some great benefits in terms of bandwidth utilization and traffic generation. Mobile agent (MA) has both mobile code, data and is able to migrate from one computer to another. So, MA is intelligent enough that it can take decision based on certain conditions [17]. MA has various applications in different domains but in wireless security it is less tested. An extensive survey of related work indicated that there is a need of framework which can help in reducing traffic for authentication, efficient key distribution with authentication and device signature based authentication.

1.2 NEED OF WIRELESS SECURITY

Security is one of important challenge which is to be handled in the era of wireless technology. Current security standards have shown that security is not keeping up with the growing use of wireless technology [18]. Time and again a new vulnerability comes in existence in existing wireless standards. Wireless security is the prevention of unauthorized access or damage to computers using wireless networks. Wireless networks are very common, both for organizations and individuals. Many laptops, computers have wireless cards pre-installed. The ability to enter a network while in motion has great benefits. However, wireless networking has many security issues. Hackers have found wireless networks relatively easy to break as compared to wired networks. As a result, it's very important that enterprises define effective wireless security policies that guard against unauthorized access to important resources. **W**ireless **I**ntrusion **P**revention **S**ystems (WIPS) [19] are commonly used to enforce wireless security policies.

The risks to users of wireless technology have increased as the service has become more popular. However, there are many security risks associated with the current wireless protocols and encryption methods. They exist at the user and corporate IT level due to carelessness and ignorance. Breaching wireless security has become much more sophisticated and innovative with wireless. Cracking has also become much easier and more accessible with easy-to-use and free Windows or Linux-based tools available on the internet.

1.3 SECURITY THREATS TO WIRELESS NETWORKS

Protection of wireless networks means protection from attacks on confidentiality, integrity and availability. Possible threats come from vulnerabilities in the security protocols. This section explains various types of security attack techniques. These techniques can be applied to violate either confidentiality or integrity, or both [20]. Different types of security attacks are shown in the Figure 1.2.

Traffic analysis: This technique enables the attacker to have access to three types of information [21]. The first type of information is related to identification of activities on the network. The second type of information is identification and physical location of access point in its surroundings. Thirdly, information about communication protocol is also important which attacker can get by doing traffic analysis. An attacker needs to gather the information about size and number of the packets over a certain period of time.

Eavesdropping: In case of eavesdropping [22] attacker secretly listens to the private conversation of others without their permission. Eavesdropping attacks include passive eavesdropping, active eavesdropping with partially known plaintext and active eavesdropping with known plaintext.

Passive eavesdropping is used to watch over an unlimited wireless session. The only condition to be fulfilled is that attacker has the access to the area of emission. With a decrypted session, the attacker is able to read the data during its transmission and gather data indirectly by surveying the packets. This kind of attack is not based on violation of privacy but information gathered in this way can be used for more dangerous kinds of attacks.

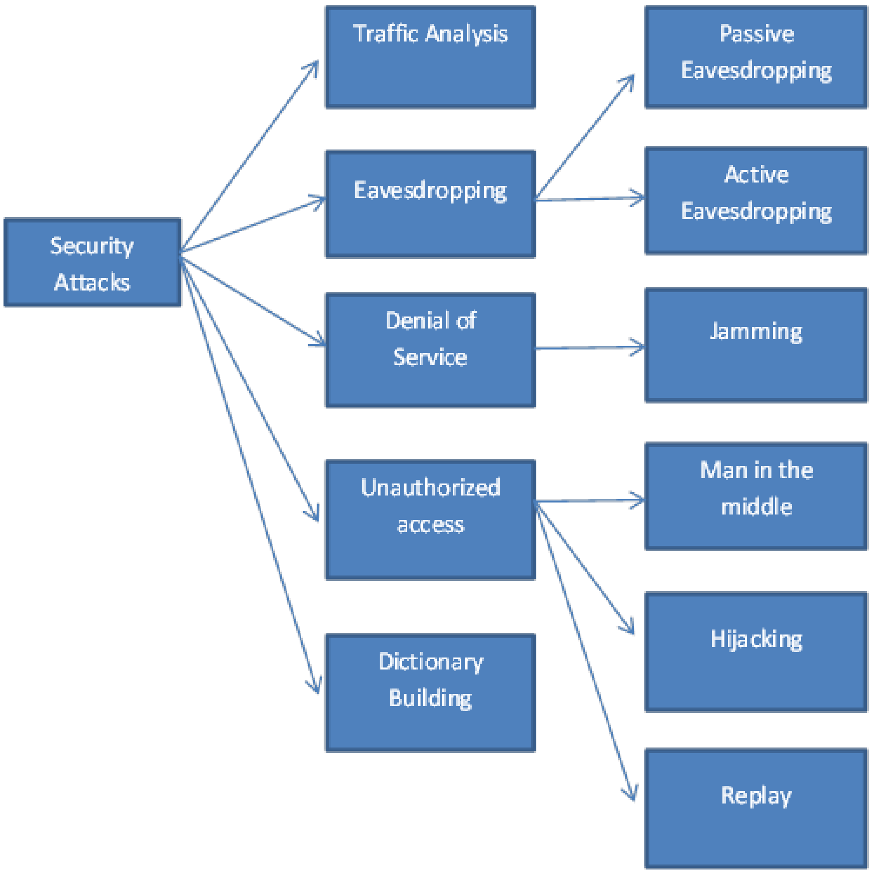


Figure 1.2 Different types of Security Attacks

In Active eavesdropping with partially known plaintext type of attack, the attacker watches over a wireless session and actively injects own messages in order to reveal the content of the messages in the session. Precondition for this type of attack is an access to communication area and some knowledge on the part of the message, such as IP address. The attacker is able to modify the content of the packet so that the integrity of the message remains preserved. Usually the attacker changes final IP or TCP address.

In active eavesdropping with known plaintext type of attack, attacker injects messages known only to him into the traffic in order to create conditions for decryption of the packets that should be received by other wireless users.

These conditions are created by creation of Initialization Vector (IV) sequence and message for each single message that is sent. After some time, when a packet with the same IV as in database appears, the attacker is able to decrypt the message. The only way to prevent this kind of attacks is to change key often.

Unauthorized access: Once the attacker gets access to the network, attacker will be able to initiate some other types of attacks or use network without being noticed. Some can be of an opinion that unauthorized use of the network is not a significant threat to the network since the access rights allocated to resources will restrict the attackers to use them. However, the unauthorized access is the key to initialize ARP (Address Resolution Protocol) attack [23]. Virtual Private Network (VPN) and IPsec solution can protect users from the attacks that directly influence the confidentiality of application data but cannot prevent attacks that indirectly ruin confidentiality. Man in the middle, high-jacking and replay attacks are the best examples of these kinds of attacks.

Man in the middle attack (MITM) enables data reading from the session or modifications of the packets which violate integrity of the session. There are several ways to implement this type of attack [24]. One way is when attacker disrupts the session and does not allow for the mobile station to establish communications again with the Access Point (AP). Mobile station tries to establish session with the wireless network through AP, but can do that only through the workstation of the attacker pretending to be AP. At the same time, the attacker establishes connection and performs authentication with the AP. Now, there are two encrypted tunnels instead of one: one is established between the attacker and the AP, while the second one is established between the attacker and the station. This enables attacker to have access to the data exchanged between working station and the rest of the network. ARP attack is a sub-type of the MITM attack since these attacks are directed towards one component of wired network and not towards wireless clients [25]. The attacker escapes authentication or provides false accreditations by this kind of attack. The attacker becomes valid user and gets access to the network as authenticated user by getting false accreditations.

In ***High-jacking*** attack [26], the attacker deprives the real owner of the authorized and authenticated session. The owner knows that there is no access to the session any more but is not aware that the attacker has taken over the session and believes that session is lost due to some network problem. Once the attacker takes over a valid session it can use it for various purposes over a certain period of time. This attack happens in a real time.

Replay attack [27][28] is used to access the network through authorization. The session, that is under an attack neither changes nor disrupted in any way. The attack does not happen in a real time. The attacker gets access to the network after original session expires. The attacker comes to the authentication of one or more sessions, and then replies to the session after a certain period of time or uses couple of sessions to compose the authentication and reply to it.

Denial of Service (DoS): An attacker tampers with the data before it is communicated to the sensor node. It causes denial of service attack due to wrong or misleading information. Jamming is one of DoS attack on network availability [29]. It is performed by malicious attackers who use other wireless devices to disable the communication of users in a legitimate wireless network [30].

Dictionary-building attacks: In these type of attacks, an attacker goes through a list of candidate passwords one by one; the list may be explicitly enumerated or implicitly defined. In this way, attacker can incorporate knowledge about the victim, and can be linguistically derived [31]. Dictionary building attacks [32][33] are possible after analysing enough traffic on a busy network.

So, even there are number of security protocols in existence and different authentication protocol hackers or intruders are still able to get either illicit access to the network or able to deny the access to the network to other users [34][35]. Attackers are able to deploy attacks like MITM, high-jacking, replay attack, DoS and dictionary building attacks.

1.4 PROBLEM IDENTIFICATION

After having extensive literature survey, some problems have been identified in the wireless domain of security of networks. These problems fall under the category of authentication, key distribution/key exchange and Big Data processing in the wireless domain.

Authentication: As different mobile nodes come in and out of the network frequently and due to the broadcast nature of these networks, authentication problem is an issue in the wireless networks. Existing authentication protocols either generates too much traffic or rely on third party certificates.

Load on the Key Distribution Center (KDC): Due to client server based nature of authentication algorithms, load on KDC increases as the number of clients increases in the network. So, scalability of KDC is an issue.

Key Exchange: Secure wireless network communication requires encryption of the data to be communicated and for the encryption algorithms key exchange is a problem. Key exchange with authentication of the nodes is an issue in the wireless network.

MapReduce: In the wireless network secure, balanced distributed computing is also an issue in terms of Big Data processing. Selection of speculative task and recognition of the machine in terms of stragglers is a challenge. Stragglers are the nodes which run slower than normal. Reduction of traffic generated is also a challenge.

1.5 RESEARCH OBJECTIVES

The main objective of the research is to study and identify issues related to wireless security. After doing literature survey, it is observed that existing protocols have some serious flaws in terms of key exchange, authentication and Big Data processing in wireless domains. Therefore objectives of the proposed work are:

- To study and identify problems in existing authentication protocols of wireless network and design a new mobile agent based framework for authentication.

- To study and identify problems related to key exchange and propose a new mobile agent based key exchange method.
- To simulate compare and analyze existing key exchange approaches with the proposed key exchange method.
- To study and identify existing techniques for authentication and propose a device fingerprint technique for authentication.
- To develop an application for creation of device fingerprint.
- To create a mobile agent based MapReduce framework for Big Data processing in wireless network.

1.6 METHODOLOGY

The methodology used during the research work includes Literature Survey, Problem Identification, Proposed Work, Simulation and Result Analysis.

1.7 CONTRIBUTIONS OF RESEARCH WORK

The main contributions of this research work are presented as follows:

Mobile Agent based Framework for Wireless Authentication (MABFWA) for implementing multiple authentication protocols. This new framework helps in incorporation of multiple authentication algorithms. With the help of mobile agent based technology, the traffic around the authenticator using this framework has been reduced to a considerable amount as compared to the EAP based framework. Algorithms like MD5, TLS and PEAP are tested on MABFWA framework and numerical analysis has showed that there is considerable amount of traffic reduction around the authenticator as compared to EAP based approach.

Key Distribution through Fingerprint based Authentication using Mobile Agent (KDFBA). A new biometric based key exchange protocol (KDFBA) is proposed which makes use of mobile agent for key exchange with proper authentication. Proposed protocol has been simulated on NS2 platform and is also compared with the existing approaches available against some parameters like traffic generated and timing analysis. Protocol is also tested against black hole attack and results for the same have also been compared with the existing approach. Results show that the proposed key exchange protocol reduces the traffic on the network considerably and also prevents the black hole attack.

Device Fingerprint based Authentication mechanism using Mobile Agent. Device features have been extracted and used for device fingerprint generation. These generated device fingerprints are used for authentication. Proposed scheme is implemented on Android platform for extraction of various parameters. Using these parameters, MD5 fingerprint has been generated which can be used for authentication of device on the network.

BigData Processing and Authentication of nodes using Mobile Agent. A new **A**gent **B**ased **M**ap**R**educe (ABMR) algorithm, which uses Mobile Agent (MA) for task splitting and relocation, is proposed. MA moves from one location to another based on how the scheduler schedules the task. For scheduling and relocation, scheduler calculates the **P**erformance **S**core (P_{score}) of each node. UsingP_{score}, scheduler maintains two arrays; one for slower machines and other for faster machines. The scheduler reschedules MAs running on slower machines to the faster machines one at a time depending upon the P_{score}. The scheduler finally recollects all the results after each MA has done its work and then presents it to the user of the application. Implementation is done on Java Agent Development Environment (JADE) and results are compared with the Hadoop native scheduler. Proposed ABMR gives better results in terms of straggler detection as compared to native scheduler.

1.8 ORGANIZATION OF THESIS

Chapter 1 provides a general overview of wireless security and introduces main challenges faced by wireless network in maintaining security of the network. This chapter explains motivational factors which are helpful in carrying out the research.

Chapter 2 explains a background detail on wireless security in the wireless network and discusses the related work in this area. In this chapter, a literature survey regarding different approaches used in this research wok is discussed.

Chapter 3 presents the design of authentication framework and describes Mobile agent based framework for wireless authentication (MABFWA). This chapter explains EAP-TLS, EAP-MD5 and EAP-PEAP algorithms on MABFWA and gives a numerical analysis in terms of traffic generated on the network.

Chapter 4 defines overall design considerations for the proposed system. It begins with a brief description, motivation and long term vision of the system. It explains detail architecture and components of the proposed Key distribution through fingerprint based authentication using mobile agent (KDBFA).

Chapter 5 presents the device fingerprint and mobile agent based authentication technique in wireless networks. Device features are extracted and using these features device fingerprint has been generated. This chapter also includes an android mobile application for fingerprint generation and its uses.

Chapter 6 defines the design of Mobile agent based MapReduce framework for BigData processing. This chapter includes implementation of the algorithm on the JADE platform and is compared with the Hadoop native scheduler.

Chapter 7 summarizes contributions of the proposed work and discusses possible future directions of the proposed work.

CHAPTER 2

RELATED WORK

2.1 INTRODUCTION

Wireless security comprises of methods of authentication, integrity, non-repudiation and confidentiality. Authentication is the system to confirm the legitimacy of a client. In the event, if the user is authentic then server permits him or her to access network resources. It is necessary for a network to utilize a robust authentication protocol so that no other client than a valid one can access the network. An individual, who is not a legitimate client and is trying to hack and to get to the system, must be blocked [36]. There are number of authentication techniques available, which provide different level of authentication. These techniques vary from simple authentication methods like a secret code or username password to more complex ones like smart card or fingerprint based methods. Many researchers have proposed different authentication protocols in past. Next section, will describe need of authentication, different methods of authentication and protocols proposed by various researchers in the past.

2.2 NEED OF AUTHENTICATION

Authentication is the methodology which permits sender and recipient to authenticate each another. If sender or receiver cannot appropriately validate each other, then there is no trust in data exchange between two parties. Let’s understand this with the help of an example: in a banking system, when someone deposits cash in other person’s account, no confirmation is required but when the cash is withdrawn, then it requires validation. For this it requires either form of signature or pin number [37]. So that, no person other than the authentic user can make any transaction without the consent. So validation is a vital thing and it relies upon the kind of data to be used.

2.2.1 METHODS OF AUTHENTICATION

There are different types of methods currently available for authentication. Currently there are many authentication methods available [36] .Some of the basic types are:

Using Card and Pin: In this type of strategy, a pin is given against each card. Card needs to be swiped and after that correct pin is to be entered. In case of ATM, pin is a four digit numeric code.

Signature Verification [37]: This kind of authentication mechanism is used in banks when cash is withdrawn with help of passbook or cheque. In this, a form or cheque is signed and bank personnel validates the signature with the earlier in the database. If it matches then person is authorized to do the withdrawal transaction. If it does not match then user is not allowed to do transaction.

Fingerprints [37]: Fingerprint is also one of the verification techniques to check legitimacy of the client. The idea behind this is likelihood of two individuals having same unique finger impression is low.

Hand Geometry [38]: This innovation was previously used in Olympics to authorize sport's persons participating. In this, first client is asked to put his hand into a box which consists of hand geometry reader. This box makes a bio metric layout of hand which is stored as user's ID badge. This badge also holds photograph of individual. At the point when any client likes to get access to the system he was asked to put his hand into the reader. If hand geometry matches, then user is a legitimate user.

Smart Card [39]: This type of concept is used in companies in which every employee of the company is assigned an ID card. When a person enters in the company, he or she has to swap that card. If that card is legitimate then respective employee can enter otherwise not. But in some cases, cards are not swiped but they are just shown to the verifier and now it is his or her job to verify id of the person.

Username and Password [40]: This method is most widely used in online login system. In this method, whenever a person access account online, he/she needs to enter a username and password. If username and password combination matches with the records then server permits access of system otherwise, access is denied.

Voice Recognition [41]: Lots of research has been carried out to enhance automatic speech recognition. In this, a system stores the voice of a user. When user speaks the same words, voice of the user is matched with the one in its records. If it matches, it means user is a valid user and can access the system.

Image Password [42]: In this, password is stored in the form of an image. One have to select set of images as a part of password in a particular pattern. When next time that person log in, he or she has to select the same pattern of images. The main advantage of this method is that remembering alphanumeric contents is difficult than remembering images. Also different types of attacks like dictionary attack, brute force etc. do not works on it.

Retina[43]: A person’s retina is unique and can help us to build very reliable authentication method. The problem with this method is that people tend to display a natural fear about damaging their eyes and they are thus anxious that the device reading their retina might not be safe.

Face Recognition[44]: It is used for facial recognitions. In this, user’s face is used as a password. This technology can commonly be seen in laptops for login. Even new techniques are there which can differentiate between someone holding photo before the camera and an actual person sitting in front of the camera.

Typing Speed[45]: This is a biometric innovation focused around behavioural action. It is normally hidden from the user itself. For this, an additional test is conducted with focus on the typing speed of the user to attain a more precise authentication.

2.3 PROTOCOLS FOR AUTHENTICATION

With increasing number of confidential information accessible over the internet, unauthorized persons should be kept away from accessing the same. For this, there is a requirement of an efficient authentication algorithm. There are currently multiple authentication methods as described in previous section. Apart from these methods, Extensible Authentication Protocol (EAP) is most widely used authentication protocol [46]. In fact, EAP provides a framework over which multiple authentication protocols can run. This framework is also used in 802.11 (Wi-Fi) and 802.16 (Wi-Max) standards [47]. With the help of EAP authentication, information between client and the network is exchanged. EAP provides request-response message exchange framework over which multiple types of authentication protocols like MD5, LEAP, TLS, TTLS and PEAP etc. can run [48]. As shown in Figure 2.1, there are three entities which are involved in any authentication process.

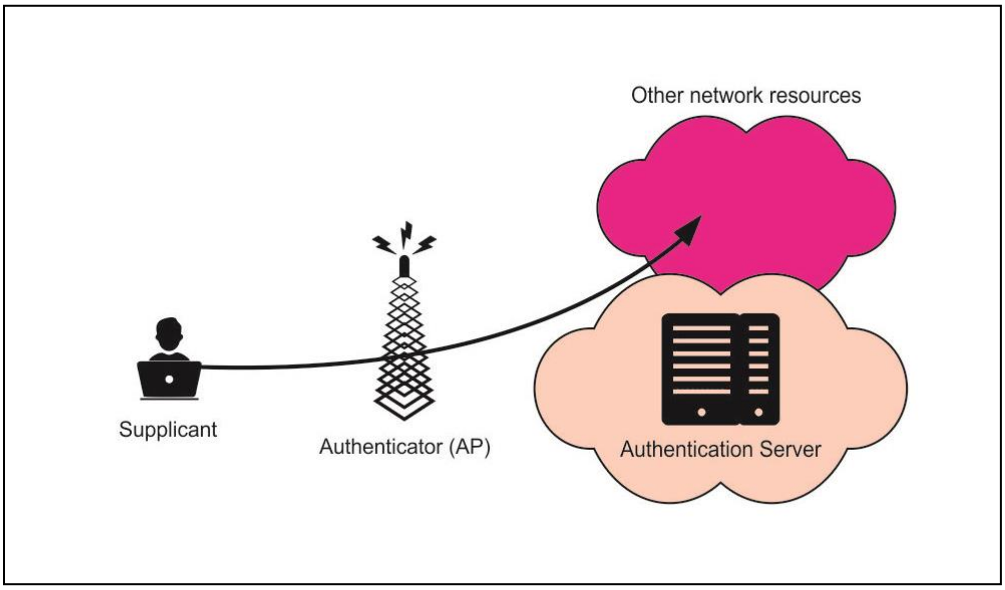


Figure 2.1 EAP entities

Supplicant/Client: The supplicant or user is a client which wants to access the network. This is generally the **M**obile **S**tation (**MS**) through which user accesses the network services.

Authenticator (A): Authenticator is an access point which provides access to the network with the help of radio waves. Prior successful authentication is mandatory for the access of the network.

Authentication Server (AS): Authentication server firstly negotiates with the client for the type of authentication algorithm. After negotiation, it validates the client depending upon credentials and provides network access to the user. AS is generally **R**emote **A**uthentication **D**ial-**I**n **U**ser **S**ervice (RADIUS) server [49].

Figure 2.2 shows basic message exchange framework where initially client and authenticator exchanges individual identity messages. After this, number of message exchanges are done between client and authenticator depending upon the type of authentication protocol [50]. Similarly, number of message exchanges between authenticator and authentication server is also done.

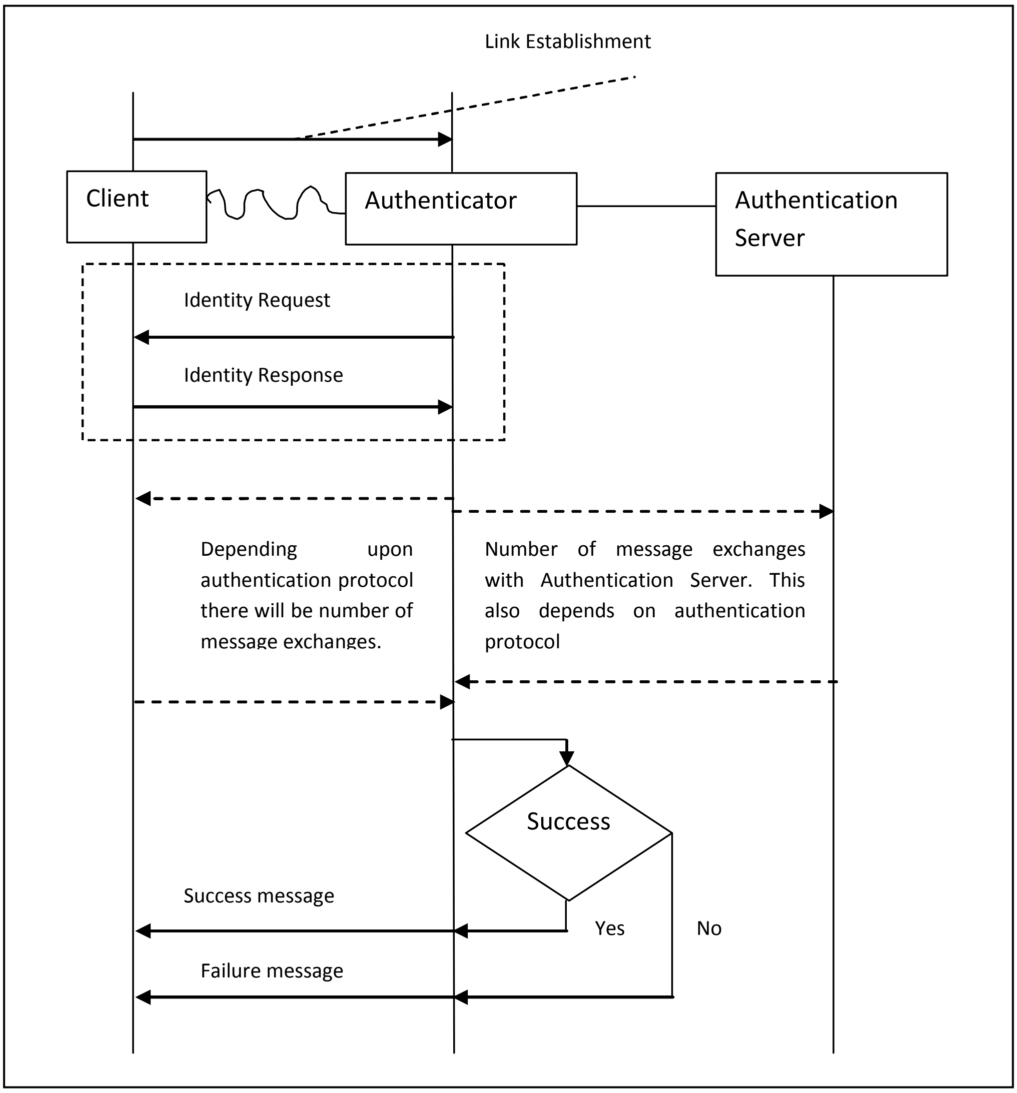

Link Establishment
Client
Authenticator
Authentication Server
Identity Request
Identity Response
Depending upon authentication protocol there will be number of message exchanges.
Number of message exchanges with Authentication Server. This also depends on authentication protocol
Success
Success message
Yes
No
Failure message

Figure 2.2 EAP Message Exchange Framework [51]

As EAP supports various security protocols, some of currently available protocols are discussed below:

2.3.1 MD5 [52]: Message digest (MD5) is focused on one way hash function. It is an EAP well known method. In hash function, a message of variable length is taken and a fixed length output called hash is received. This is used to check legitimacy of client only. In this, client's password is not stored in plain text and also not sent through any medium in plain content. MD5, methodology begins with the registration stage in which client enters his username, secret key and other details required by the server.

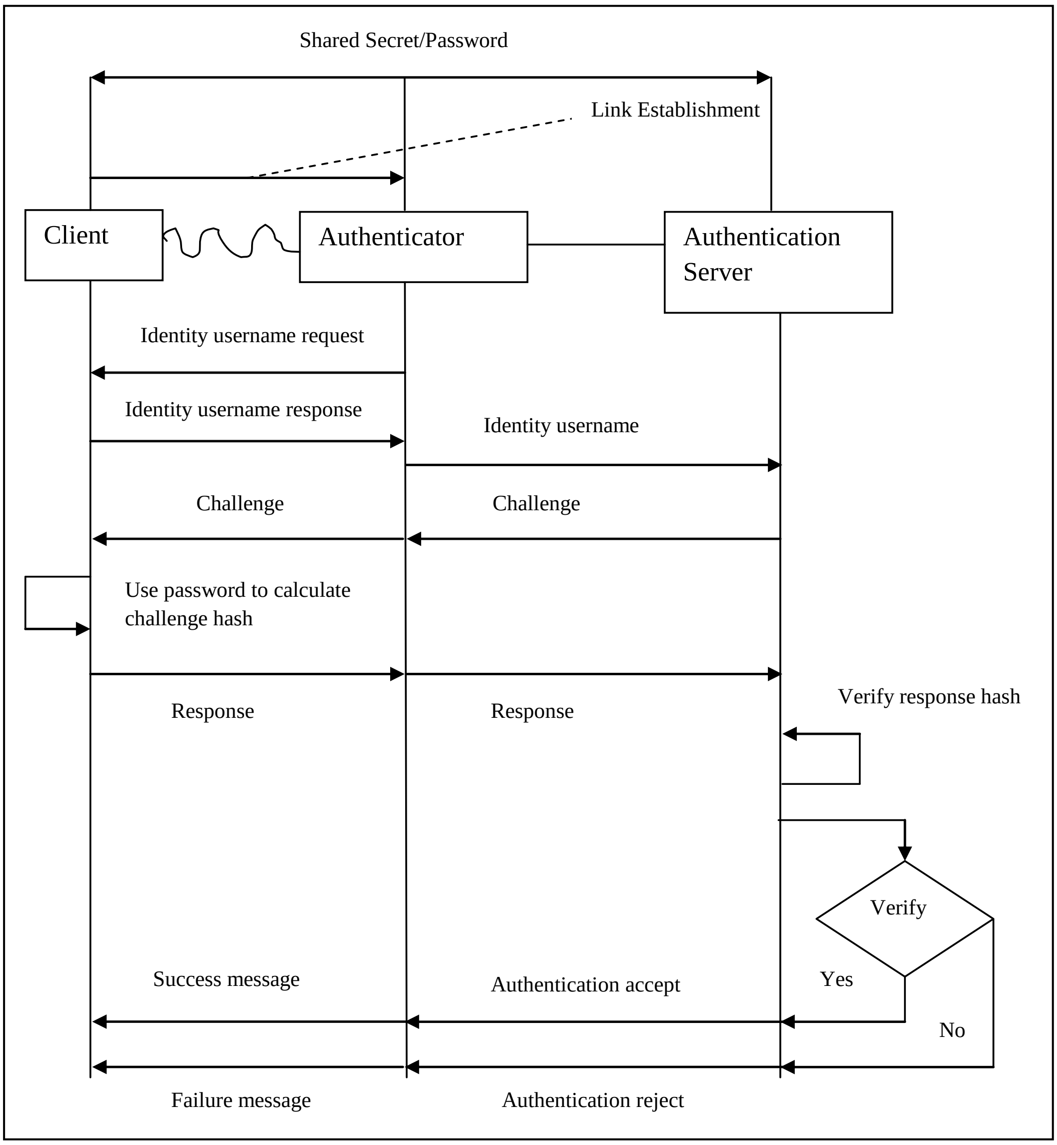


Figure 2.3 EAP Message Exchange using MD5 [51]

At the point when client submits data then hashing algorithm present at the client side ascertains hash of the password and sends it to the Authentication Server (AS) via Authenticator (A). Figure 2.3 shows complete message exchange of EAP-MD5 scheme. Server stores hash of secret key against each username. When client enters password then hash of the password is calculated and sent to the AS. AS compares both stored and received hash value. It shows welcome message after successful matching. The primary preference of this convention is that it is easy to implement.

Second, if some way or another someone gets that password, he/she can't decrypt because producing the message having same hash value is extremely difficult. There are few disadvantages of this technique. There is no mutual authentication in this method which causes uncertainty in client server validation. It doesn't infer a session key for its each session. It also suffers from different type of attacks like reply attack, birthday attack, dictionary attack etc. It doesn't meet prerequisites given in the RFC 4017 [53]. So, it should not be utilized for wireless communication.

2.3.2 LEAP [54] [55]: LEAP (Lightweight Extensible Authentication Protocol) is developed by Cisco. It is based on challenge and response procedure. It also covers two major weaknesses of **W**ired **E**quivalent **P**rivacy (WEP) i.e. mutual authentication and session key [56]. Complete LEAP message exchange is shown in Figure 2.4.

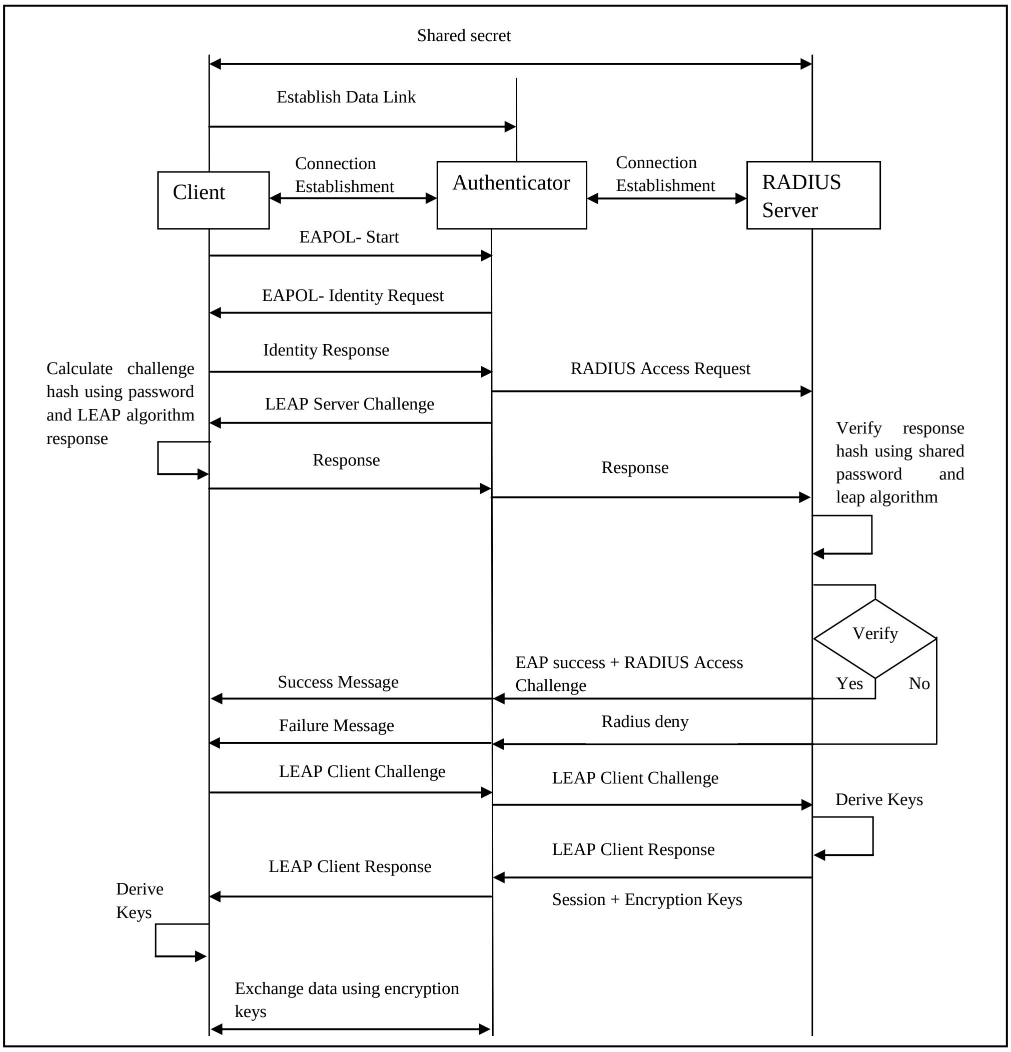


Figure 2.4 EAP Message Exchange using LEAP [51]

LEAP authentication uses a pre shared secret key. At first, client sends a random challenge to server. The server decrypts the challenge and responds it. The challenge response is encrypted with session key. The client decrypts the challenge with session key. If the value of the challenge is same as it is stored at client side then server is valid. Similarly server also verifies the client by similar method. MSCHAP (MicroSoft extension to Challenge Handshake Authentication Protocol) protocol is also used in this method [57]. As it overcomes drawback of WEP but it also suffers with attacks like identity protection because whole message is sent in plain text. And also a hacker can easily sense the challenge-response pair transfer between client and server. It also suffers with a common dictionary attack.

2.3.3 EAP-TLS [58]: EAP-TLS is defined in RFC 2716 [59]. TLS (Transport Layer Security) is a standardized version of SSL (Secure Socket Layer) [60]. It is developed by Microsoft. It is a certification based Authentication method for wireless LAN. It installs certificate on both client and server based on X.509 [61] to provide dynamic session key distribution. At first, client sends a random request to server. In response, server sends its public key certification to the client and request for client certification. As the client verifies the public key certificate, it sends its certificate encrypted with server public key certificate to the server. As the server receives client public key certificate, it verifies that certificate from the issuing party. In this, there is a third party who issues certificate to the client and the server. Complete TLS process is shown in Figure 2.5.

It is also possible that both (client and server) are certified by different organization. Any organization who issues certificate charge some fixed amount of fee after a fix interval of time. As the certificate verification is done, server sends a session key encrypted with client public key. Messages encrypted by public key of user can be decrypted by private key of that particular user only. This results in resisting most of the attacks like replay attack, MIMA (Man in the middle attack) [62] etc. It also supports fast reconnect defined in RFC3748 [63].

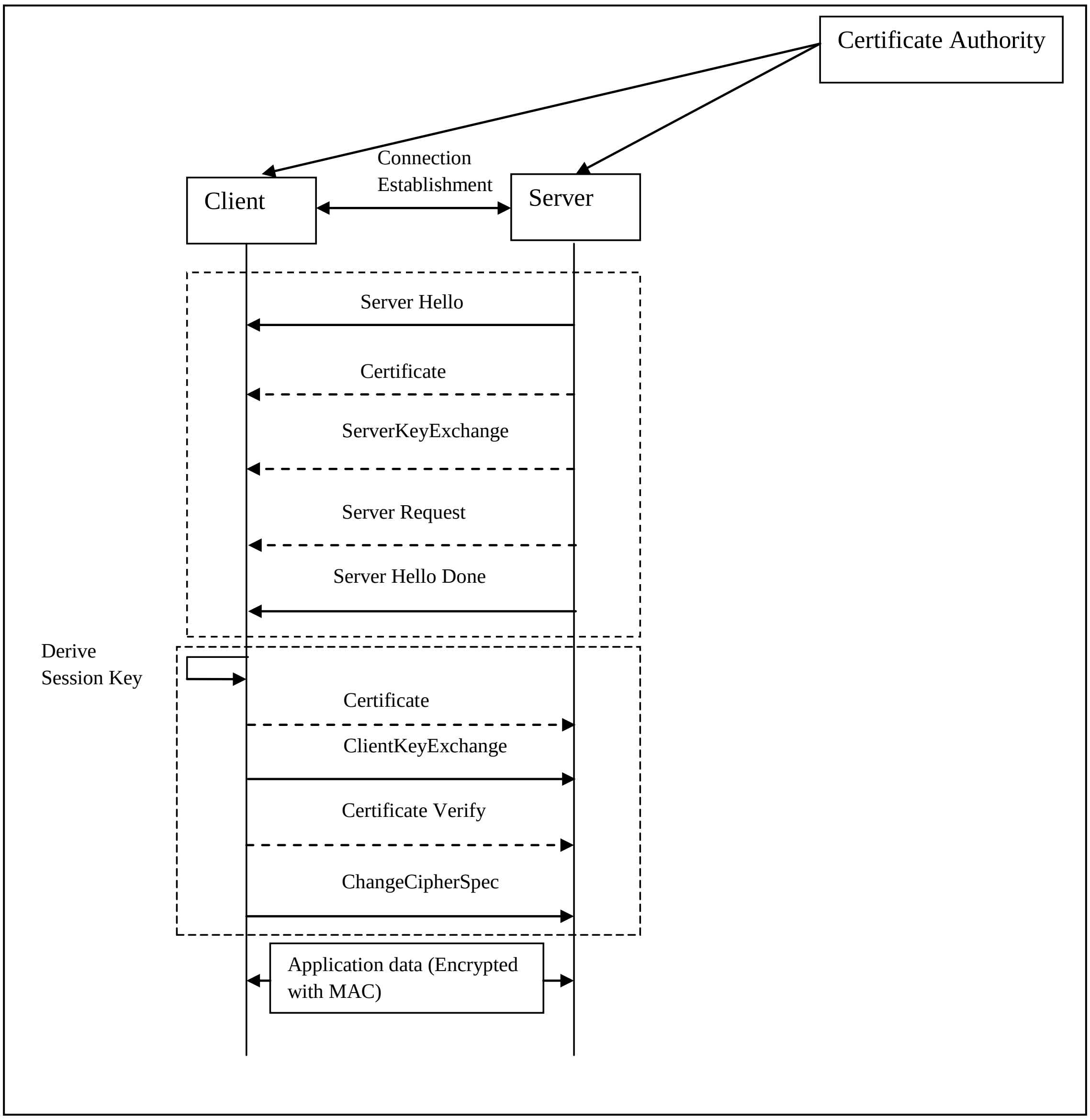


Figure 2.5 EAP Message Exchange using TLS [51]

2.3.4 EAP-TTLS [64][65]: EAP–TTLS is developed by Funk Company to solve TLS certificate problem (EAP-Tunneled TLS). It consists of two steps. In first step, client authenticates server with the help of its certificate and also derives a session key. Secondly, the client is authenticated in tunnel. Tunnel is created to maintain confidentiality of the information. EAP-TTLS provides high security during authentication process. It supports almost all authentication protocols including legacy protocols and avoids the use of pre key infrastructure on the client side. This results in reduction of overall cost of implementation. With the help of tunnel, client's identity is hidden so that hacker does not get any information regarding communication as well as the user.

2.3.5 EAP-PEAP [66]: This is EAP Protected Extensible Authentication Protocol (EAP-PEAP). This is designed for wireless domain and is a wireless version of EAP-TLS. This protocol works in two phases. In first phase, a secure tunnel is created using EAP TLS. Second phase implements methods of EAP for client authentication. PEAP also supports quick re-authentication of the same user. Figure 2.6 explains the number of message exchanges between authentication server, client and server.

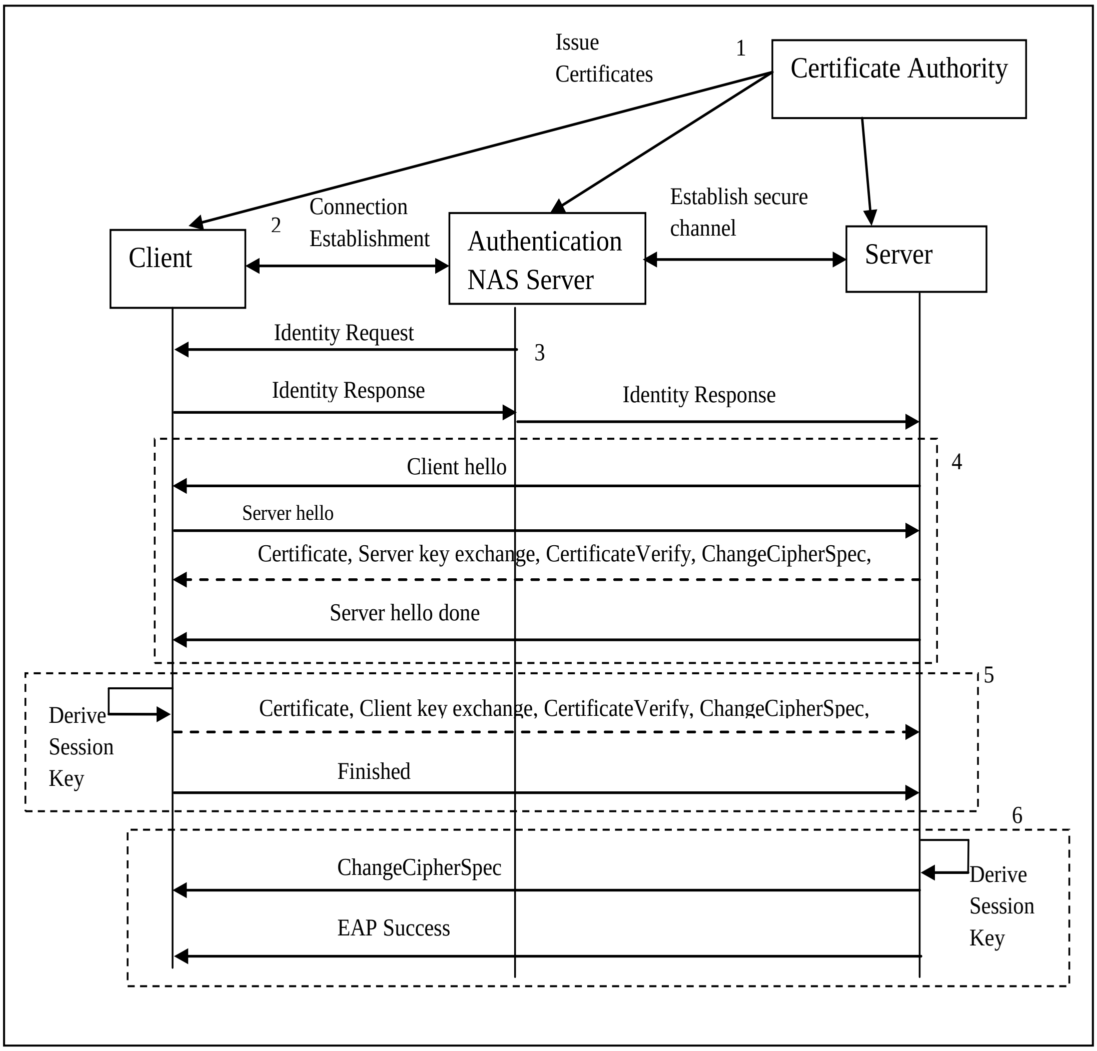


Figure 2.6 EAP Message Exchange using PEAP [51]

2.4 COMPARISON OF AUTHENICATION SCHEMES

Table 2.1 presents a multi-aspect qualitative comparison between five different protocols. Table compares important aspects like deployment, certification (whether it is a valid client or server), different type of attacks like dictionary, Man in The Middle (MITM), replay attack, cost and other security issues.

Table 2.1: Comparison of various authentication schemes

EAP-Type	**EAP-MD5**	**EAP-LEAP**	**EAP-TLS**	**EAP-TTLS**	**EAP-PEAP**
Mutual Authentication	No	Yes	Yes	Yes	Yes
Deployment Difficulties	Easy	Easy	Hard	Moderate	Moderate
MITM Attack Protection	No	Yes	Yes	No	No
Dictionary Attack Protection	No	No	Yes	Yes	Yes
OTP Generator	No	No	No	No	No
Delivery of password by Mail and SMS	No	No	No	No	No
Low Cost	Yes	Yes	No	No	No
Remember Password	Yes	Yes	No	Yes	Yes
Reuse Attack Protection	No	No	Yes	No	No

2.5 RELATED WORK

Apart from these traditional protocols discussed above, there are various researchers who are continuously working in the field of authentication. Some of these are:

Soo-Cheol Kim et al. [67] has proposed a personalized light weight authentication protocol for **I**nternet **P**rotocol **T**ele**v**ision (IPTV) services. This protocol makes use of RFID for authentication of user. User authentication information is stored in the **S**et **T**op **B**ox (STB) with the help of Universal Mobile Telecommunications System Subscriber SIM (USIM) card. Protocol supports two kinds of authentication; one is for critical tasks and second is for less critical tasks. Critical tasks authentication is done with the help of USIM and less critical authentication is provided with the help of Agent tag. This protocol is a very light weight authentication protocol which requires very few steps for the authentication. The main drawback of this scheme is that it requires extra hardware in terms of RFID and USIM card.

Ya-ling Zhang et al. [68] has proposed a mobile agent based authentication protocol which makes use of threshold theory. This protocol uses mobile agent technology for dynamic and cooperative authentication. Threshold theory used by the protocol is (t, n) where t<n. Here the communication between atleast ‘t’ members is required for the creation and use of the key. They have also proposed a method for authentication of certificates issued by the certificate authority. Every communication in the protocol takes place with the help of mobile agent. So, protocol supports all benefits of mobile agent in terms of interoperability, autonomy and ability to take decisions. This protocol also reduces the traffic as compared to client server technology, as less number of message exchanges is required for authentication.

Sandeep K. Sood [69] has proposed a new model of authentication which makes use of multiserver architecture. Multiserver architecture makes life of an attacker severe by sharing the information on more than one server. This model is an enhancement of the Liao and Wang protocol [70] which was having some flaws in terms of malicious server attack and malicious user attack. As mentioned, it uses more than one server and a different level of security mechanism is imposed on each server. This enhances the security of the system and prevents it from attacks like malicious server and malicious user attack [71]. Model is efficient in terms of computational cost but has some drawbacks in terms of lost or stolen smart card and control server key theft.

Chin-Ling Chen et al. [72] observed some drawbacks in Sandeep K. Sood’s scheme in terms of lack of authentication between server and user. So, they have proposed a new authentication scheme for multiserver architecture which has guaranteed mutual authentication and prevention of attack in which legal user thefts the control server key. Computation cost of this scheme is higher as compared to the previous scheme but it provides added security.

Xiumei Liu et al. [73] has proposed a password based group authentication protocol. This group authentication protocol greatly reduces the traffic as multiple clients use shared encryption key. Firstly, every client chooses a random password and registers with the server. If there are n clients then n passwords are shared with the server to produce an encryption key. This protocol provides improved computational cost for clients and server. Communication cost between client and server is also improved to a great extent. Proposed protocol also resists against attacks like dictionary attacks and man in the middle attack.

Gang Yao et al. [74] has proposed a group authentication protocol which is based on 2^d – cube group key exchange protocol. This protocol works in three stages. In first stage, clients get temporary session key by sharing independent password with the server. In second phase, group key exchange occurs and in last stage, key confirmation is done. This protocol is secure against many attacks like password guessing and provides a good authentication mechanism for multiple clients in group. Table 2.2 summarizes various multiple authentication protocols discussed so far with their advantages and disadvantages.

Authentication protocols discussed above have been evolved over time and provide some real benefits for authentication of user. Apart from authentication, key exchange is always an issue in case of wireless networks. Key exchange and distribution is always an issue in wireless networks due to open broadcast nature of these networks. For successful operation of these networks, a robust encryption algorithm is required. The success of encryption algorithm in the network depends upon how the network manages required keys.

Table 2.2: Authentication protocols

Protocol	**Objectives**	**Methodology**	**Advantages**	**Disadvantages**
Soo-Cheol Kim et al. [67]	• Personalized authentication	• Authentication via Universal Subscriber Identity Module	• Minimizes the load factor on authentication server. • Reduction in	• Requirement of extra hardware in terms of RFID,

		(USIM) and via Agent tag	computing power	USIM
Ya-ling Zhang et al. [68]	• Virtual certificate authority model • To implement inter-verification between two different certificates • To set up a dynamic coordination mechanism among all members.	• Threshold Theory	• Interoperability of different certificate schemes.	• Use of multiple certificate trust chain takes long time.
Sandeep K. Sood [69]	• Smart card based authentication mechanism • Updation of Hsiang and Shih protocol against various attacks	• Secure dynamic identity based authentication protocol for multiserver architecture using smart cards	• Protection against various attacks like parallel session attack, man in the middle attack.	• Lost smart card or stolen smart card problem persists and legal user can steal the control server key.
Chin-Ling Chen et al. [72]	• Mutual authentication between server and user. • Up gradation of Sandeep K. Sood protocol.	• To achieve the mutual authentication between server and user.	• Protection against various attacks like parallel session attack, man in the middle attack as well as prevention of user to	• Smart card based authentication scheme. • Requires additional hardware and card reader.

			have control server secret key.	
Xiumei Liu et al. [73]	- nPAKE: Common session key between group members and server.	- To reduce communication cost and computation cost at the server while achieving the group authentication	- Reduces communication and computation cost at the server. Protects against dictionary attack, MITM, Forward secrecy	- If a client in the group leaves or the password of a client is compromised, the shared password has to be updated and no one is able to distinguish a client from another, and it is impossible for a subset of group to securely establish a session key.
Gang Yao et al. [74]	- Cube group key exchange protocol	- To use different password scheme for each user and password exchange between different users.	- Uses different password for group members. Algorithm is secure against many attacks.	- Authentication of the different entities communicating in the group is still a challenge.

2.6 KEY EXCHANGE

In the recent times various researchers have worked extensively on methods for management of keys. Zhihong Liu [75] et. al. have extensively studied the concept of key exchange and identified problems related to key exchange. Paper explained about very low storage requirements of wireless sensor nodes and proposes a new Asymmetric Key Pre Distribution scheme (AKPS). The proposed AKPS can also be applied to mobile sinks for partly storing the public keying material. Algorithm also provides a good tradeoff between public information size and user memory storage.

Some researchers like, Urbi Chatterjee [76] et. al. have used the concept of Physically Unclonable Functions (PUF's) to provide authentication of devices on the internet. Proposed algorithm makes use of identity based encryption, PUF's to authenticate and exchange of keys among multiple devices. Proposed algorithm proves that it is secure against Unauthenticated Link and Authenticated Link model. Algorithm also does not make use of certificate or any kind of certificate authority.

Apart from this PUF based scheme, smart cards have also been tested by Wen-Shenq Juang [77] et. al. who have proposed password authenticated key exchange scheme. This scheme makes use of smart cards. After registration, user is provided a smart card which is equipped with ID, hash of ID and password. Proposed scheme makes use of card reader whenever user wants to connect to the network. Once card is authenticated, user can make use of network resources. The session key generation can also takes place with the help of smart card. This scheme does not make use of certificate or any kind of certificate authority.

Some researchers concentrated on certificate less key management protocol i.e. CL-EKM. Seung-Hyun Seo [78] et. al. have worked on CL-EKM protocol which supports key updates as well as forward and backward key secrecy. This model provides a robust approach against adversaries who have illegal access of secret keys of different nodes. This protocol can also work with hierarchical encryption [79] and key agreement scheme. S. S. Al-Riyami [80] et. al. also have proposed the concept of certificate less public key cryptography. In this scheme the private key of the user is generated by the KDC partially i.e. the key generated by the KDC is not complete. The key is made complete by adding user's own secret value. Once the valid user on secret value and partial key from the KDC is combined the secret key is generated for decryption process.

Anjani[81] et. al. have proposed a new key management model which provides a good resilience against node capture attack. Model is basically for low connectivity problem of the hybrid symmetric design. The model uses Symmetric Balanced Incomplete Block Design (SBIBD) design to construct key rings. Algorithm makes use of a parameter d as the number of blocks which are to be merged. Work has been compared with different values of d, Hybrid Symmetric Design (HSYM), and SBIBD. Results show that proposed algorithm has better resilience as compared to other schemes.

Some researchers have concentrated on frequent movement of users in the network and maintenance of backward and forward secrecy in the network. Daghighi[82] et. al. has proposed a new approach named HISCOM. This approach performs well when users in the network move more frequently. This model makes use of a new rekeying mechanism which is based on the time when the member joins and the time when the key in the particular area is changed. This model increases scalability of key management and maintains backward and forward secrecy while in movement.

Key exchange helps in encryption of a session between communicating parties. Apart from encryption, user authentication is another big concern which needs to be addressed. Signature of a communicating device with in the network can be one of the mechanism for authentication.

Device fingerprint is a unique signature of the device [83]. Device signature mechanism uses some unique features of the device to calculate authenticity of the user or device within the network. Various researchers have proposed an authentication technique based on different methods of trust calculation or signature evaluation.

Shimshon Berkovits et al. [84] have proposed a model for mobile agent authentication using trust based mechanisms. The model checks authenticity during transmission of the mobile agent over its itinerary. Model uses the reference monitor that decides about grant of the request, operation of the request and access rules also.

Weidong Fang et al. [85] have worked on the model of reputation evaluation system for wireless sensor nodes. This model makes use of trust calculation based on previous communication of the node. Trust calculation can be either direct or indirect. Direct trust is calculated based on the trust distributed across the network. After this, trust value from the adjacent nodes is calculated and used for indirect trust calculation.

Govind P. Gupta et al. [86] have proposed energy and trust aware mobile agent migration (ETMAM) protocol for trust and energy calculation. Proposed model adopted data aggregation model for computation of trust for mobile agent. In the proposed model, comprehensive trust is calculated based on direct trust and aggregated trust value from the neighbouring nodes. Every node in this model is equipped with Trust Manager Component (TMC). TMC calculates the trust value of neighbours based on some predefined values. After that comprehensive trust is calculated based on these values. This model leaves nodes from the mobile agent itinerary if the node trust calculation is not up to the mark.

Napa Sae-Bae et al. [87] have proposed a mechanism of online signature verification for mobile devices. In this approach, online signatures are represented using a histogram. Histograms are designed specially to capture essential details of the signature as well as some of the relationships between multiple attributes of the signature. This approach can also be used in mobile agent authentication process.

Guenther Starnberger et al. [88] have proposed a Quick Response Transaction Authentication Numbers (QR-TAN) for mobile transaction authentication. This technique uses two dimensional QR barcodes. The advantage of this method is that terminals do not require any up gradation from the current state. Colour barcodes can be used for further enhancement of the technique.

G. Geetha et al. [89] have proposed a new method for trust and reputation management in terms of mobile agent security. Trust value of each host is calculated and stored in the routing table along with other parameters. This trust value is updated over the period of time and the path for the mobile agent is selected based on the best value of the routing table.

Bhavin Shah et al. [90] have proposed neural network based technique i.e. neural network based back propagation model for intrusion detection or malicious node detection in the network. This paper also compares various techniques of intrusion detection and focuses on reducing size of the mobile agent to be transmitted over the network. Paper suggests mobile agents for data communication in client server architecture for data transfer.

Dilli Prasad Sharma et. al. [91] have proposed concept of authentication of the mobile agent for distributed environment. In this model, whenever a system wants to authenticate a particular node, the system proxy calls the mobile agent. Mobile agent is equipped with necessary authentication information. Model uses the signature concept to sign and verify the signature.

Mianxiong Dong et al. [92] have proposed a mobile agent based model for energy and time efficient data collection. Model uses dynamic route selection of mobile agent itinerary based on greedy approach. The dynamic route selection helps in planning the mobile agent itinerary in a much better way. Using this mechanism, data can be collected from the distributed nodes in very less time.

Michael Riecker et al. [93] have proposed the lightweight and energy efficient model for authentication of the node. The model makes use of consumption of energy as the vital element in determining the node to be malicious. Node which is consuming abnormal energy is detected using mobile agents carrying necessary information across the network. Deviations from the normal consumption of the energy must be strong enough that it should be detectable.

In the past, various schemes have been proposed for authentication using unique signature generated from different parameters of the device. Table 2.3 shows the proposed mechanisms with their advantages and disadvantages.

Table 2.3: Comparison of various schemes of Digital Signature Based Authentication

Sr. No.	**Model Name**	**Methodology**	**Advantages**	**Disadvantages**
1.	Lampson [94]	• Certificate based	• A single certificate authority is used.	• Complexity is higher. • Requires new

			• Certificates are refreshed after some interval of time. • Restrictions of power distribution can be applied to both the parties • Arguments can be passed remotely using **R**emote **P**rocedure **C**all (RPC).	libraries and principals to be installed.
2.	BTRES: Beta-based Trust and Reputation Evaluation System for wireless sensor networks [95]	• Prediction of Trust	• Prediction of node trust is calculated based on the communication between communicating nodes. • Trust value of the node is also calculated based on neighbour's reply of the trust value of the node.	• Neighbour nodes can send incorrect value if hijacked or may not reply at all. • One unintentional wrong transaction can lead to low trust value.
3.	Signature verification on mobile devices [96]	• Histogram and signature mechanism	• Histogram from signature sample is created. • Features are extracted from the histogram and these features are used by the matcher to verify the signature. • Lightweight protocol.	• Cloned signatures will be verified and cannot be prevented.

4.	QR-TAN: Secure Mobile Transaction Authentication [97]	• QR Codes	• Transaction signing mechanism using QR codes is used.	• Additional requirement for QR code generation and reading • Higher resolution camera and computational power of the device.
5.	Trust and Reputation management for Mobile Agent [98]	• Trust value	• Reputation value of a particular node is calculated from number of nodes connected in the network. • Lightweight protocol.	• Forged node can send wrong trust value.
6.	Improving Performance of Mobile Agent Based Intrusion Detection System [99]	• Neural Network	• Backpropagation Neural Network Technique (BPNN) is used for intrusion detection. • Zero packet dropping rate to improve the efficiency.	• Training of Neural Network takes long time. • Network parameters and node parameters can change after some time. So re-training may be required for BPNN.
7.	Mobile Agent-Based Authentication [100]	• Digital Signature	• Key pair is generated. • Signature algorithm is used to sign and verify the message.	• Vulnerable to attacks if keys are compromised.

In distributed computing, authentication and data processing at various nodes are some of the problems. In this type of computing, a bigger task is splitted and distributed among multiple machines in the network. Then results are combined from distribute machines and processed depending upon the problem. As data set is increasing day by day for the organizations, the term used to refer these types of data sets is known as BigData [101].

2.7 BIGDATA

BigData is considered to be a data collection that has grown so large that it cannot be effectively or affordably managed (or exploited) using conventional data management tools: e.g., classic **R**elational **D**atabase **M**anagement **S**ystems (RDBMS) [102] or conventional search engines. In the wireless domain, processing of BigData securely on distributed machines is also required due to broadcast nature of these networks. There are four levels of security in terms of BigData [103]:

Authentication: This is first level of security i.e. authentication. There are multiple methods of authentication are available in terms of username, password combination or biometric information based authentication mechanism.

Access (Authorization): This type of security mechanism describes the type of access to the available resources. In terms of a file, the type of access can be either read, write or execute.

Visibility: This type of security mechanism describes about the quantity and type of information visibility to different type of users.

Data Security (Encryption): This type of security describes the data storage in encrypted format, data transmission with encryption etc.

BigData processing uses MapReduce technique [104] for distributed data processing. This technique was developed by Google. Hadoop, which is a BigData processing tool, also follows the MapReduce technique for straggler detection.

Since a job in MapReduce framework does not complete until all map and reduce undertakings are done, even few stragglers can deteriorate the overall execution time for job. Stragglers are the nodes which are taking longer time than usual in completion of distributed task. Straggler problem has been taken seriously and numerous stragglers moderating methods have been created [105-108]. As processing is distributed so authentication of participating nodes becomes even more important. A malicious node can become a straggler or can produce the wrong information.

There are multiple attack scenarios [109] which are possible and can affect the time taken by a job to complete. Some of these are:

Communication Denial: In this kind of attack the attacker can either switch off the machine or make the machine to run repetitive tasks so that it is not able to respond [110]. Although, Hadoop has the mechanism to detect unresponsive node and replicating the task to some other machine. For detection of unresponsive node, node alive messages are used. But attacker can overcome this type of defence by responding only to node alive messages.

Communication Delay: In this kind of attack node induces delay in communication in terms of response given to the master node. This can be done by delaying communication of the TCP packets. Delay in TCP packets leads to increased number of timeouts [110] [111]. As the communication time increases, master node has to wait for the results to be given by the compromised node [112]. Although internal messages like node alive message can still be communicated but this unnecessary delay can make this attack almost similar to communication denial attack [113] [114].

Denial of Service (DoS) attack: By sending a lot of unnecessary messages to target machine, an attacker can make machine's input or output buffer full. In case of TCP connections, SYN flood [115] is the most common type of attack which leads to denial of service. SYN messages are used for connection establishment in TCP which is a three way handshake process. After SYN is received by the machine, it replies with SYN-ACK message. After completing the connection machine also responds with the final ACK message. So, in this type of attack, attacker floods the multiple SYN messages to the target machine which leads to input or output buffer overflow situation.

Data Corruption: As data processing is distributed in case of Hadoop, different blocks of files are assigned to multiple slave machines by the master. A slave node can store multiple blocks of a file to be processed. If name of the blocks is interchanged, then data or image can be corrupted. Hadoop has no mechanism to check contents or interchanged name of the block [116]. So, using this type of attack, output of MapReduce job can be manipulated.

All these attacks can make the target machine a straggler. Once a machine becomes straggler, it is very important to identify the straggler and to reassign the task to a different machine for its completion. There are multiple methods of straggler detection, some of these are:

HADOOP native scheduler: Multiple splitted tasks are distributed on different machines using MapReduce and task score is calculated on a regular basis and according to that a speculative task is identified and rescheduled.

Longest Approximate Time to End (LATE) scheduler [117]: LATE comes up with an alternate system that approximates the completion time for same type of tasks to predict foresees potential stragglers. Based on the time to finish LATE reschedules the task if beneficial.

MANTRI [118]: The root of MANTRI's advantage lies in integrating static knowledge of job structure and dynamically available progress reports into a unified framework that identifies outliers early, applies cause-specific mitigation and does so only if benefit is higher than the cost. Based on task progress reports, MANTRI estimates for each task the remaining time to finish, i.e. t_{rem}, and the predicted completion time of a new copy of the task, t_{new}. Tasks report progress once every 10s or ten times in their lifetime, whichever is smaller.

Apart from the techniques mentioned above, in the recent past, there are various researchers who have worked on the issue of straggler detection, prevention or impact of attacks like blocking of data, inducing the unnecessary delay into data transmission and distributed denial of service attack (DDoS). Some of these are:

William Glenn et al. [109] have shown the impact of multiple types of attacks on the Hadoop cluster. Authors have evaluated attacks like input ports blocked, packet delay and SYN flood attack. Using the MapReduce configuration, total time taken by the job to complete increases manifolds if above mentioned attacks are conducted.

Sufian Hameed et al. [119] have proposed a new framework for DDoS attack detection. Authors have used various phases for DDoS detection. In the first phase, traffic is captured which is being communicated into the network. After the traffic is captured, log is generated in the second phase. In the third phase, this generated log file is used for detection of DDoS attack in to the network. Solution solves complexity problem in attack detection as it uses parallel data processing feature of Hadoop. Analysis of the algorithm shows that around 75% of the time for detection of DDoS is spent in traffic capturing phase.

Yeonhee Lee et al. [120] have summarized the use of counter based method for DDoS detection. In this method algorithm counts the number of requests and total traffic from the slave nodes to the master node. If the count is greater than a threshold value then slave node is behaving in a suspicious manner. Another method described by the author is access pattern based method. This method assumes that access pattern of the two infected nodes will be same. This method calculates the time spent on the page and number of bytes transmitted. Main drawback of this scheme is that computation time is higher for detection of DDoS attack.

Yoon-Su Jeong et al. [121] have worked on the weakness of Hadoop native file management (HDFS) security. According to authors, HDFS does not provide a robust security for authentication of users to access the file system of Hadoop. So, HDFS is susceptible to attacks like replay and impersonation. A new token based authentication mechanism is proposed which allows the authentication of nodes using block access token. This scheme, which works on the client server architecture, tokens are generated using the Elliptic curve based cryptography. Proposed scheme provides a good defence against replay and impersonation attack.

Durbadal Chattaraj et al. [122] have proposed a new authentication protocol i.e. Hadoop Efficient Authentication Protocol (HEAP). This protocol is based on two server models for authentication. HEAP uses digital signature standard and verification process using elliptic curve cryptography. HEAP also protects various well known attacks on Hadoop platform. Author has also given the detailed evaluation of the protocol which shows that computation overhead is reasonable as compared to the existing state of the art technologies. As the model uses the concept of two server architecture, it gives additional advantage of prevention against single point of failure (SOF) and single point of vulnerability (SOV).

Shen et al. [123] have used a different mechanism altogether i.e. Trusted Computing Platform (TPM). This mechanism protects the system from hardware attacks as well as software attacks. In this type of scheme the authentication information of the user is stored in the hardware. Authors have also developed a software module known as Trusted Platform Software Stack (TSS) with the help of this, TPM services can be easily used by the cloud computing applications. Although this mechanism is very secure and powerful mechanism against various attacks, it has some drawbacks. First drawback is the requirement of additional hardware and its compatibility with existing hardware. Second drawback is a new processing layer is introduced which increases the processing time to a certain extent.

If the node is authenticated during MapReduce it can also help in preventing attacks like Distributed Denial of Service attack (DDoS). A new framework for BigData processing and straggler detection is required which can also prevent attack like DDoS. Work discussed above follows traditional Client-Server (CS) paradigm. In this paradigm, server responds to multiple clients whenever request comes. In the next section, CS paradigm is explained in detail.

2.8 CLIENT SERVER PARADIGM

As shown in Figure 2.7, a server can serve multiple clients at any instance of time. Server provides different kind of services like railway ticket reservation, flight ticket bookings, database access [124] etc.

Although, Client-Server (CS) technology is very popular and is currently used by various service providers, have some drawbacks. Some of the issues related to CS technology are:

Centralized Management: In CS mode, all decisions are taken at the server level. So, as number of client increases, efficiency of the server decreases. Another issue with this approach is that if server fails then complete message exchange process stops.

Bandwidth: As number of clients and hierarchy increases in the network, traffic increases manifolds. Much of the bandwidth available between client and server will be wasted in carrying management traffic.

Scalability: As network grows and number of clients increases into network, load on server increases. It is very difficult to upgrade the capability of server to serve increasing number of requests from clients. As network grows and becomes more complex, capabilities are difficult to scale.

Response Time: As number of hop count between client and server increases, response time increases to a great extent in CS approach.

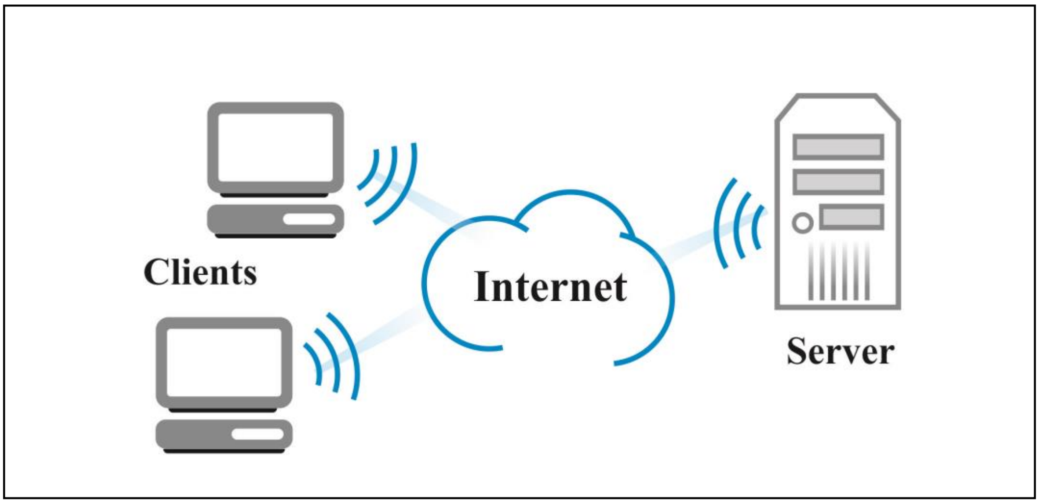


Figure 2.7 Client Server Paradigm

Fault Tolerance: Another drawback associated with CS approach is that its fault tolerance is almost zero. If server fails, due to any problem, then all services provided will standstill until server is available.

So, there are some serious drawbacks in client server architecture and in the recent years, traffic on the internet has grown to a great extent. Number of users on the internet is growing continuously. By 2021 there will be more than 635.8 million internet users in India only [125]. According to the Cisco VNI, 2017 globally IP traffic growth is tremendous as and is increasing day by day. The report shows that growth of IP traffic from 2016 to 2021 is around 24%.

As the traffic on the internet is growing at an unimaginable rate, there is a need for updating existing techniques which can also reduce the traffic on the internet. Mobile agent technology can be one of the solutions to reduce the traffic on the internet [126].

2.9 MOBILE AGENT

Mobile agent is an autonomous code which can roam inside the network by following multiple nodes on network which is called itinerary of the mobile agent [128]. This itinerary can be dynamic or static [129][130]. Mobile agent can roam in to network following itinerary. Mobile agent technology has seen a good amount of growth over last few years. In some scenarios, this technology can be used instead of client server architecture for reduction in traffic around the network.

2.9.1 MOBILE AGENT LIFECYCLE

Mobile agent follows a typical life cycle. A mobile agent can have multiple states during its lifecycle. Complete lifecycle is explained in Figure 2.8.

Initiate: This is the first state when mobile agent is being initiated.

Active: After initiation, mobile agent is in active state and becomes an autonomous code which can take decision of its own.

Waiting: Mobile agent can go in waiting state due to some internal or external factor.

Suspend: This is the state which represent that mobile agent is not executing currently.

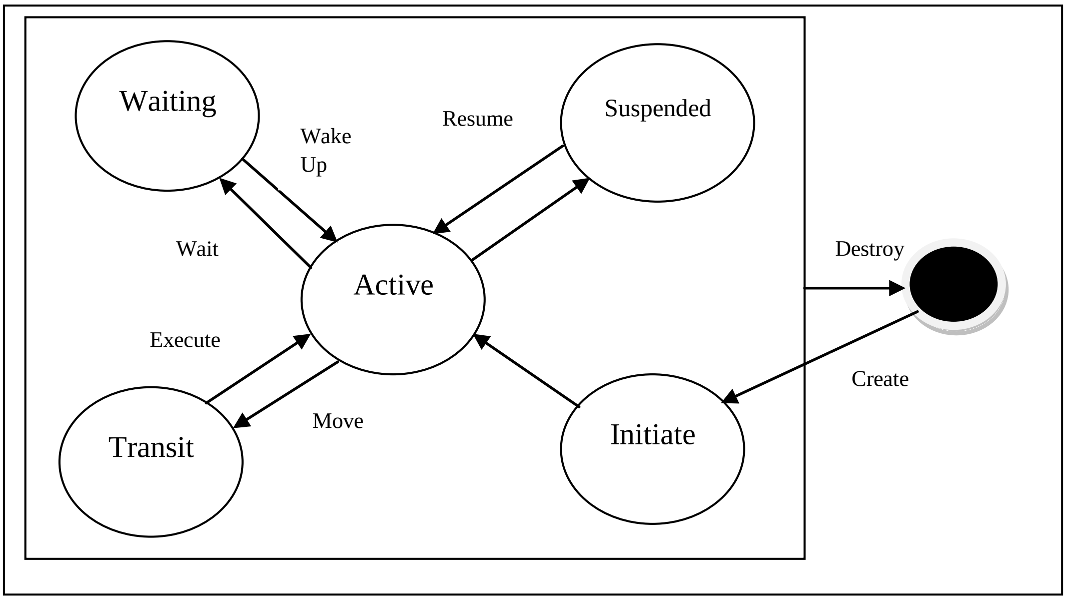


Figure 2.8 Mobile Agent Lifecycle

Transit: As mobile agent can move from one node to another during its execution so, it enters in transit state when it is in move from one node to another.

Deleted: Mobile agent once finishes its job can be destroyed

2.9.2 ADVANTAGES OF MOBILE AGENT

There are number of advantages of mobile agent as compared to client server model. Some of these are:

Efficient bandwidth utilization: Mobile agent requires less data to be transmitted as compared to original data. Mobile agent can also pre-process data and can transmit compressed information.

Asynchronous: Mobile agents are asynchronous in nature. Once mobile agents are downloaded, they can perform their task asynchronously without any interaction from the parent machine.

Heterogeneous network support: Mobile agent depends only upon the framework in which it is designed with no limitation on underlying architecture. So, mobile agents are separate from the environment in which they are running by the framework in which they are developed.

Increased availability of resources: Mobile agent helps in reduction of traffic so, there will be increase in availability of resources to clients.

2.10 SUMMARY

This chapter provides a brief overview about wireless security and various methods for providing wireless security. A large extent of work in the area of wireless authentication, key exchange, authentication and processing in distributed environment is already done using different mechanism. The main objective of this survey is to introduce and evaluate contributions of these work and reveal existing problems in the area of wireless security. From the survey, it is concluded that various techniques for handling authentication problem, key exchange, distributed environment data processing and straggler detection are provided but there are some issues in terms of reliance on third party certificates, generation of too much traffic, load on KDC, key exchange with authentication and straggler detection. This literature survey provides basic knowledge about the previous work and some drawbacks of the existing work.

CHAPTER 3

MOBILE AGENT BASED FRAMEWORK FOR WIRELESS AUTHENTICATION (MABFWA)

3.1 INTRODUCTION

Authentication is the most important aspect of security in wireless communication. Extensible Authentication Protocol (EAP) provides the framework for wireless authentication over which various other protocols like EAP TLS, EAP-TTLS, EAP MD5 etc. can be executed. These protocols use client server based mechanism during communication. These mechanisms can have problems in highly distributed systems like generation of too much traffic on the network, third party certificate reliance and server failure etc. So a new mechanism which can support all these protocols in a distributed environment is needed.

3.2 PROPOSED WORK

Proposed **M**obile **A**gent **B**ased **F**ramework for **W**ireless **A**uthentication (MABFWA) makes use of mobile agent for communication among various entities. Model assumes that all the entities participating in the authentication process has anyone of mobile agent (MA) execution environment like Java Agent Development Framework (JADE) or AGLETs installed. MABFWA separates the message exchange from the authentication process and provides the extensibility in terms of support for multiple authentication algorithms. It can support authentication algorithms like **P**rotected **E**xtensible **A**uthentication **P**rotocol (PEAP), **T**ransport **L**ayer **S**ecurity (TLS), **T**unnelled **T**ransport **L**ayer **S**ecurity (TTLS) and **F**lexible **A**uthentication via **S**ecure **T**unnelling (FAST) etc. In the next section, various entities participating in the communication are explained in detail.

3.2.1 KEY AUTHENTICATION ENTITIES AND THEIR INTERACTION MODEL

In general, MABFWA contains three entities: Client/Supplicant, Authenticator and Authentication Server. The layered structure of the proposed work is shown in Figure 3.1.

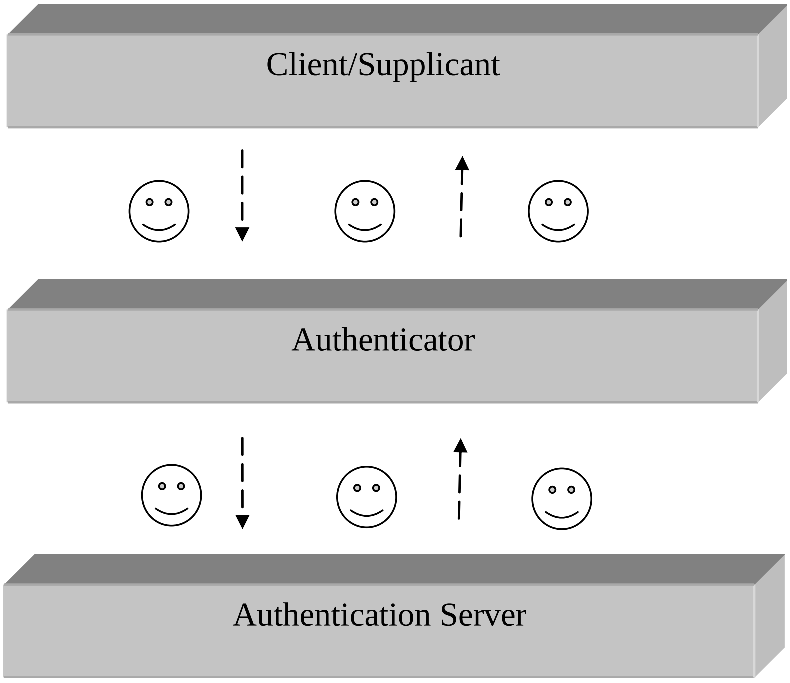


Figure 3.1 Interaction among layers

Client/Supplicant: Client/Supplicant can be a remote machine or mobile device which needs to be authenticated for a particular service. Client supplies all necessary credentials for authentication process to the mobile agent forwarded by the authenticator. Client dispatches the mobile agent which is loaded with authentication data to the authenticator.

Authenticator: Authenticator creates mobile agent which is dispatched to client/supplicant. Authenticator receives the mobile agent carrying credential information from client/supplicant. Authenticator also communicates with the Authentication server.

Authentication Server: Authentication **S**erver (**AS**) is a third party server which verifies all credential information of client/supplicant. It receives mobile agent from the authenticator and verifies credentials based on the authentication algorithm used.

After verification, Authentication Server sends success or failure message to the authenticator via mobile agent.

3.2.2 LAYERED AUTHENTICATION FRAMEWORK

If an entity wants to access the network or resources associated with the network, first of all entity needs to be authenticated. The entity can be a device and/or a user which needs to be authenticated. Authorization process depends upon the type of algorithm used for the authentication.

Complete authentication process is explained in the Figure 3.2. The Supplicant accesses the network with the help of authenticator port and this access can be controlled at a single point.

3.2.3 MOBILE AGENT BASED MESSAGE EXCHANGE FRAMEWORK

The proposed system, as shown in Figure 3.3, shows the mobile agent based communication among three entities i.e. Supplicant (S), Authenticator (A) and Authentication Server (AS). The first exchange could be an identity exchange.

Step 1 is optional in which identity request and response messages are exchanged. In Step 2 all exchanges between Supplicant and Authenticator is done with the help of Mobile Agent (MA). Message exchange depends upon the type of authentication algorithm deployed for authentication process. The Step 3 shows message exchange between Authenticator and Authentication Server. In final step success or failure message is communicated to the Supplicant or Client. Framework shows the flow of message exchanges among three entities.

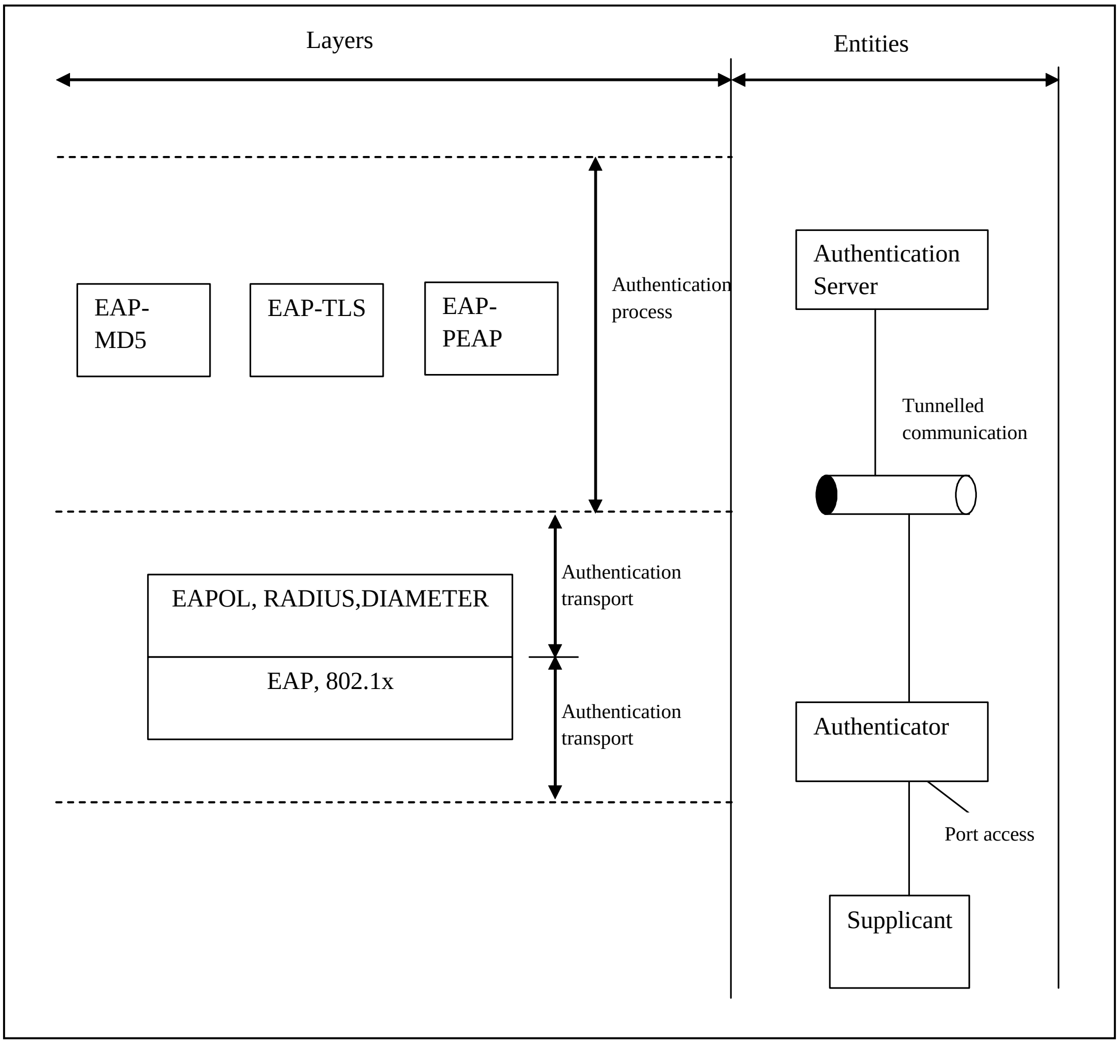


Figure 3.2 Layered authentication framework

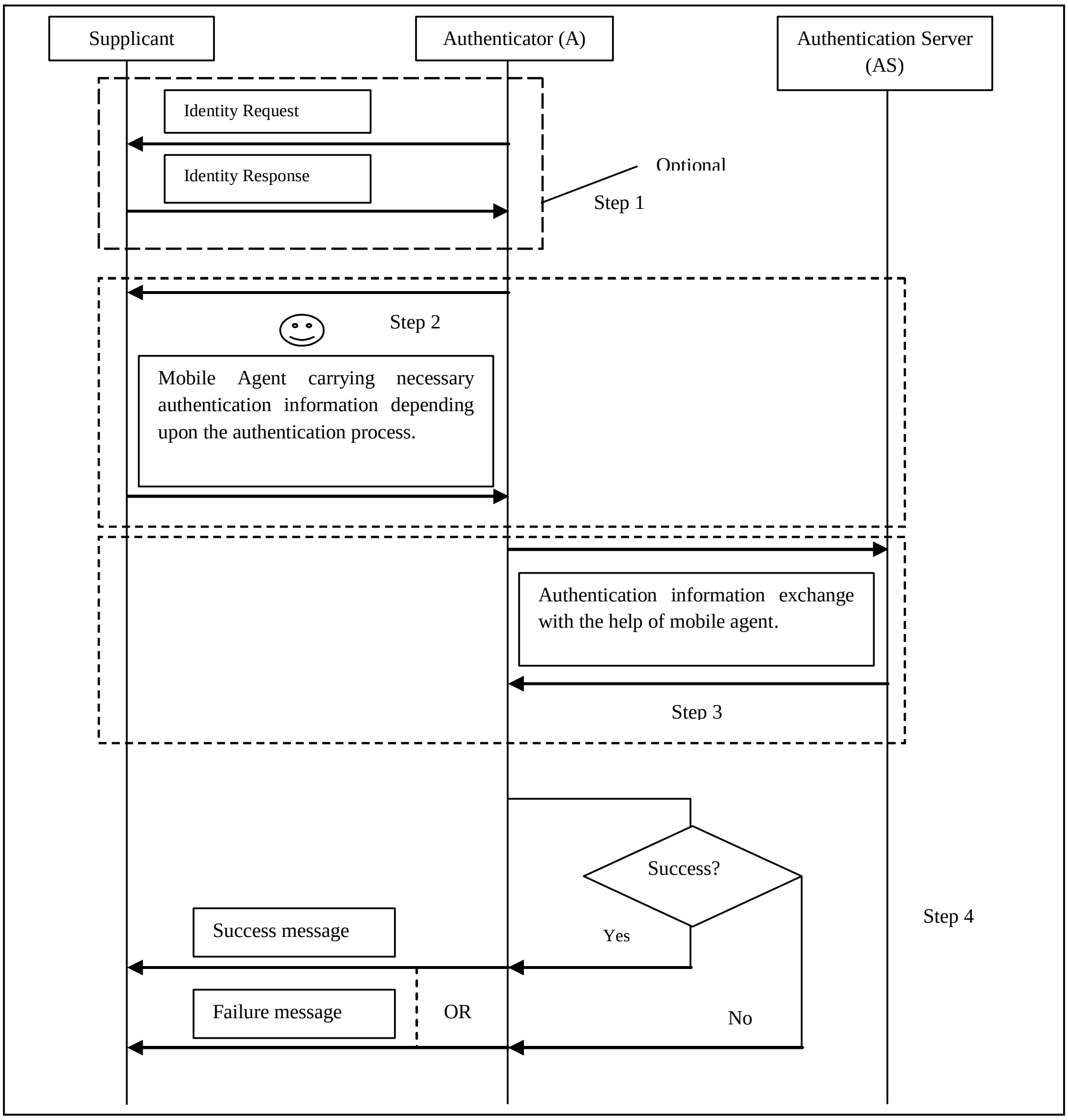

Supplicant
Authenticator (A)
Authentication Server (AS)
Identity Request
Identity Response
Optional
Step 1
Step 2
Mobile Agent carrying necessary authentication information depending upon the authentication process.
Authentication information exchange with the help of mobile agent.
Step 3
Success?
Step 4
Yes
Success message
Failure message
OR
No

Figure 3.3 MA based Message Exchange Framework

Algorithm 3.1 shows the complete process of wireless authentication. Whenever a client comes in to the range of the authenticator, **credential_Collector( )** agent is created and dispatched to the client. Mobile Agent upon reaching the client collects the necessary authentication information and returns back to the authenticator.

```
Algorithm: Wireless_Authentication

Input:

1. Necessary information for authentication process e.g. username, password or device fingerprint
2. Client addresses i.e. client(i)
3. Authenticator address i.e. authenticator_address
4. Server address

Output:

Success or failure of authentication

1. Repeat for every client stored in client(i) who comes in the range of authenticator

2. while (client(i) != Null)

        2.1 Create and dispatch Mobile Agent credential_Collector( client(i) ) from authenticator to client.

        2.2 Mobile Agent credential_Collector( ) upon reaching the client collects the necessary credential either username, password or device fingerprint.

        2.3 Mobile Agent credential_Collector( ) loaded with authentication data is dispatched to the authenticator.

3. Authenticator creates a Mobile Agent credential_Verifier( ), loads it with necessary   authentication information from credential_Collector( ).

        3.1 dispatch credential_Verifier( )  to the Authentication Server.

4. Authentication Server sends success or failure message to the credential_Verifier( ) based on the authentication information.

5. dispatch credential_Verifier( ) mobile agent to authenticator.
```

Algorithm 3.1: Algorithm for Wireless Authentication

Authenticator creates a mobile agent **credential_Verifier( )**, loads it with credentials from the **credential_Collector( )** agent and dispatches it to the third party server. Server sends the success or failure message to the client.

Algorithm 3.2 shows the method for bulk data transmission. Load_Data agent is created and is loaded with necessary information to be transmitted either from client to server or from server to client. Necessary information can be encrypted using the session key.

```
Algorithm: Data_Transmission
Input:
1. Data to be transmitted
2. Destination address (server or client)
3. session_key
Output:
Success or failure of data transmission
1. Repeat while client or server has data to send
        1.1 create agent Load_Data( ).
        1.2 load agent with the requisite data and session_key.
2. dispatch agent to the destination.
```

Algorithm 3.2: Algorithm for Data Transmission

Thus using above flexible framework, Mobile Agents could be generated for authentication process according to the algorithm selected for authentication process. This flexible framework can support multiple authentication algorithms like MD5, PEAP and TLS etc. The framework also has advantage of dynamically changing the authentication process without changing the underlying architecture so much. This hybrid approach has two advantages:

- This approach can dynamically adapt new authentication algorithms without so much of changes in the existing framework.
- The network level problems of Client Server architecture like scalability, more message exchanges and resource requirement in terms of bandwidth can be avoided.

The itinerary for mobile agent for authentication process depends upon the network topology and scheme adopted for authentication. Proposed mobile agent based framework for authentication can support multiple type of authentication mechanism some of these are: MD5, TLS and PEAP etc. In the next section, numerical analysis of the client server approach with the proposed mobile agent based approach will be discussed.

3.3 NUMERICAL ANALYSIS

In this section numerical analysis in terms of traffic generated around the authenticator and remote interaction time is done. Traditional EAP framework works on the client server based architecture. In this architecture multiple clients communicate to the server in parallel and server responds to the clients. Numerical analysis and comparison is done for traditional client server based approach and proposed mobile agent approach.

Client-Server Approach: In Client-Server approach, the Authenticator receives traffic from multiple clients seeking authentication within the network. The approach can be understood with the help of Figure 3.4. Multiple clients connected to the server send the request message and server responds with the response message.

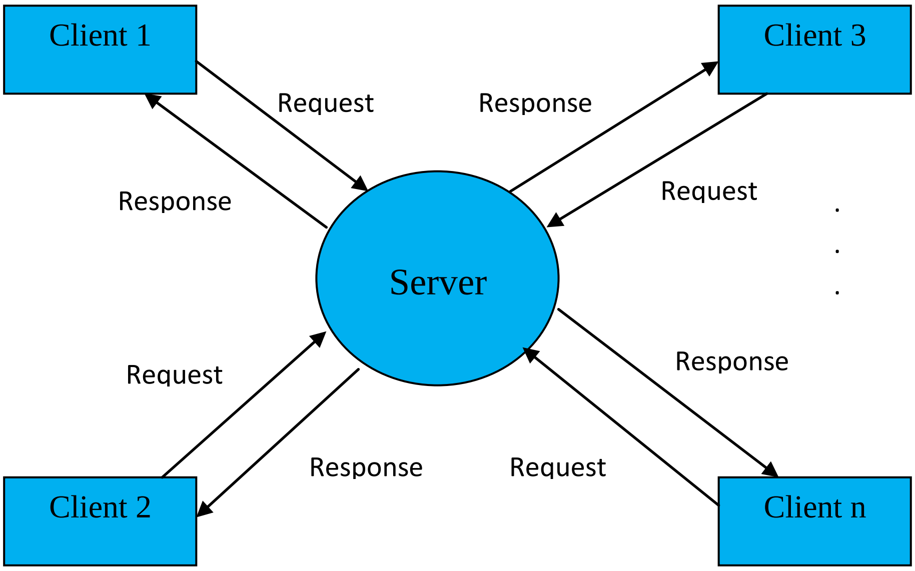


Figure 3.4 Client-Server architecture

In traditional EAP approach authenticator serves multiple clients and authentication server serves multiple authenticators. Equation 3.1 can be used to calculate the complete traffic around authenticator within the network:

$$TrC_{cs}^{m} = \sum_{i=1}^{n} \begin{Bmatrix} (Sreq + Sres) * x * Avg.no.of\ sessions + \\ (Sreq + Sres) * y \end{Bmatrix} \quad \dots (3.1)$$

where, $Sreq$ is the size of request packet from client to the server and

$Sres$ is the size of response packet from the server to client.

x is number of message exchanges between client and authenticator depending upon authentication protocol.

n is number of clients.

y is number of calls to authentication server depending upon protocol.

Remote interaction time is the time required by the authenticator to validate clients over the network. It will depend upon the bandwidth available and will be calculated as in equation 3.2.

$$TmC_{cs}^{r} = \sum_{i=1}^{n} \frac{(Sreq + Sres)}{Bw_i} + 2Lt_i \qquad \dots (3.2)$$

where, TmC_{cs}^{r} is the remote interaction time for one message exchange with n number of clients in the client server architecture.

Lt_i is the latency time between authenticator and ith client.

Mobile Agent (MA) based approach: Management cost in terms of network traffic generated at the authenticator i.e. TrC_{ma}^{m} will be calculated as follows in equation 3.3.

$$TrC_{ma}^{m} = \{Sma + \sum_{i=1}^{n} Spr\} \qquad \dots (3.3)$$

where, Sma is the size of mobile agent carrying the authentication algorithm code to be executed.

Spr is the partial result generated by each client.

So, according to the above equation the amount of traffic generated at the authenticator depends on the size of the mobile agent.

Remote interaction time is the time required by the authenticator to validate the clients over the network will depend upon the bandwidth available and will be calculated using equation 3.4 for MA:

$$TmC_{ma}^{r} = \sum_{i=1}^{n} \frac{(Sma + Spr)}{Bw(i-1,i)} + Lt(i-1,i) \qquad \dots (3.4)$$

where, $Lt(i-1,i)$ is the latency time between i-1 and ith node.

3.4 Comparison of Client Server (CS) Model and Mobile Agent (MA) Based Model

In this section the comparison of CS approach and MA approach is being done and performance comparison results of both the approaches are being shown in Table 3.1. Management cost and remote interaction time are taken as performance parameters.

Table 3.1: Comparison of CS and MA based model

Performance Matrix	**Client Server Model**	**Mobile agent based model**
$\boldsymbol{TrC^{m}}$ **(management cost in terms of network traffic around authenticator)**	Directly proportional to number of clients and number of message exchanges.	Proportional to the size of any information collected.
$\boldsymbol{TmC^{r}}$ **(remote interaction time)**	Directly proportional to number of messages exchanged by number of clients	As interaction is local between MA and client, it does not increase with increase in number of clients.

3.4.1 Network Traffic Related Performance

In this section both client server and mobile agent based approach is compared in terms of traffic generated. *Sreq* is the size of request packet from client to the server and *Sres* is the size of response packet from the server to client.

Management cost computation in terms of authentication can be done in CS approach for one particular node is as follows.

Typical *Sreq* size for client server architecture is around 50 Bytes.

Sreq = 50 Bytes

Sma (MA size) is 3 KB= 1024*3 = 3072 Bytes

Table 3.2 summarizes the various parameters used in numerical analysis.

Table 3.2: Parameters used

Parameter Name	**Description**	**Value**
Sreq	Request size from client to authentication server	50 Bytes
$\boldsymbol{Sma}$	Size of the mobile agent	3 KB
$\boldsymbol{\alpha}$	Multiplication factor to the size of mobile agent in case of client server architecture traffic calculation	Example values: 6 or 29

As traffic in CS mode is more as compared to MA mode, it is assumed that the traffic in CS mode is α times the size of mobile agent i.e. *Sma*.

$$(Sreq + Sres) = 50 + \alpha * Sma \qquad \dots (3.5)$$

Putting these parameters in equation 3.5

Case A: Taking α = 6,

50+6*3072 =18482 Bytes

Case B: Taking α = 29,

50+29*3072 = 89138 Bytes

Putting these parameters in equation 3.3, the management cost at authenticator i.e. TrC_{ma}^{m} in MA approach can be calculated as

= (3072+200)

= 3272 Bytes

It can be analysed from the Table 3.3 that the traffic in the CS based approach increases many times as compared to MA based approach and as the number of nodes increases traffic increases many folds around authenticator.

Table 3.3: Traffic around authenticator in MA and CS based model

No. of Nodes	**MA**	**CS α = 6**	**CS α = 29**
1	3272	18482	89138
5	4072	92410	445690
10	5072	184820	891380
20	7072	369640	1782760
50	13072	924100	4456900

The results can also be analysed by the graph as shown in Figure 3.5. Graph shows the results for MA, CS (α=6) and CS (α=29) calculations. For MA based approach the traffic is almost constant as the number of users increases. The proposed MA based approach helps greatly in reducing the traffic around the authenticator.

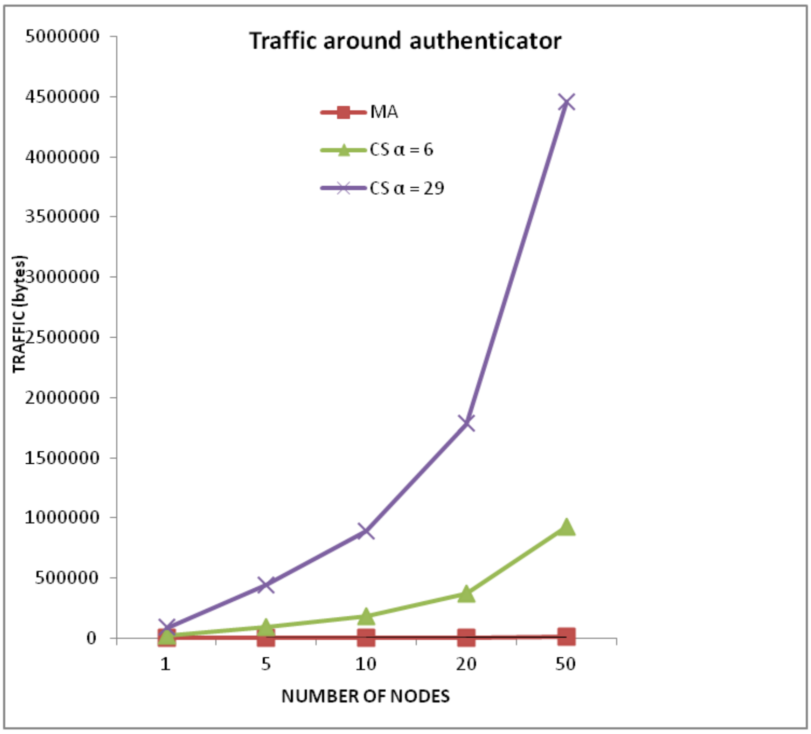


Figure 3.5: Traffic analysis for CS vs MA approach

3.5 Performance parameters in Mobile Agent based authentication and client server paradigm:

Following parameters have been identified for comparison of traditional client server approach with the Mobile Agent based approach.

- ***Number of message exchange among the entities:*** During the authentication process the number of messages exchanged between different entities like Client (C), Authenticator (A) and Authentication Server (AS).
- ***Total traffic generated during authentication process:*** During the authentication process total traffic generated around Client (C), Authenticator (A) and Authentication Server (AS).
- ***Time taken for authentication process:*** This is the total time taken for a single authentication process to complete.

These parameters can be used for performance analysis of Mobile Agent and client server based paradigm.

3.6 Example Algorithms

There are multiple authentication algorithms currently available. In this section some example algorithms like TLS, MD5 and PEAP has been taken and their mobile agent based implementation is done on the MABFWA platform. Message exchange of each algorithm on MABFWA platform is compared in terms of number of message exchanges with the conventional client server architecture.

3.6.1 TLS Message Exchange

TLS choreography as shown in Figure 3.6 shows how the messages will be exchanged between different entities. As soon as the client comes in to the range of authenticator, identity request and response messages are exchanged. Firstly client sends the hello message. Authenticator creates a mobile agent, loads it with the server hello message, certificate, server key, server request message and server hello done message. Server dispatches it to the client. After this, client derives the session key. Then client creates a mobile agent loads it with certificate, client key, certificate verify and change cipher specification values. Client then dispatches it to the server. Server also derives a session key. Server sends a message ‘finished’ to client about change cipher specifications. Now the client and server can exchange bulk data transmission between each other. Complete algorithm for the same is shown in Algorithm 3.3. TLS in the MABFWA framework really helps in reducing the number of message exchanges around the authenticator.

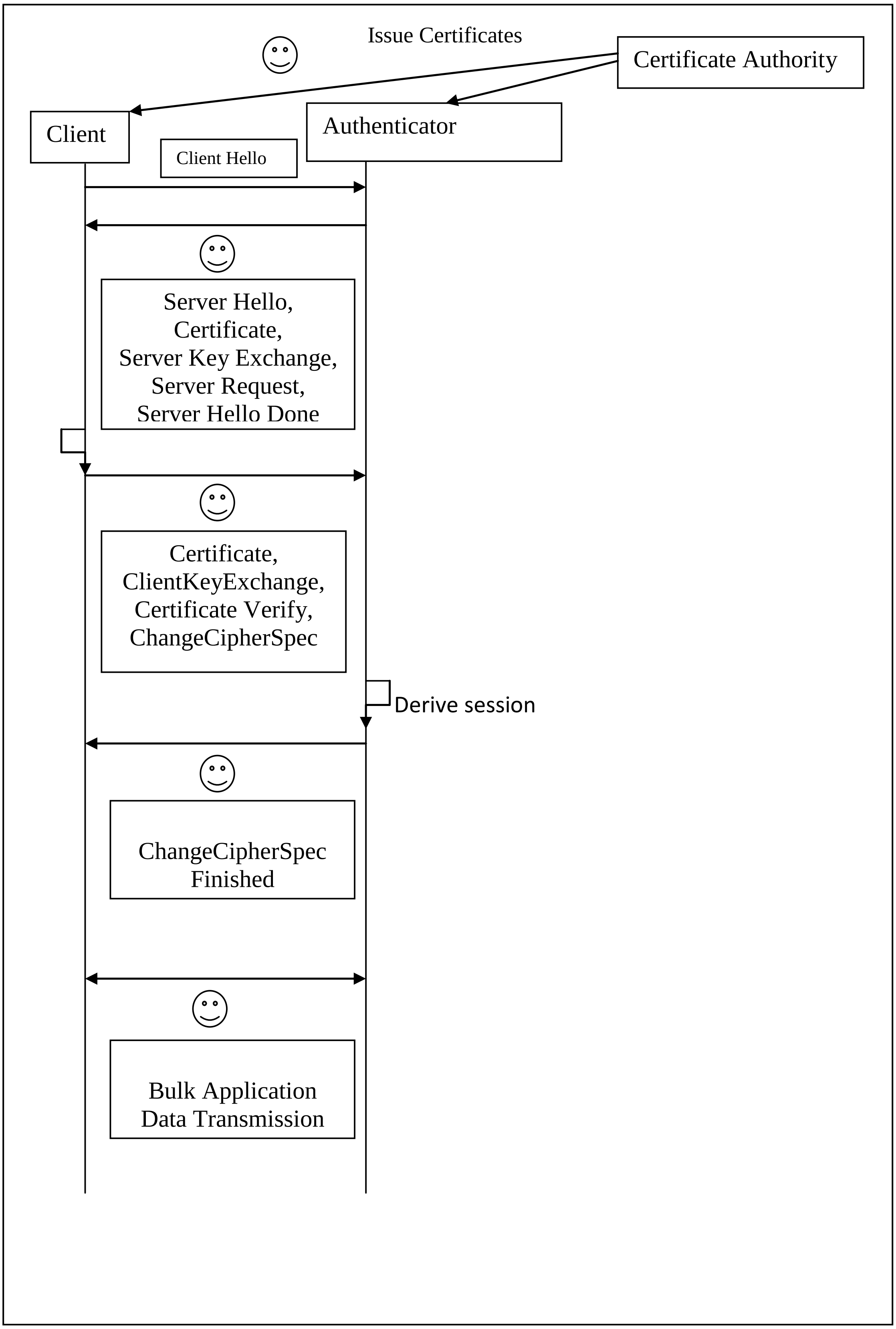


Figure 3.6: Message Exchange of TLS algorithm

```
Algorithm: Wireless_Authentication_TLS

Input:

1. Necessary information for authentication process e.g. username, password, device fingerprint or valid certificate

2. Client addresses i.e. client(i)

3. Server address

Output: Success or failure of authentication

1. Repeat for every client stored in client [ ] who comes in the range of authenticator

2. Repeat  for every client (i)   steps 1 to 6 while (client (i) != Null)

        2.1 Send Client_hello message to server.

        2.2 Server creates the mobile agent, loads it with the following information

                a. server hello
                b. certificate
                c. server key
                d. server request
                e. server hello done

        2.3 Server dispatches the mobile agent to the client (i).

3. Client (i) derives the session key.

4. Client (i) creates the mobile agent, loads it with the following information:

                a. certificate
                b. client key
                c. certificate verify
                d. change cipher specification

5. Client (i) dispatches the mobile agent to the server.

6. Server derives the session key.

7. Server responds with the changed cipher specification.

8. Bulk application data transmission continues.
```

Algorithm 3.3: Wireless Authentication TLS

3.6.2 MD5 Message Exchange

MD5 is another authentication protocol which requires a shared secret between client and server. Shared secret is usually a password which is used for authentication of the client. In MD5 method a random challenge is given to the client and client responds with the hash of the challenge. Shared secret is used for creating hash of the challenge. Authentication server recalculates the hash using the shared secret and responds with the success or failure of authentication depending upon the matched recalculated hash. Figure 3.7 shows how messages will be exchanged between the different entities.

As soon as the supplicant comes in the range of authenticator, authenticator demands the identity from the supplicant. Supplicant creates a mobile agent, loads it with username and password; and dispatches it to the authenticator. Authenticator forwards it to the Authentication Server. After reaching Authentication Server, it gives a challenge code (to be solved by authentication algorithm) to mobile agent. Mobile Agent calculates the challenge hash and gives it to the authentication server. Authentication server verifies the response hash and gives success or failure message to the authenticator accordingly. Authenticator forwards the message to the supplicant. Bulk data transmission can be done using mobile agent after successful authentication. Algorithm 3.4 shows the detailed algorithm.

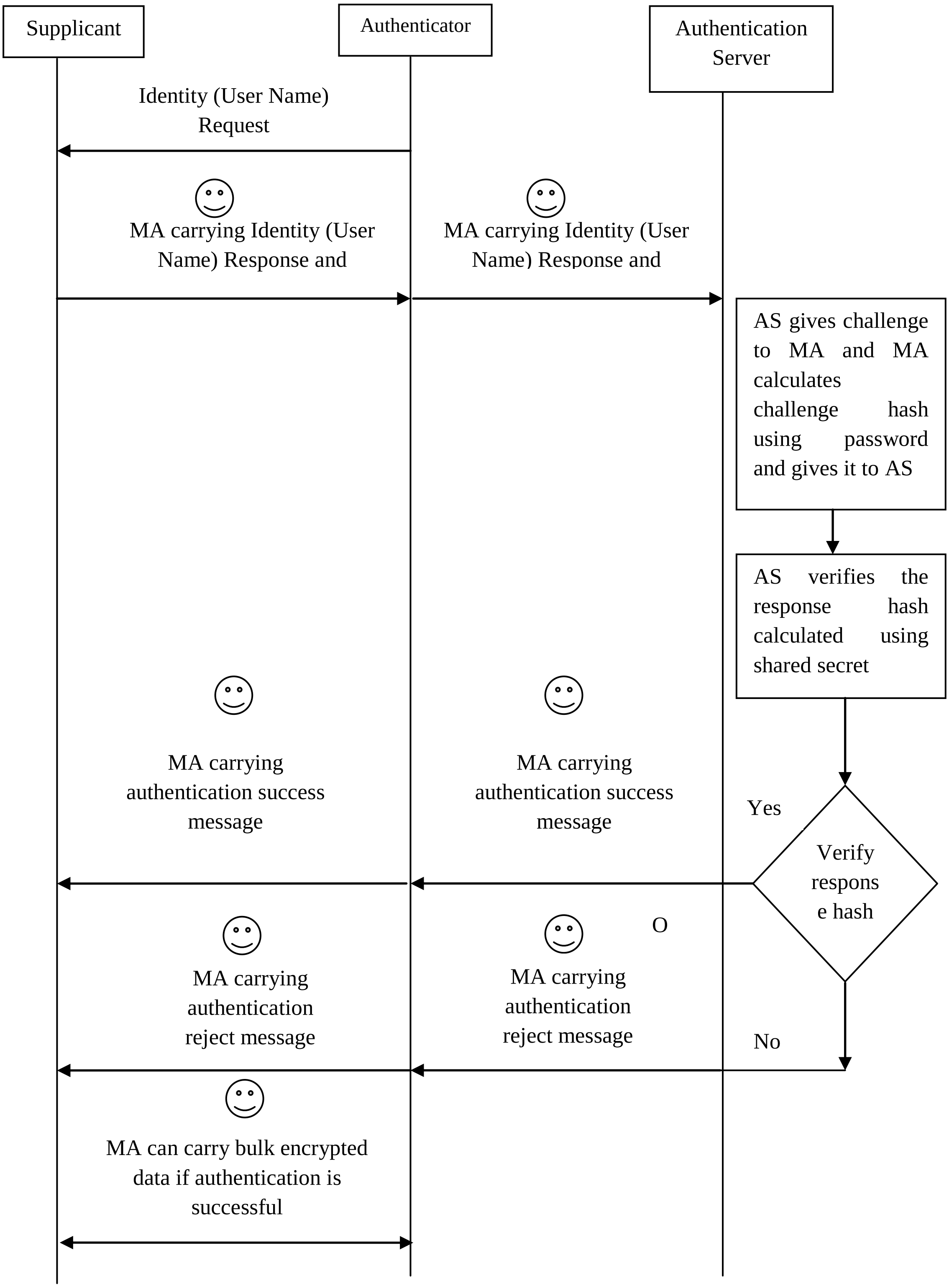

Supplicant
Authenticator
Authentication Server
Identity (User Name) Request
☺
MA carrying Identity (User Name) Response and
☺
MA carrying Identity (User Name) Response and
AS gives challenge to MA and MA calculates challenge hash using password and gives it to AS
AS verifies the response hash calculated using shared secret
☺
MA carrying authentication success message
☺
MA carrying authentication success message
Yes
Verify respons e hash
☺
MA carrying authentication reject message
☺
O
MA carrying authentication reject message
No
☺
MA can carry bulk encrypted data if authentication is successful

Figure 3.7: Message Exchange of MD5 algorithm

```
Algorithm: Wireless_Authentication_MD5

Input:

1. Necessary information for authentication process e.g. username, password or device fingerprint.

2. Client addresses i.e. client (i)

3. Server address

Output:

Success or failure of authentication

Algorithm is for every client (i) who comes in the range of authenticator

1. Repeat for every client (i)   steps 1 to 4 while (client (i) != Null)

        1.1 authenticator demand the identity request from the client (i).

        1.2 client (i) creates the mobile agent and loads it with the following information:

                        (a) identity response and

                        (b) password

        1.3 client (i) dispatches the mobile agent to the authenticator.

2. authenticator dispatches the mobile agent to authentication server.

3. authentication server gives challenge to mobile agent.

        3.1 mobile agent calculates challenge hash and gives it to the authentication server.

        3.2 authentication server verifies the challenge hash.

        3.3 if (verification= =success)

                send success message to authenticator.

        3.4 else

                send failure message to the authenticator.

4.  authenticator forwards the message to the client (i).
```

Algorithm 3.4: Wireless Authentication using MD5

3.6.3 Protected Extensible Authentication Protocol (PEAP) Message Exchange

PEAP is another authentication protocol which is somewhat similar to the TLS protocol. Figure 3.8 explains how messages will be exchanged between different entities. Most of the communication in PEAP is between Client and the Authentication Server. Authenticator acts as a pass-through.

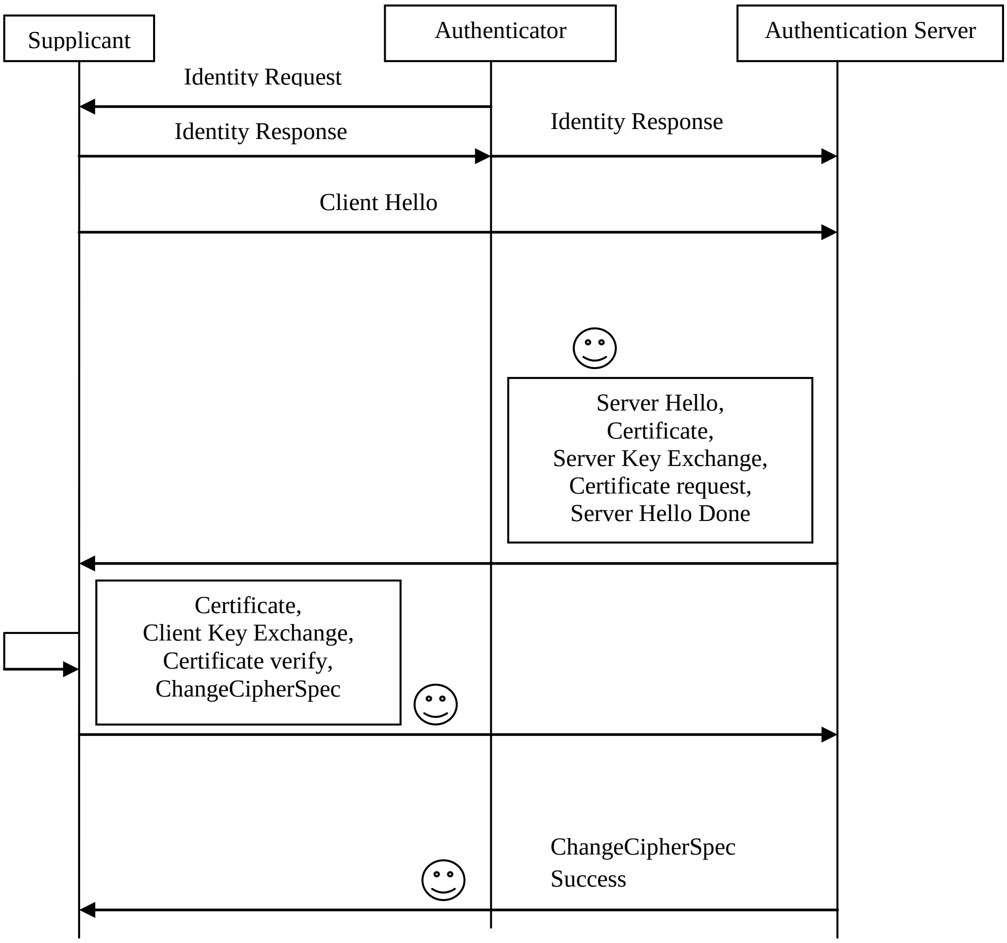


Figure 3.8: Message Exchange of PEAP algorithm

As soon as the client comes into the range of the authenticator client's identity is exchanged. After this, server sends the certificate, server key exchange, certificate request and server hello done message to the client as collective message with the help of mobile agent. Client in turn responds with the certificate, client key exchange, certificate verify and cipher specification collectively with the help of mobile agent. Server in turn responds with the cipher specifications and success message.

.

3.6.4 Comparison of MA approach to traditional EAP approach

In this section comparison of traditional client server based approach is done with the proposed mobile agent based approach in terms of number of message exchanges. Formula in equation 3.6 can be used for this purpose.

$$Percentage\ of\ Improvement = \left(\frac{No.of\ message\ exchange\ in\ MA\ based\ approach}{No.of\ message\ exchange\ in\ EAP\ approach}\right) * 100 \qquad \ldots (3.6)$$

Table 3.4 shows the comparison between the client server based approach and mobile agent based approach.

Table 3.4: Improvement using MABFWA

Sr. No.	Algorithm	Traditional Approach (EAP)	Mobile agent based approach (MABFWA)	Percentage of improvements
1	**MD5**	4+ data transmission	3+ data transmission	75
2	**TLS**	10+data transmission	3+data transmission	30
3	**PEAP**	7+ data transmission	3+data transmission	42.86

In MD5 algorithm there is around 4 number of message exchanges between the client and the authenticator but in MABFWA framework this is around 3 only. There is a 75 percent improvement in terms of number of message exchanges. In TLS algorithm there is around 10 number of message exchanges as compared to 3 only in proposed approach. This gives 30 percent improvement. Similarly in case of PEAP algorithm number of message exchange is around 3 which give the 42.86 percent improvement over traditional method.

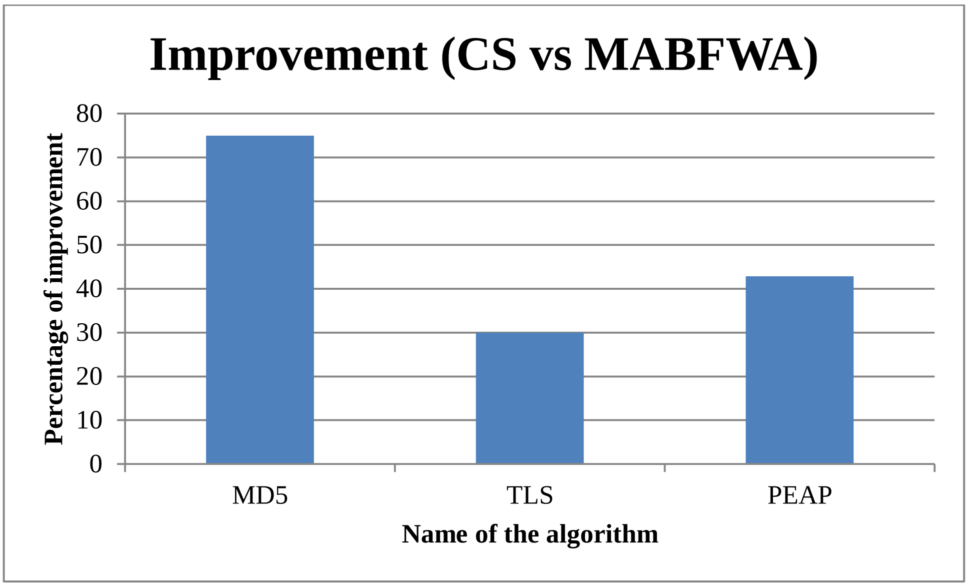


Figure 3.9 Percentage of improvement CS vs MABFWA

Graph in Figure 3.9 summarizes the operation and gives a comparison view among the traditional and proposed MABFWA based approach. There is a considerable amount of improvement in terms of number of message exchanged around the authenticator using the MABFWA based framework.

3.7 Conclusion

Proposed mobile agent based approach reduces the amount of traffic on the network to the great extent as the traffic just depends upon the size of the mobile agent to be transmitted between two entities. Furthermore the algorithm carried by the mobile agent for the authentication process can be dynamically changed without any change in the underlying architecture. So the proposed framework supports multiple authentication algorithms without any change in the underlying architecture. Algorithms like MD5, PEAP and TLS etc. can be incorporated in the proposed architecture and the results clearly show that there is a considerable amount of improvement in terms of message exchanged around the authenticator.

CHAPTER 4

KEY DISTRIBUTION THROUGH FINGERPRINT BASED AUTHENTICATION USING MOBILE AGENT (KDFBA)

4.1 INTRODUCTION

Secure communication on the network is one of the mandatory requirement for the users. A communication is secure, if it cannot be eavesdropped or intercepted. Cryptography [131] is used for implementing the security services. Two parties involved in communication can secure their transmission with the help of cryptography. For cryptography either symmetric or asymmetric key approach is used. In symmetric key cryptography, single shared secret key is used for encryption and decryption process and in asymmetric key cryptography different key pair is used for encryption and decryption. In case of symmetric key cryptography key distribution is a biggest challenge. Wireless sensor networks increases the problem of key distribution to many folds due to the broadcast nature of these networks, lack of deployment topology, limited memory and processing power. RSA [132], Diffie –Hellman [133], Elliptic Curve [134], Curve25519 [135] and FourQ [136] are some of the well-known key exchange protocols. Apart from these protocols in the recent times various protocols has been proposed [137-139]. Key Distribution Center (KDC) is one of the approach where keys are generated and distributed. Extensive literature study has shown that mobile agents for key exchange are very less used. Mobile agent is the code which is dynamically executed and can be dispatched to the remote location where it can resume its execution. As the number of users increases in the network, load on the KDC increases to manifolds. So, definitely there is a need to reduce load at the KDC with the help of some alternative mechanism. Mobile agent approach can also help in reducing the traffic at KDC to a great extent.

4.2 KEY DISTRIBUTION THROUGH FINGERPRINT BASED AUTHENTICATION

Key generation and distribution is a research challenge since the encrypted digital communication started. Key Distribution Center (KDC) is the most common mode of key generation and key distribution. KDC is always been the most profitable and worthy target for the adversaries. Once KDC is compromised then security protocols, encryption protocols, firewalls etc. will be of no use or will backfire. Current key exchange algorithms do not take into account authentication of nodes during key exchange. A new biometric based key exchange protocol (KDFBA) is proposed which makes use of the mobile agent for key exchange with proper authentication. Proposed protocol has been simulated on NS2 platform and is also compared with the existing approaches available against some parameters like traffic generated and timing analysis.

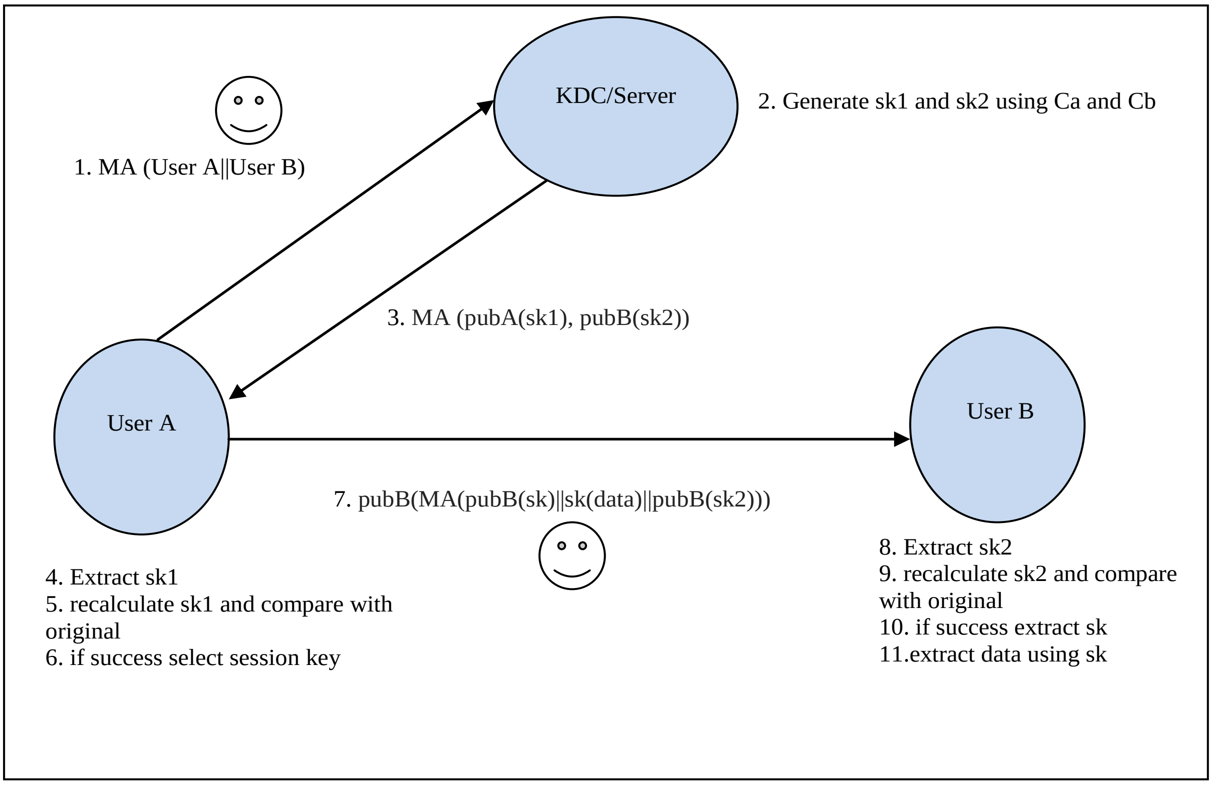

KDC/Server
2. Generate sk1 and sk2 using Ca and Cb
1. MA (User A||User B)
3. MA (pubA(sk1), pubB(sk2))
User A
User B
7. pubB(MA(pubB(sk)||sk(data)||pubB(sk2)))
4. Extract sk1
5. recalculate sk1 and compare with original
6. if success select session key
8. Extract sk2
9. recalculate sk2 and compare with original
10. if success extract sk
11.extract data using sk

Figure 4.1: Schematic of the KDFBA

Protocol is also tested against the black hole attack and the results for the same have also been compared with the existing approach. Results show that the proposed key exchange protocol reduces the traffic on the network considerably and also prevents the black hole attack. Figure 4.1 shows various steps of the proposed work.

Every user in the network registers to the KDC with its biometric data in the form of cancelable template. Cancelable template is the biometric data of the user in the protected form. As biometric data of the user must be stored in the protected form so, cancelable template generation is an important aspect for a biometric based authentication system. Here User A wants to securely communicate with User B with the help of KDC/Server. User A sends his willingness to the KDC with the help of mobile agent. KDC generates two keys (sk1 and sk2) using the cancelable template of user A and user B. KDC encrypts these keys using the public keys of respective parties and sends it to the user A. User A confirms the sk1 and selects a random session key (sk) for encryption of data and public key of user B for encryption of sk2. User A sends this to User B. User B extracts sk2 and confirms sk2. It then extracts the random session key and decrypts the data. Algorithm 4.1 shows the steps of KDFBA in detail.

In the proposed algorithm the fingerprint data of all registered users are stored in the form of cancelable template. Next section shows the method of generation of cancelable template [140].

Minutiae Points Extraction: Fingerprint picture is taken as data and minutiae feature are extracted from the fingerprint impression picture. For the most part, two sorts of minutiae points are considered in our approach named: ridge ending and ridge bifurcation points. (x, y, ϴ) triplet represents the minutiae point where (x, y) is the x and y value of the coordinates and ϴ is the minutiae point alignment angle.

Figure 4.2 shows the complete process for cancelable template generation using the Random Triangle Hashing. Above process shows five points which fall in the range of angle [0, 60], [120, 180], [240, 300] and [300, 360]. If a point doesn’t fall in the angle range then count is set to zero. This step is repeated for multiple triangles. Vector generated from multiple triangles are merged to form a hash vector. This technique is called Random Triangle Hashing which is used to generate a cancelable template.

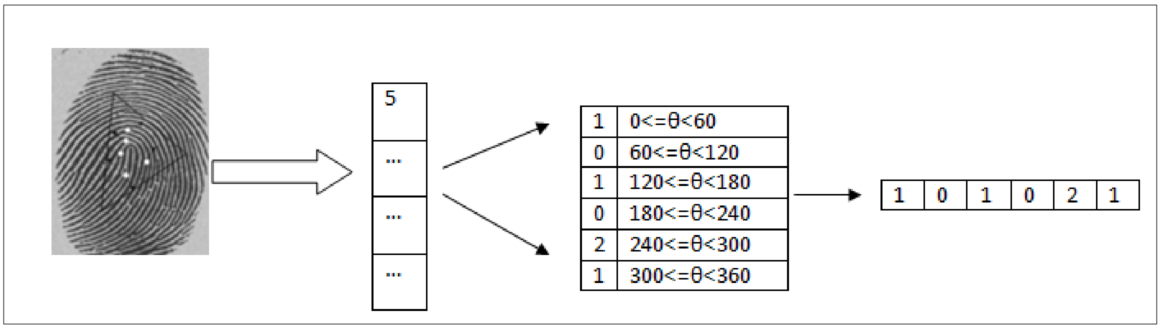


Figure 4.2: Conversions of biometric data to cancelable template

Algorithm 4.1 shows the algorithm for the KDFBA. Whenever a user wants to communicate on the network, it sends a request to key distribution centre (KDC). KDC fetches biometric data of both communicating nodes from database and generates the keys sk1 and sk2 using this. KDC creates a Mobile Agent (MA) and loads it with keys generated from the biometric data mentioned as sk1 and sk2 in the algorithm. These keys are encrypted using public keys of the respective nodes.

Algorithm: Key_Exchange_KDFBA

Input:

1. Address of UserA and UserB

2. Cancelable Template C_a and C_b of UserA and UserB

3. KDC address

Output: ***status of key exchange between UserA and UserB***

```
Send $U_a||U_b$ request to KDC
for each request at KDC do
    Extract Ca and Cb of UserA and UserB from database
    KDC generates session key sk1 and sk2 using Ca and Cb
    KDC creates MA loads it with MA(pubA(sk1), pubB(sk2)) and sends to A
end for
Node A extracts sk1 and recalculates sk1* using biometric data.
if(sk1!=sk1*) then
    abort and block KDC
else
```

```
11.     node A selects session key (sk) randomly
12.     load and encrypt the MA and dispatch to User B
13.     pubB(MA(pubB(sk)||sk(data)||pubB(sk2)))
14. end if
15. Node B extracts MA, sk and sk2 by its prvB.
16. Node B recalculates sk2* using biometric data.
17. if(sk2!=sk2*)
18.     abort and block UserA
19.else
20.     Node B extracts sk using prvB.
21.     Node B extracts data using sk.
22. after fixed interval of time repeat from step 11.
```

Algorithm 4.1: KDFBA algorithm

When this MA arrives at the requesting node from KDC, it extracts sk1 and recalculates sk1* using fingerprint data at the node. If the calculated key does not match with sk1 then it is assumed that KDC is compromised and algorithm will block the compromised KDC. If the key matches then node A select the session key randomly and encrypts it with public key of user B and loads mobile agent (MA). MA is then dispatched to user B. User B extracts sk2 using its private key and recalculates sk2*. If both keys are matched then node A and KDC is authenticated and data transmission proceeds which are encrypted with the session key sk. After some interval of time, sk expires and is recalculated using the same procedure. For message exchange, Mobile Agent is used which can help in reducing traffic in the network also.

4.2.1 Performance Metrics

Various metrics used for the evaluation purpose of proposed algorithm are Packet Analysis, Average Delay, Energy Consumption, Average Network Throughput and Packet Delivery Ratio.

***Packet Analysis*:** A network packet is a formatted unit of data carried by a packet-switched network. A packet consists of control information and user data, which is also known as the payload. Control information provides data for delivering the payload. Send, receive and forward packets during communication are used for packet analysis purpose.

***Average Delay*:** The delay in the network is measured based on the average time in which data packets are transmitted from source node toward the destination. Delays are initiated due to packet buffering, queuing and propagation delays. Probability of the packet drop is directly proportional to the distance between source and destination. As the distance increases, probability of packet loss also increases. $Average\ end2end\ delay$ is calculated using the following formula in equation 4.1:

$$\boxed{Average\ end2end\ delay = \frac{\sum_{i=1}^{n}(Received\ Packet\ Time - Send\ Packet\ Time) * 1000(ms)}{Total\ Number\ of\ Packets\ Delivered\ Successfully} \quad \ldots\ (4.1)}$$

***Energy Consumption*:** The total energy consumed is calculated by total energy used to transmit and receive packets by all nodes during the simulation. The energy consumption are summation of spend energy of all the nodes in the network. It includes summation of energy spend for communication (E_c), packet transmit (E_{pt}), packet received (E_{pr}), and idle packet (E_{pi}). Formula for the same is shown in equation 4.2.

$$\boxed{E_{consumption} = \sum_{i=1}^{n} E_c + E_{pt} + E_{pr} + E_{pi} \quad \ldots\ (4.2)}$$

Where, $E_{consumption}$ *= total consumed energy*

E_c*= Energy spend on communication*

E_{pt}*= Energy consumed in packet transmit*

E_{pr}*= Energy consumed in received packet*

E_{pi}*= Energy consumed in idle packet*

***Average Network Throughput*:** The average network throughput is the sum of total data packets reached to the destination within a given simulation time. A high ratio of dropped packets will ultimately lead to lower throughput and ultimately performance will be degraded.

Following formula shown in equation 4.3 is used for calculation of throughput

$$\boxed{Network\ Throughput = \frac{PacketSize}{(PacketArrival - PacketStart)} \qquad \ldots (4.3)}$$

Where, $PacketSize$ = size of ith packet of file reaching to destination

$PacketArrival$ = time when last packet arrived to destination

$PacketStart$ = time when first packet arrived to destination

***Packet Delivery Ratio (PDR)*:** Packet Delivery Ratio is measured as ratio of total data packets received by the destination to the number of packets generated by the source node. High packet transmission in the network causes high network performance. The mathematical calculation of PDR is shown in equation 4.4.

$$\boxed{Packetdeliveryratio\ (PDR) = \frac{\mathrm{R_p}}{\mathrm{R_g}}} \qquad \ldots.. (4.4)$$

Where, $\mathrm{R_p}$ = number of received packets

$\mathrm{R_g}$= number of packets generated

For comparison with the existing work, and to identify the need of authentication two scenarios are proposed, one is with authentication mentioned in the above section (KDFBA) and another is without authentication (KDFBWA). KDFBWA is key distribution with fingerprint without authentication. In this algorithm, key distribution is done through the same method as in KDFBA except the authentication part. So this algorithm is used for comparison with KDFBA in order to explain the need of authentication during key exchange mechanism. Comparison of these two scenarios is done with different parameters like network density, simulation time variation etc. Comparison clearly shows the need of authentication during the key exchange.

4.2.2 Simulation Result Analysis

The parameters for simulation of network model are shown in Table 4.1. The performance comparison analysis is done on Network Simulator 2 (NS2) platform against various types of scenarios like Network density, Simulation time, Number of network connections and Packet Size.

Various steps followed for implementation of the proposed algorithm are shown in Figure 4.3.

Table 4.1: Simulation Parameters of KDFBA using Mobile Agent in NS2.

Parameters	Values
Network Simulator	Network Simulator 2
Routing Protocol	AODV
Standard	IEEE 802.11
No. of Nodes	10, 20, 30, 40, 50
No. of Connections	2-8
Packet Size	128-2048 bytes
Simulation district	1000m × 1000m
Keys	Fingerprint
Simulation time	100, 200, 300, 400 and 500 Seconds
Application Layer	UDP
Traffic type	CBR
Antenna	Omni-directional
Channel	Wireless
Radio-propagation pattern	Two ray ground
Interface queue	Drop/Tail/PriQueue
Network interface mode	Phy / WirelessPhy
Performance Metrics	Packet Analysis, Average Energy, Throughput and Packet Delivery Ratio
Attack	Black hole

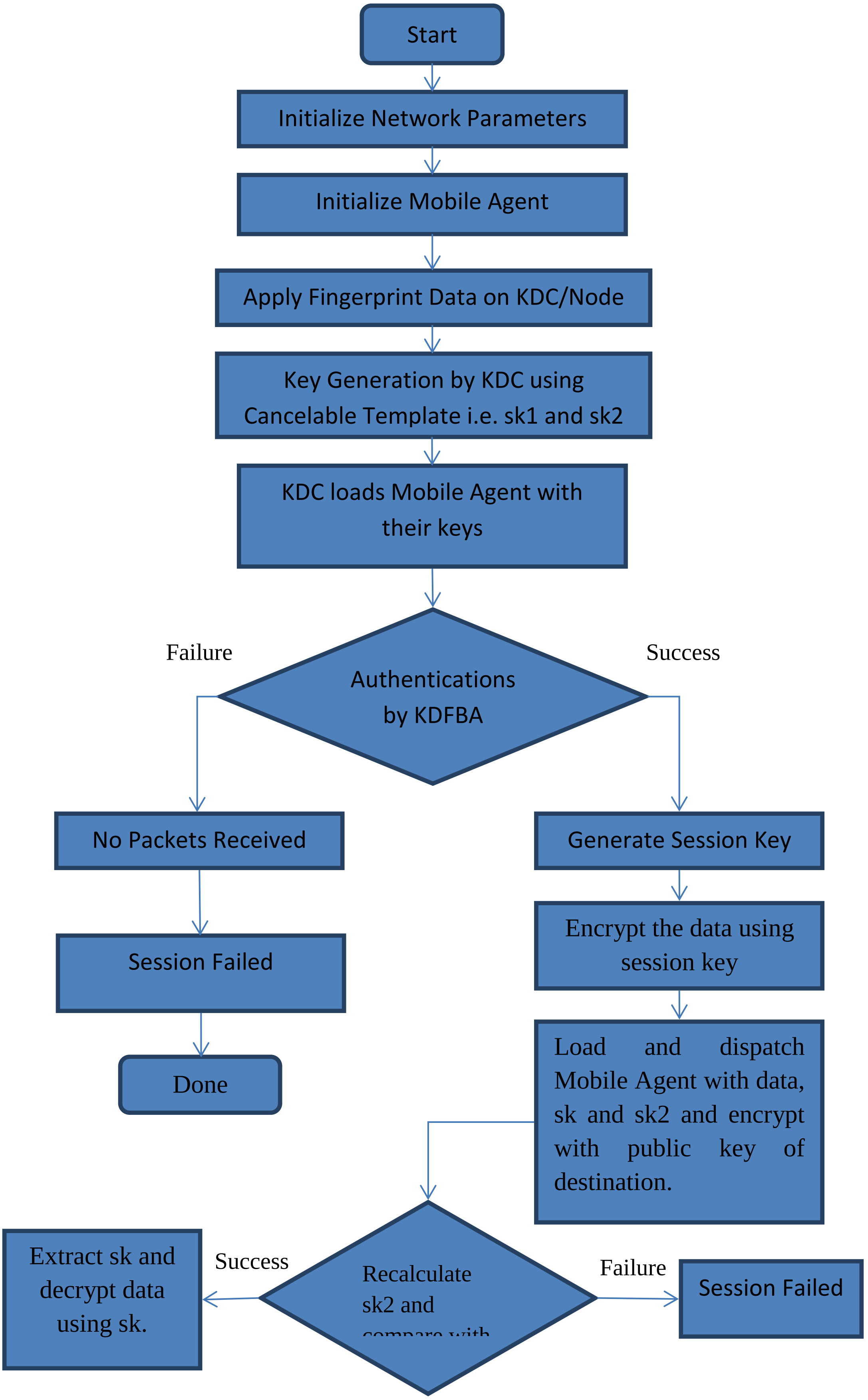

Start
Initialize Network Parameters
Initialize Mobile Agent
Apply Fingerprint Data on KDC/Node
Key Generation by KDC using Cancelable Template i.e. sk1 and sk2
KDC loads Mobile Agent with their keys
Failure
Authentications by KDFBA
Success
No Packets Received
Session Failed
Done
Generate Session Key
Encrypt the data using session key
Load and dispatch Mobile Agent with data, sk and sk2 and encrypt with public key of destination.
Extract sk and decrypt data using sk.
Success
Recalculate sk2 and compare with
Failure
Session Failed

Figure 4.3: Flow chart for Key Distribution through Fingerprint based Authentication

A. Effect of network density i.e. number of nodes on Mobile Agent using KDFBA and KDFBWA Algorithms

This section compares the effect of network density on the KDFBA. Same algorithm is also tested where authentication is not taken into consideration i.e. Key distribution fingerprint based without authentication (KDFBWA).

KDFBWA is tested with KDFBA using MA and without MA to prove the need of authentication during key exchange.

Both algorithms are compared for number of packets against the number of nodes as shown in Table 4.2. Algorithm is also tested against the blackhole attack. As AODV protocol for routing is used for simulation which requires next hop information from the neighbor nodes. Malicious node replies that it is having best route to the destination node but in reality it does not have any route to the destination. So, when the blackhole attack is simulated results in Figure 4.4 to Figure 4.6 clearly shows that KDFBA has an upper edge in terms of send, received and forward packets.

Algorithm is also analyzed for packet delivery ratio (PDR), average throughput and average energy. Data in Table 4.3 is captured for the analysis of the same. Graphs in Figure 4.7 and Figure 4.8 shows the advantage of KDFBA over KDFBWA. Figure 4.9 shows average energy consumed in both the algorithms are almost same.

B. Effect of Simulation Time variation on Mobile Agent using KDFBA and KDFBWA Algorithms

This section compares the effect of simulation time variation on KDFBA and KDFBWA algorithms with and without MA approach. Table 4 shows the data which is captured for analysis purpose. Graphs in Figure 4.10 to Figure 4.12 clearly show the advantage of KDFBA using MA over KDFBWA.

In Table 4.5 data for analysis of PDR, average throughput and average energy is collected. Graphs in Figure 4.13 to 4.15 shows the advantage of KDFBA to a great extent.

Table 4.2: Effect of network density

No. of Nodes	Key Distribution through Fingerprint					
	Send Packet (count)		Received Packet (count)		Forward Packets (count)	
	KDFBA using MA	KDFBWA using MA	KDFBA using MA	KDFBWA using MA	KDFBA using MA	KDFBWA using MA
10	2258	2252	1756	754	512	1499
20	2259	2252	2258	753	1526	2000
30	2260	2250	2260	752	1534	2000
40	2258	2252	2256	1249	1531	519
50	2259	2252	2258	1751	1545	1005

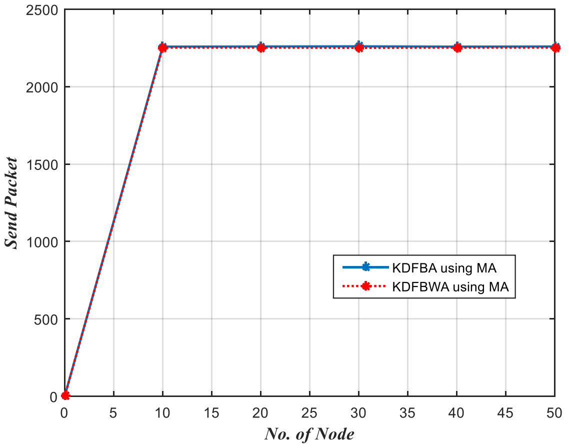


Figure 4.4: Number of packets sent
(KDFBA vs KDFBWA)

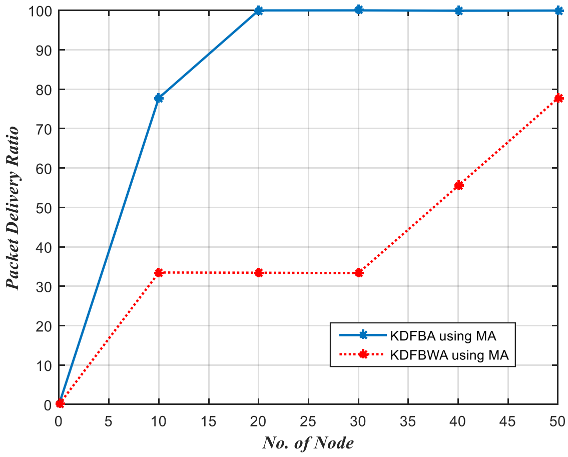


Figure 4.5: Received Packets
(KDFBA vs KDFBWA)

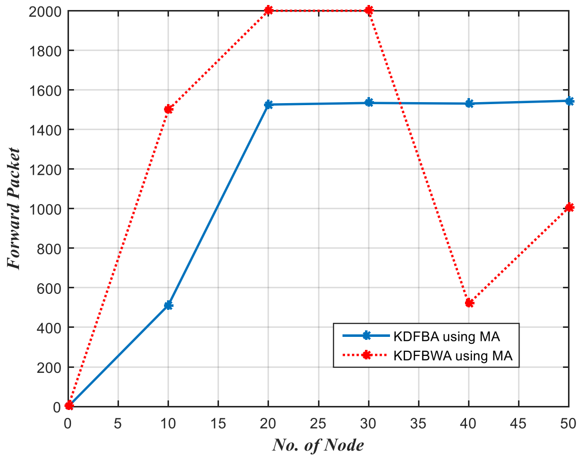


Figure 4.6: Forward Packets
(KDFBA vs KDFBWA)

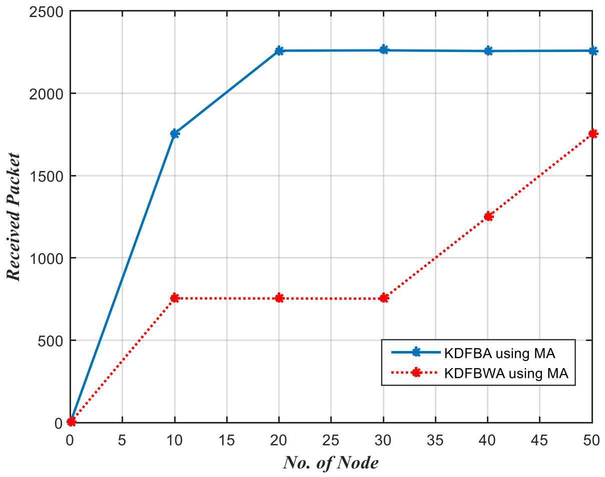


Figure 4.7: Packet Delivery Ratio
(KDFBA vs KDFBWA)

Table 4.3: Average PDR, Throughput and Energy vs No. of nodes

	Key Distribution through Fingerprint					
	Packet Delivery Ratio (count)		**Average Throughput (kbps)**		**Average Energy (J)**	
No. of Node	**KDFBA using MA**	**KDWA using MA**	**KDFBA using MA**	**KDWA using MA**	**KDFBA using MA**	**KDWA using MA**
10	77.77	33.48	79.7626	44.0884	20	20
20	99.99	33.43	102.526	44.029	10	10
30	100	33.32	102.526	44.028	6.67	6.67
40	99.91	55.46	101.685	56.8625	5	5
50	99.95	77.71	102.45	79.715	4	4

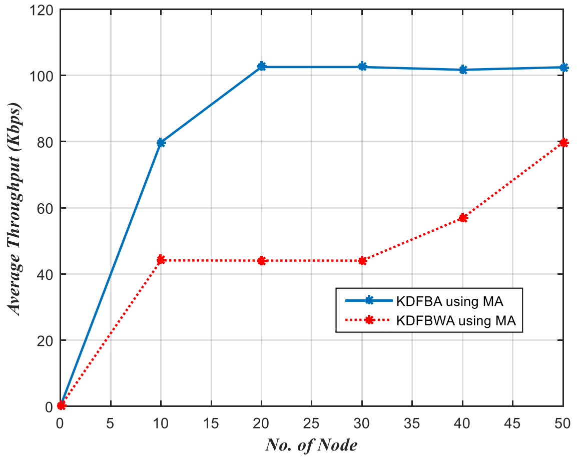


Figure 4.8 Average Throughput
(KDFBA vs KDFBWA)

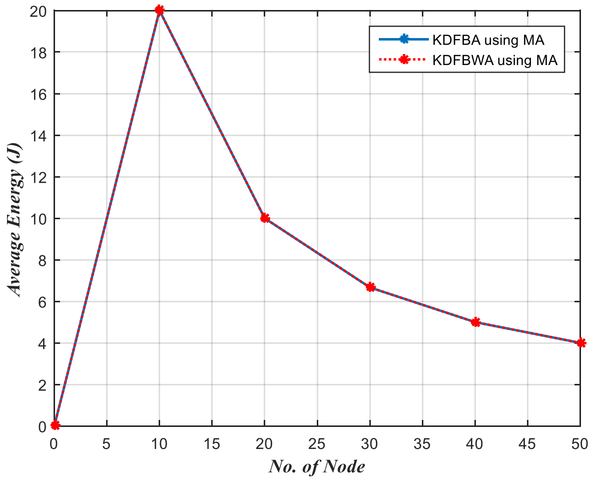


Figure 4.9 Average Energy
(KDFBA vs KDFBWA)

Table 4.4: Packet Analysis based on the simulation time variation

Simulation Time (Seconds)	**Key Distribution through Fingerprint**					
	Send Packet (count)		**Received Packet (count)**		**Forward Packets (count)**	
	KDFBA using MA	**KDWA using MA**	**KDFBA using MA**	**KDWA using MA**	**KDFBA using MA**	**KDFBWA using MA**
100	4007	4001	2805	839	5864	2986
200	9009	9003	2805	924	5864	8362
300	14007	13670	3929	924	9358	9902
400	19007	16170	3929	924	9358	9902
500	24007	18670	3929	924	9358	9909

Table 4.5: PDR, Throughput and Energy vs Simulation time

Simulation Time (Seconds)	Key Distribution through Fingerprint					
	Packet Delivery Ratio (count)		Average Throughput (kbps)		Average Energy (J)	
	KDFBA using MA	KDWA using MA	KDFBA using MA	KDWA using MA	KDFBA using MA	KDFBWA using MA
100	70	20.96	127.551	38.5277	3.34	4
200	43.6119	10.26	127.551	21.528	3.34	4
300	28.65	6.759	111.656	21.528	3.6	4
400	20.67	5.7142	111.656	21.528	3.6	4
500	16.36	4.94	111.656	21.528	3.6	6

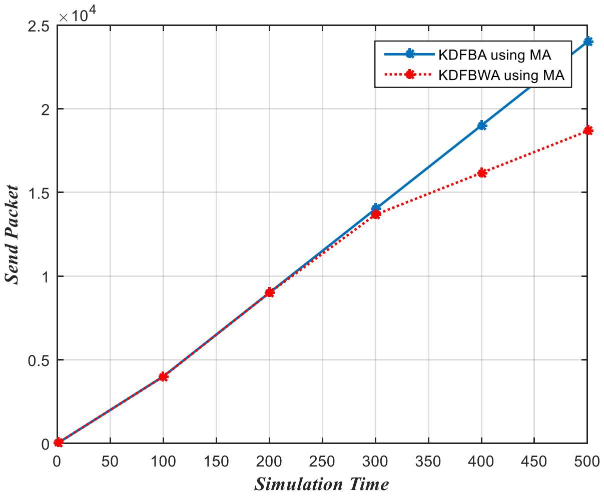


Figure 4.10: Send Packet vs Simulation Time

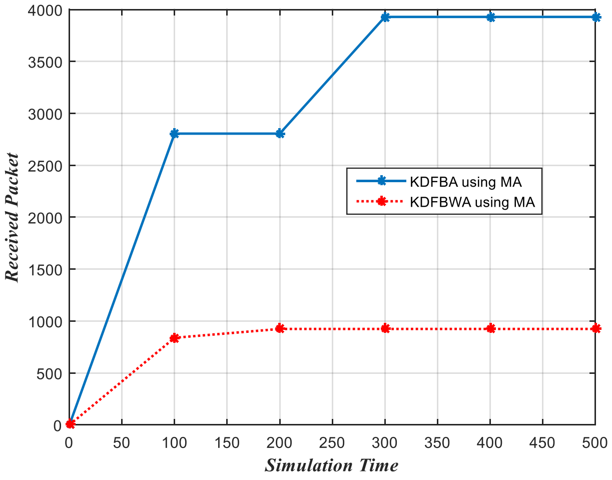


Figure 4.11: Received packets vs Simulation Time

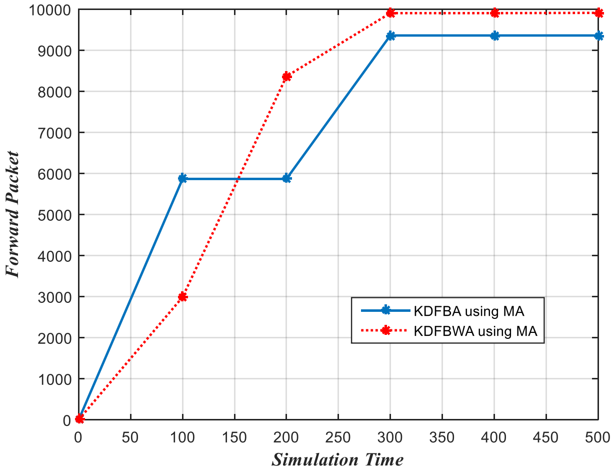


Figure 4.12 Forward packets vs Simulation Time

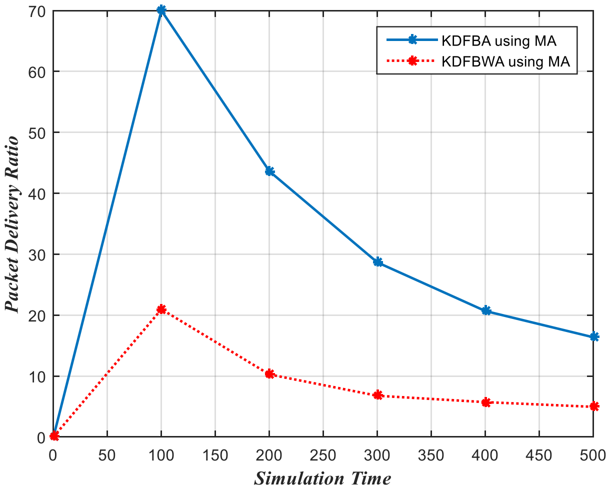


Figure 4.13 Packet Delivery Ratio vs Simulation Time

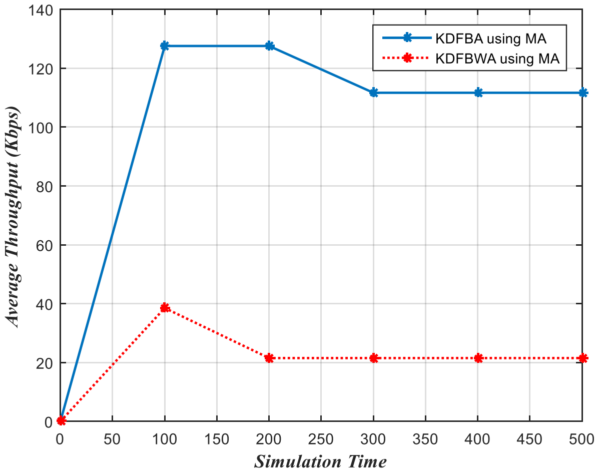


Figure 4.14 Average Throughput vs Simulation Time

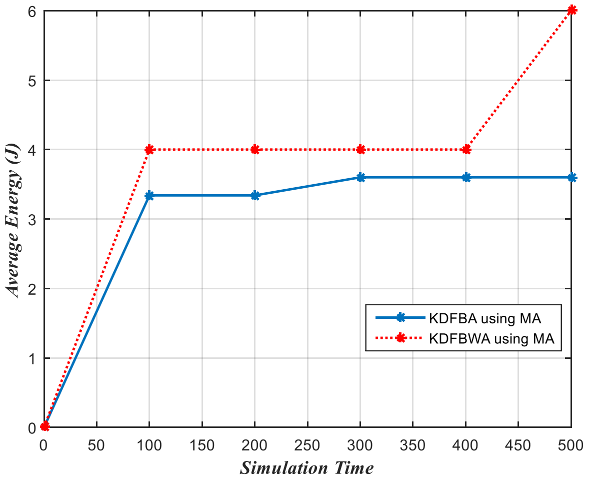


Figure 4.15 Average Energy vs Simulation Time

C. Effect of network connection variation on Mobile Agent using KDFBA and KDWA Algorithms

As the number of connections increases in the network, the performance of the network varies. Proposed algorithm is also tested against increasing number of connections in the network.

Data in the Table 4.6 is captured during the simulation for the analysis purpose and graphs in Figure 4.16 to Figure 4.18 clearly show the advantage of KDFBA using mobile agent. As blackhole attack is simulated and we go without authentication in KDFBWA performance degrades. But in case of KDFBA performance is good in terms of send, received and forward packets.

Table 4.6: Packet analysis using variation in number of connections

	Key Distribution through Fingerprint					
	Send Packet (count)		**Received Packet (count)**		**Forward Packets (count)**	
No. of Connection	**KDFBA using MA**	**KDWA using MA**	**KDFBA using MA**	**KDWA using MA**	**KDFBA using MA**	**KDFBWA using MA**
2	2257	2250	2179	1488	5011	4931
4	3507	3500	2657	896	5050	4505
6	1758	1751	1424	848	3569	2095
8	2258	2252	2253	750	4030	1521

Table 4.7: PDR, Throughput, Energy vs No. of Connection

No. of Connection	Key Distribution through Fingerprint					
	Packet Delivery Ratio (count)		**Average Throughput (kbps)**		**Average Energy (J)**	
	KDFBA using MA	**KDWA using MA**	**KDFBA using MA**	**KDWA using MA**	**KDFBA using MA**	**KDFBWA using MA**
2	96.54	66.13	98.97	67.7284	3.2	4
4	75.76	25.6	100.71	80.3745	3.2	4
6	81	55.35	64.58	60.37	3.2	4
8	99.77	33.33	102.412	61.428	3.2	4

In Table 4.7 data for analysis of PDR, Average throughput and Average Energy is captured.

Graphs in Figure 4.19 to Figure 4.21 shows the performance of KDBFA is much better in terms of PDR, Throughput and Energy when even blackhole attack is simulated.

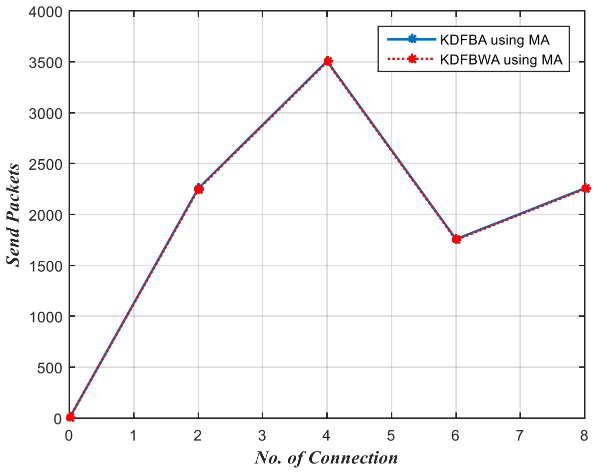


Figure 4.16: Send Packets vs No. of connections

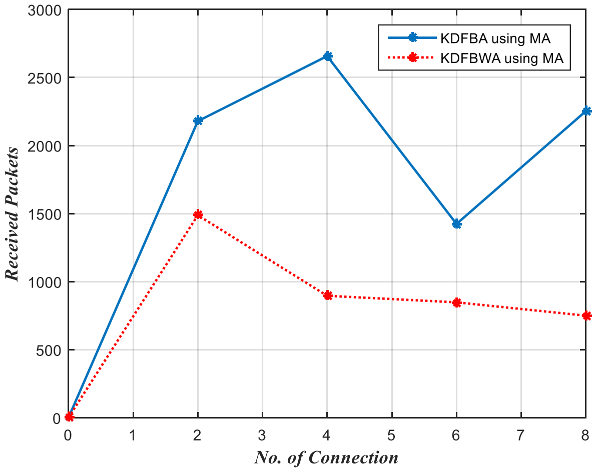


Figure 4.17: Received Packets vs No. of connections

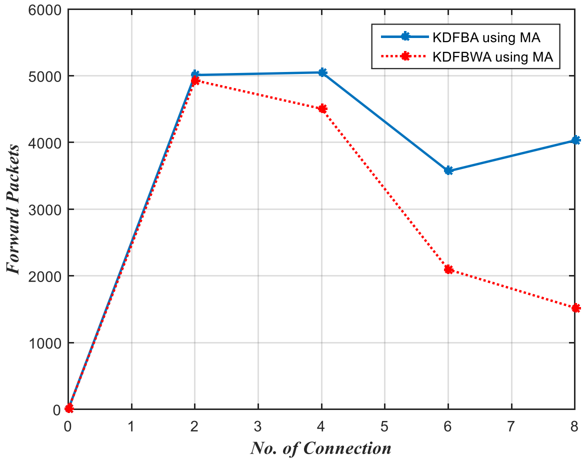


Figure 4.18: Forward Packets vs No. of connections

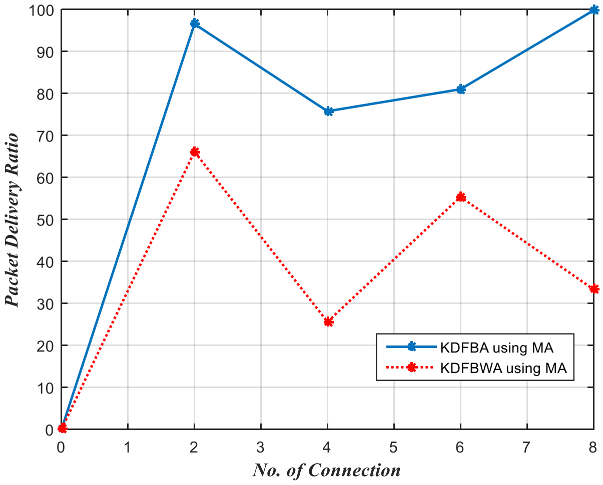


Figure 4.19: Packet Delivery Ratio vs No. of connections

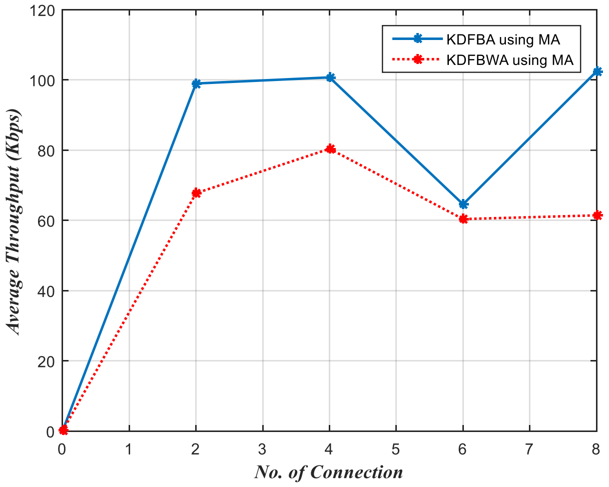


Figure 4.20: Average Throughput vs No. of connection

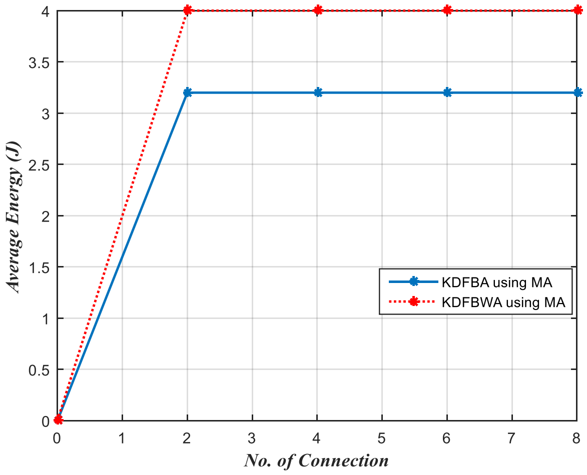


Figure 4.21: Average Energy vs No. of connection

D. Effect of packet size change on the Mobile Agent using KDFBA and KDFBWA Algorithms

Proposed algorithm is also tested against the variable size of packets. Table 4.8 shows the data captured for the analysis purpose and graphs in Figure 4.22 to Figure 4.24 shows that performance of KDFBA is better using mobile agent.

In Table 4.9 data for PDR, Average throughput and average energy is captured during the blackhole attack simulation. Graphs in Figure 4.25 to Figure 4.27 shows the performance of KDFBA using MA is much better.

Table 4.8: Packet analysis against variation in Packet Size

Packet Sizes (bytes)	Key Distribution through Fingerprint					
	Send Packet (count)		Received Packet (count)		Forward Packets (count)	
	KDFBA using MA	KDWA using MA	KDFBA using MA	KDWA using MA	KDFBWA using MA	KDFBWA using MA
128	18022	18016	9831	6003	17907	6810
256	9014	9008	6824	3010	14841	4027
512	4510	4502	3585	1502	9853	2886
1024	2258	2252	2125	750	4454	1521
2048	2258	2252	2125	750	4454	1521

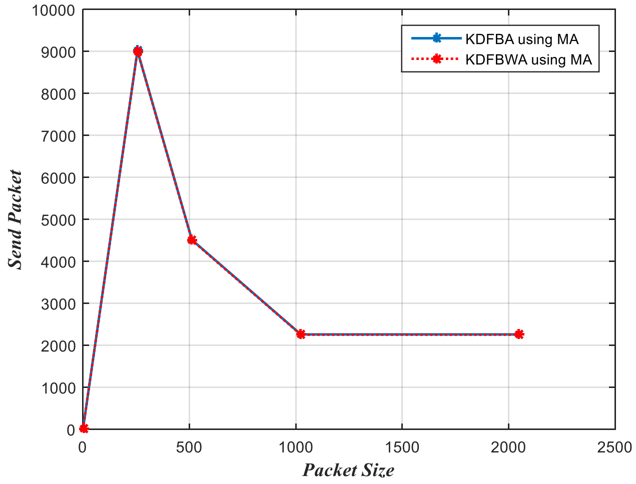


Figure 4.22: Send Packet vs Packet Size (bytes)

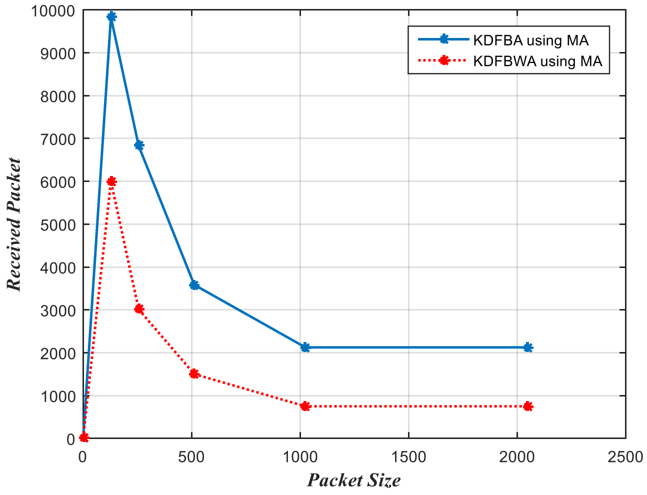


Figure 4.23: Received Packet vs Packet Size (bytes)

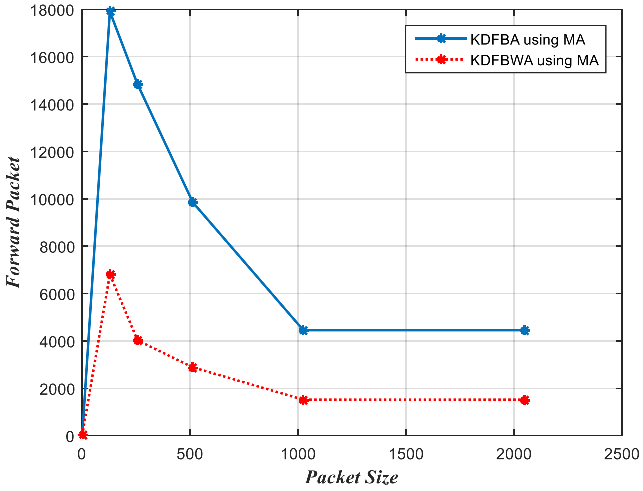


Figure 4.24: Received packet vs Packet Size (bytes)

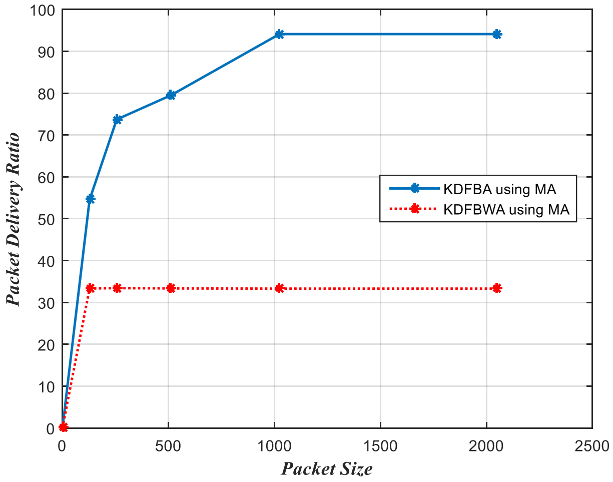


Figure 4.25: PDR vs Packet Size (bytes)

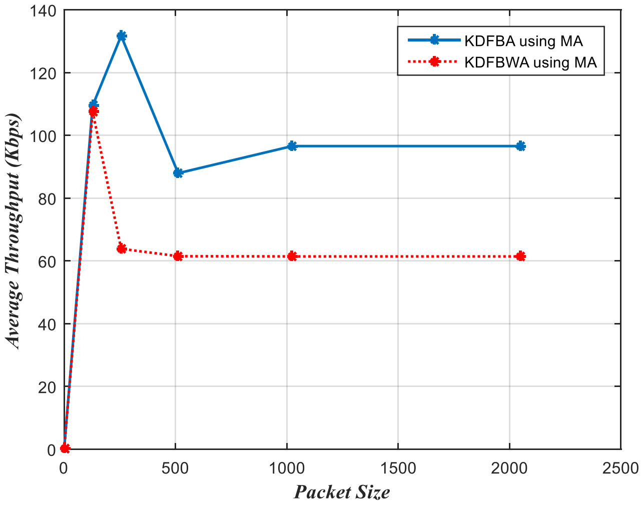


Figure 4.26: Avg. Throughput vs Packet Size (bytes)

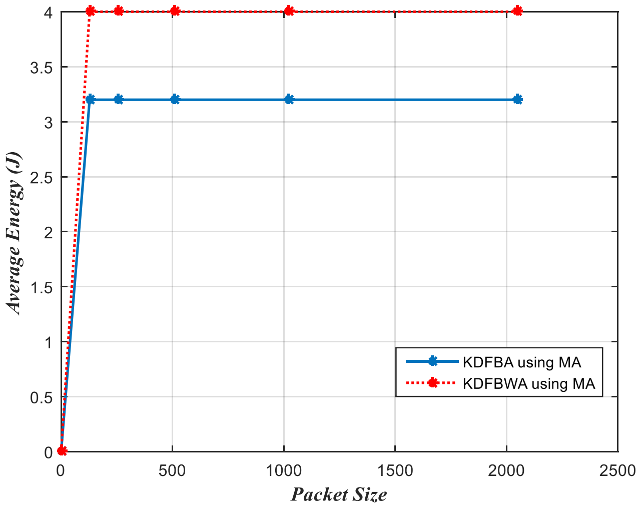


Figure 4.27: Average Energy vs Packet Size (bytes)

Table 4.9: Performance in terms of Packet Delivery, Throughput and Energy

Packet Sizes (bytes)	**Key Distribution through Fingerprint**					
	Packet Delivery Ratio (count)		**Average Throughput (kbps)**		**Average Energy (J)**	
	KDFBA using MA	**KDWA using MA**	**KDFBA using MA**	**KDWA using MA**	**KDFBA using MA**	**KDWA using MA**
128	54.55	33.32	109.466	107.563	3.2	4
256	73.7	33.41	131.707	63.8634	3.2	4
512	79.49	33.34	87.949	61.5145	3.2	4
1024	94.10	33.30	96.586	61.428	3.2	4
2048*	94.10	33.30	96.586	61.428	3.2	4

4.3 Comparison Analysis

The proposed KDFBA and KDFBWA work is also compared with the existing DSR with both mobile agent and without mobile agent that uses hybrid encryption scheme to improve performance. DSR with mobile agent uses symmetric keys to encrypt the routing data and the authorization of communicating nodes. The proposed scheme is compared against the packet delivery ratio and average end to end delay as shown in Table 4.10 and Table 4.11 respectively. Graph in Figure 4.28 shows the graphical representation of comparison. Graph clearly shows the KDFBA has an upper edge as compared to other schemes. As fingerprint based mechanism provides robust mechanism against authentication and mobile agent helps in reduction of traffic, so the proposed scheme provides improvements in packet delivery ratio as compared to the existing schemes like DSR.

Table 4.10: Comparison of various schemes

Packet Delivery Ratio (percentage)					
Simulation Time (Seconds)	**KDFBA using MA**	**KDWA using MA**	**DSR using MA[32]**	**DSR without MA**	**DSR in ns2**
0	98.50	97.09	94.90	96.8	90.80
30	98.62	98.19	95.90	97.80	93.80
60	98.87	98.37	96.40	98.00	93.81
120	98.91	98.71	97.00	98.15	95.60
300	99.1	98.87	98.35	98.50	96.00
600	99.32	99.37	99.12	99.22	97.35
900	99.79	99.78	99.78	99.78	97.99

Figure 4.29 shows the graph of comparison which shows there is a significant improvement in the average end to end delay of around 33% in the proposed scheme. This calculation is based on comparison of average end to end delay of proposed and existing scheme. For example for simulation time of 60 seconds, average end to end delay is 2.1 seconds in case of KDFBA and 6.2 seconds in case of DSR in NS2. So improvement in average end to end delay is:

$$= (2.1/6.2)*100$$

$$= 33.87\ \%$$

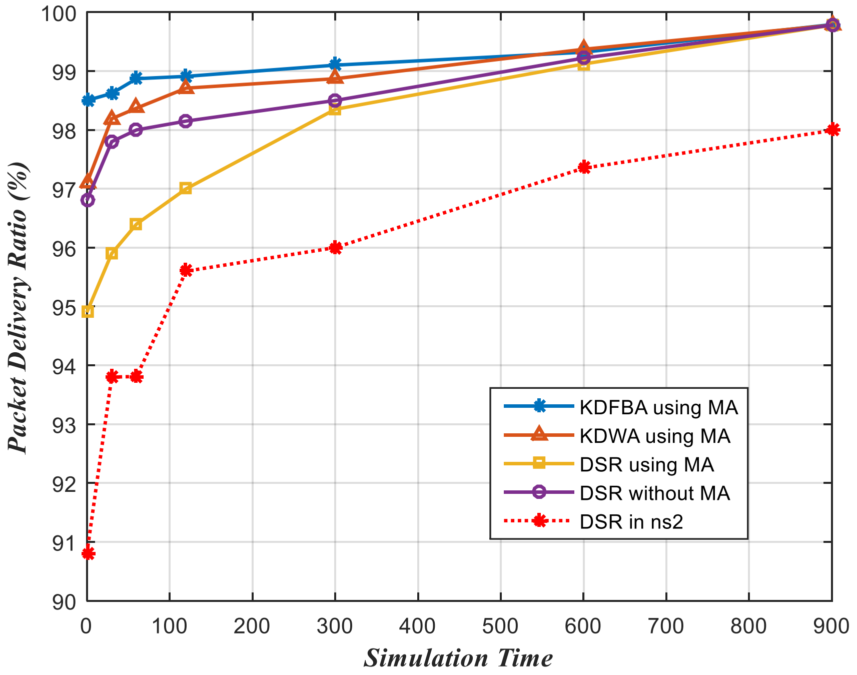


Figure 4.28: Comparison of various schemes

Table 4.11: Average End to End delay of various schemes

Average End to End Delay (seconds)					
Simulation Time (seconds)	**KDFBA using MA(seconds)**	**KDWA using MA(seconds)**	**DSR using MA[32] (seconds)**	**DSR without MA(seconds)**	**DSR in NS2(seconds)**
0	2.1	1.32	3.1	1.9	6.7
30	2.1	1.19	3.1	1.9	6.1
60	2.1`	1.07	3.1	1.9	6.2
120	2.1	1.03	3.1	2.0	6.9
300	2.1	1.04	3.1	1.9	7.0
600	2.1	1.04	3.1	1.9	7.5
900	2.1	1 .04	3.1	2.1	7.0

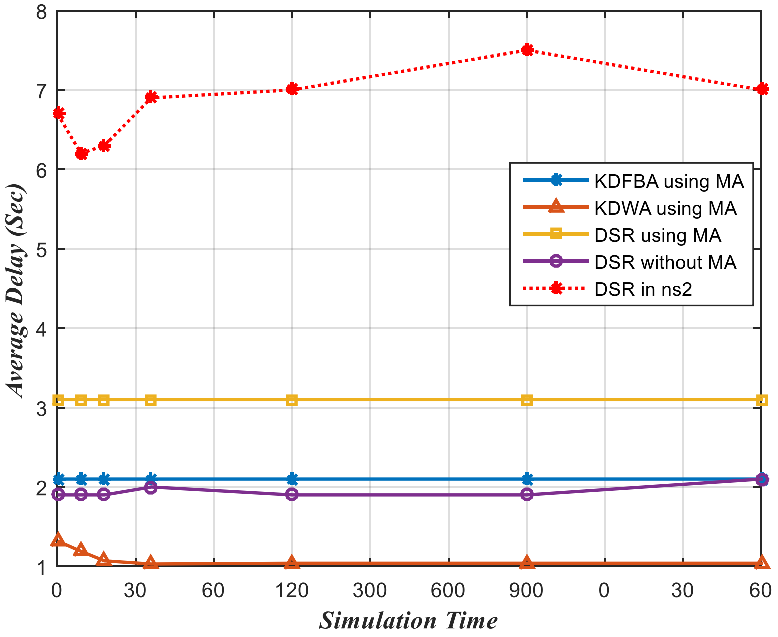


Figure 4.29: Average Delay of various schemes

4.4 Summary

In this chapter a new biometric based authentication and key exchange method is proposed. Authentication is based on biometric data registered on the server. Biometric data is never transmitted over the network. So, it is safe against adversaries. The proposed protocol KDFBA is simulated on NS2 platform and results with respect to various parameters are compared with the existing method. Black hole attack is also simulated and results are extracted and compared against various parameters. Experimental results clearly show that the proposed method gives better results than existing methods in terms of packet delivery ratio and average delay. We also carried the simulation and comparison of proposed method with one of the existing method i.e. DSR with mobile agent. Results clearly show that proposed scheme gives considerable improvement in terms of packet delivery ratio and average delay.

CHAPTER 5

DEVICE FINGERPRINT BASED AUTHENTICATION USING MOBILE AGENTS

5.1 INTRODUCTION

In the recent years, traffic on the internet has grown to a great extent. Number of users on the internet is growing continuously. In India only 323 million people who are 24 percent of country’s population accessed the internet in 2016. By 2021 there will be more than 635.8 million internet users in India only. According to the Cisco VNI, 2017 globally IP traffic growth is tremendous as and is increasing day by day. The report shows that growth of IP traffic from 2016 to 2021 is around 24%. As the traffic on the internet is growing at an unimaginable rate, there is a need for updating existing techniques which can reduce the traffic on the internet. Mobile agent technology can be one of the solutions to reduce the traffic on the internet. Mobile agent is an autonomous code which can roam inside the network by following some of the nodes which is called itinerary of the mobile agent. This itinerary can be dynamic or static. Mobile agent technology has seen a good amount of growth over the period of time. In some scenarios, this technology can be used instead of client server architecture for reduction in the traffic around the network. In adhoc wireless networks, authentication is always a big challenge. Recently, Device fingerprinting has emerged as one of the solutions for collecting information about the devices which are connected in a wireless network. Device fingerprinting technique can be used in mobile agent based frameworks for authentication of mobile agent and the device which generated this. The main contribution of this chapter is to develop a technique using Device Fingerprinting mechanism to authenticate the mobile agents in the **M**obile **A**gent **B**ased **F**ramework for **W**ireless **A**uthentication (**MABFWA**). Proposed technique will also help in preventing various attacks like man in the middle, hijacking, replay, jamming and eavesdropping. This chapter concentrates on providing a solution for authentication of the devices meanwhile reducing the traffic on the internet.

5.2. DEVICE FINGERPRINT AND MOBILE AGENT BASED AUTHENTICATION

This section will describe the overview for device fingerprint based authentication model. Model describes the working of various components like **A**ccess **P**oint (AP), Authenticator (**A**) and **A**uthentication Server (AS). Various mobile nodes are connected to the access point with the help of 802.11 technologies. Architecture allows heterogeneous communication devices to contact each other in a secure manner. Here the authenticator acts as an interface between the AP and the AS. The authentication process goes through the access point to authenticator and then to the authentication server. Figure 5.1 shows the architecture and working of Device Signature based Authentication Model. Here multiple **AP**'s are connected to the **A**uthenticator (**A**). Authenticator in turn is connected to the **A**uthentication **S**erver (**AS**).

1. **Access Point (AP)**: Access point provides the network access to multiple devices within the wireless range. Multiple access points can be within the range of a single device. Device is connected to an access point with the strongest signal after registration process (if not registered) and authentication. An access point supports multiple devices.

2. **Authenticator (A):** All access points are connected to the authenticator for authentication purposes. Authenticator contains three modules:

- ***Device Fingerprint Scanner*:** It extracts the device fingerprint parameters from the mobile agent and creates fingerprint with the help of that parameters. It stores it in cache memory and forwards it to the Authentication Server for registration purpose. If the device requires authentication then device fingerprint is forwarded to the cache memory.

- ***Cache Memory*:** Cache memory stores the fingerprint of most recent devices which are connected to the AP or the devices which are recently registered on the network. Cache memory forwards device fingerprint to the threshold comparator.

- ***Threshold Comparator*****:** It compares two fingerprints and gives true or false depending upon the value of the threshold matched.

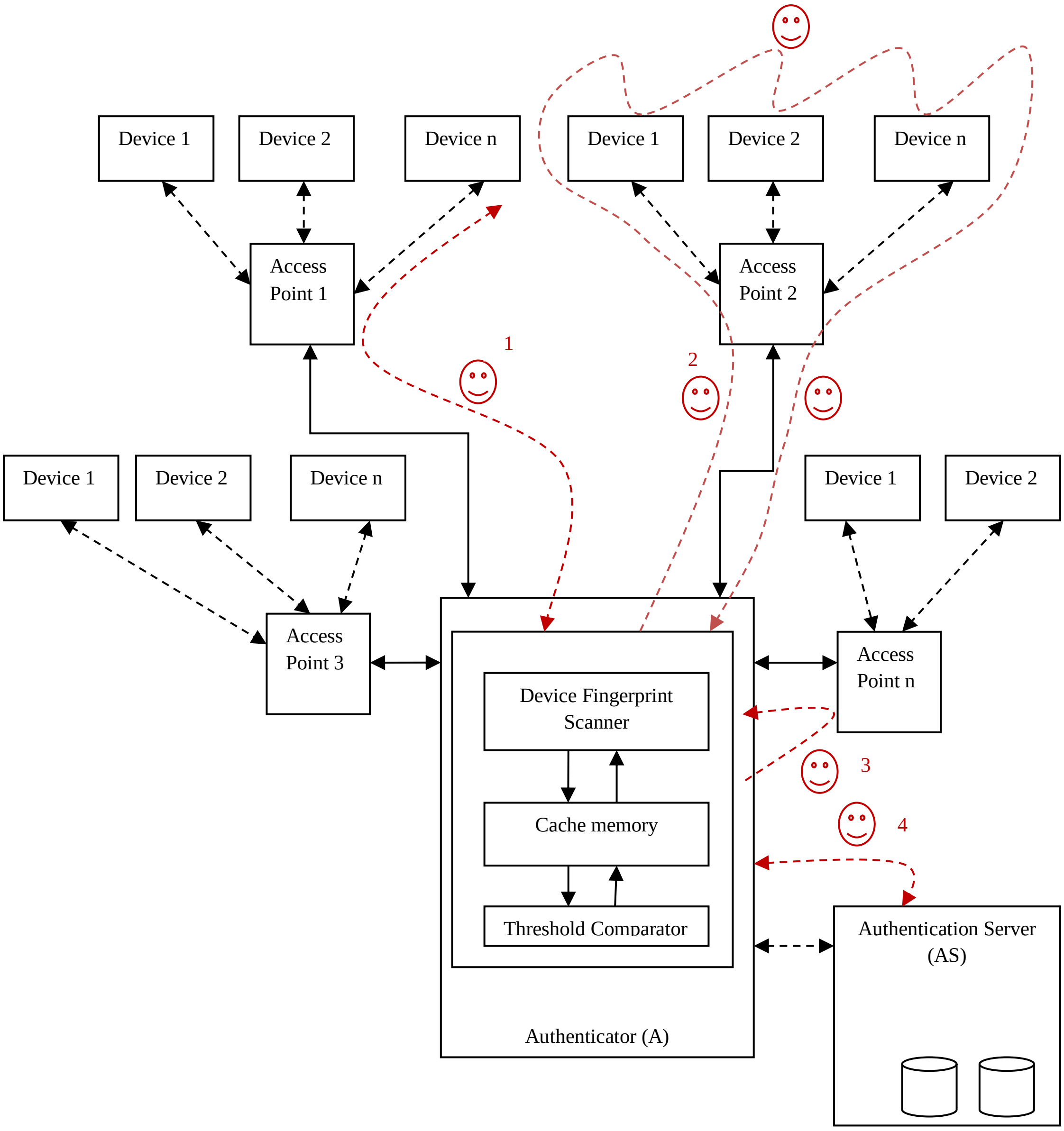


Figure 5.1: Architecture and working of the proposed model

3. **Authentication Server (AS): A**uthenticator is connected to the **A**uthentication **S**erver**. AS** provides the necessary authentication information for the devices and access points within the network. It registers all the device signatures in the database and provides the signature value when required by the authenticator.

5.2.1 WORKING OF PROPOSED MODEL

Proposed model supports both single device authentication and multi device authentication. Step 1 in Figure 5.1 shows the single device authentication involving only the single device. Multi device authentication will take more time as compared to single device authentication as in former case mobile agent will have to traverse through multiple devices for device signature extraction. Step 2 in Figure 5.1 shows the description of multi device authentication involving multiple devices.

***Single Device Authentication*:** Single device authentication is required when a new device comes into the range of an access point and wants to connect to the network. This single device can be either a user mobile device or a fixed access point.

Multi Device Authentication: Proposed model also supports the authentication of multiple devices in one go. Mobile agent from authenticator follows the itinerary for multiple devices and carries the device signatures of these devices. If the signature matches to the calculated one then it gives success otherwise failure. This loaded mobile agent then comes to the authenticator along with necessary information.

5.2.2 DEVICE FINGERPRINT CALCULATION

Device fingerprint is a unique signature of a device. This device fingerprint is generated depending upon the various parameters. These parameters will be extracted from the device using mobile agent which is being sent by the Authenticator and executed on the target machine.

Various parameters that are used to calculate a device signature value are listed below. It may be the case that device have only some parameters value. Using these available parameters of the device, fingerprint will be calculated.

Parameters to calculate device fingerprint:

- ***Geo location lat/long*:** This is the geographic location of a device. The location can be calculated by the network or **G**lobal **P**ositioning **S**ystem (GPS) (if supported).

- ***IP addresses*****:** This is the unique 32/128 bit address which is allocated to the device. The version of the IP address can be IPv4 or IPv6.
- ***MAC addresses*****:** This is the **M**edia **A**ccess **C**ontrol (MAC) address which is assigned to the network interface card of the device.
- ***Network ID*****:** Every existing network has a unique id value associated with it. This is the id of the network to which the device is connected.
- ***Browser name and version*****:** This is the name and version of the browser currently installed. Device may have more than one browser then the details of all are included in the fingerprint.
- ***OS name and version*****:** This is the name and current version of the operating system of the device.
- ***Electronic Serial Number (ESN)*****:** This number is generated by the manufacturer on the microchip on mobile devices.
- ***International Mobile Equipment Identity or Mobile Identification Number (IMEI / MIN)*****:** It is the unique identification number that all mobile phones have.
- ***Received Signal Strength Indicator (RSSI)*****:** With this parameter the received signal power from access point to mobile device can be measured. RSSI is usually measured in decibels relative to a milliwat (dBm). The stronger the signal is closer it is to zero.
- ***Basic Service Set Identifier (BSSID)*****:** This is the MAC address of the access point which is the combination of the organization unique identifier and identifier for the radio chipset.
- ***Service Set Identifier (SSID)*****:** This is the name which is assigned to the wireless local area network. Mobile devices use SSID to identify the network and to join the network also.
- ***CenterFreq*****:** It is the measure of the frequency between upper and lower frequency cutoffs. Arithmetic mean or geometric mean of the lower and upper cutoff frequency is used to define this.
- ***Frequency*****:** This is the frequency value in MHz. This is the value over which the communication will take place.

- ***Level*****:** This is the strength of the GSM/CDMA signal received. Lower the signal level, lower will be the level of the signal.
- ***Timestamp*****:** This contains the time of a particular communication between device and the access point. This value is generally mentioned in microseconds.
- ***VenueName*****:** This is the name of the location that is distributed by the access point.
- ***Device to AP RTT Supported*****:** This is the value of the inbuilt function supported by the device. Using this function, mobile device can calculate the distance between the access point and device.
- ***isTdlsSupported*****:** This is the value of the tunneled direct link setup. IEEE 802.11z supports this. Its value will be true if supported.
- ***WifiConfiguration.GroupCipher*****:** This is the cipher mechanism supported by device. Its value can be CCMP which is AES in Counter mode with CBC-MAC or TKIP which is Temporal Key Integrity Protocol.
- ***LinkSpeed*****:** This is the current speed in Mbps of the channel between device and the access point.
- ***Capabilities*****:** It describes the authentication, key management, and encryption schemes supported by the access point.
- ***Camera characteristics*****:** If a device has camera features then camera characteristics can also be included in the signature.
- ***Gateway address*****:** This is the address of the router which is maintained by **I**nternet **S**ervice **P**rovider (ISP).

These parameters are entered in to the device fingerprint generator which generates the device fingerprint. Some of the parameters mentioned above will give the real time dynamic values like location, IP address, Link Speed and some values will remain unchanged like IMEI number, MAC address and camera characteristics etc.

Every time the mobile agent fetches the device fingerprint parameters, it is compared with the earlier stored one. If the percentage of signature matching is more than threshold, then access can be granted.

This threshold value can be decided by the administrator depending upon the criticality of the application. In the proposed model, a device needs to be registered to the **A**uthentication **S**erver for having access of the network resources. The step by step registration algorithm is shown in Algorithm 5.1.

```
Algorithm: Registration
Input:
1. Number of Nodes
2. Client addresses i.e. client(i)
3. Authenticator address i.e. authenticator_address
4. Authentication Server address
5. itinerary[ ]
Output: Success or failure of registration.
1.  create itinerary[ ] of i number of nodes for MA
2.  create MA sign_Collector( )
3.  while itinerary[ ] is not empty
        3.1 if node itinerary[i] is active then
            3.1.1 dispatch sign_Collector( ) to the itinerary[i] node.
            3.1.2 compute device signature and append it to the results
        3.2 else
            3.2.1 continue
        3.3 end if
4.  end while
5.  authenticator stores all signatures in the cache memory and forwards it to the
    Authentication Server.
```

Algorithm 5.1 Algorithm for registration

After the registration, device can use the network services. If a device comes later on in the network for the use of network services, then the device needs to be authenticated. Mobile agent used for the authentication purpose, collects all the authentication information from either single or multiple devices. The itinerary created in this approach can use the location fingerprinting technique to create the three dimensional graph of various mobile stations. Later on **L**ocal **C**losest **F**irst (**LCF**) or **G**lobal **C**losest **F**irst (**GCF**) technique is applied to create the itinerary for the mobile agent. Authentication steps are shown in Algorithm 5.2.

```
Algorithm: Authenticator

Input:
1. Necessary information for authentication process e.g. username, password
   or device fingerprint
2. Client addresses i.e. client(i)
3. Authenticator address i.e. authenticator_address
4. Authentication server address
5. itinerary[ ]
Output: Success or failure of authentication.


1. create and load mobile agent signature_Verifier( ) with device signature of all
the devices.
    1.1 For all the devices whose signature are not present at the Authenticator's
        cache memory
            1.1.1 creates a mobile agent signature_Collector(devices [ ]), loads it
                  with the device id's of all devices.
            1.1.2 Server loads the signatures from the database.
            1.1.3 dispatch signature_Collector( )  to the Authentication Server.
            1.1.4 append the signatures to signature_Verifier( ) agent.
    1.2 end for
2. create itinerary[ ] of MA using itinerary algorithm like LCF or GCF
3. while itinerary[ ] is not empty
    3.1 dispatch mobile agent signature_Verifier( ) from Authenticator to node
```

```
        itinerary [i].
    3.2 agent collects and checks the device fingerprint against the stored
        signature.
    3.3 agent gives the success/ failure message depending upon the threshold
        value of the signature verification result.
4. end while.
5. Last client dispatches the signature_Verifier(  ) to authenticator.
6. Authenticator collects success or failure message from mobile agent
   signature_Verifier( ) based on the device signature verification and provides
   the access.
```

Algorithm 5.2 Algorithm for authenticator

Authentication Server stores information of all devices that are registered and authorized to use network resources. Algorithm for the same is presented in Algorithm 5.3:

```
Algorithm: Authentication Server
Input:
1. Mobile agent
Output:
Successful retrieval or storage of signature.
Repeat for every request from Authenticator
1. if signature_Collector( ) arrives for registration
    1.1 retrieve signatures from mobile agent
    1.2 store it into the database.
2. if signature_Collector( ) arrives for device signature
    2.1 retrieve signatures from database of Authentication Server
    2.2 load and dispatch the mobile agent signature_Collector( ) to
Authenticator.
```

Algorithm 5.3 Algorithm for authentication server

Bulk data transmission facility which is not provided in traditional EAP based approach has been incorporated into the proposed mobile agent based approach. Agents can be loaded with bulk data and can be dispatched to the destination. Steps for the same are presented in algorithm 5.4:

```
Algorithm 4: Data_Transmission
Input:
1. Data to be transmitted
2. Destination address (server or client)
Output: Success or failure of data transmission
1. Repeat while client or server has data to send
        1.1 create agent Load_Data( ).
        1.2 load agent with the requisite data and session_key.
    2. dispatch agent to the destination
```

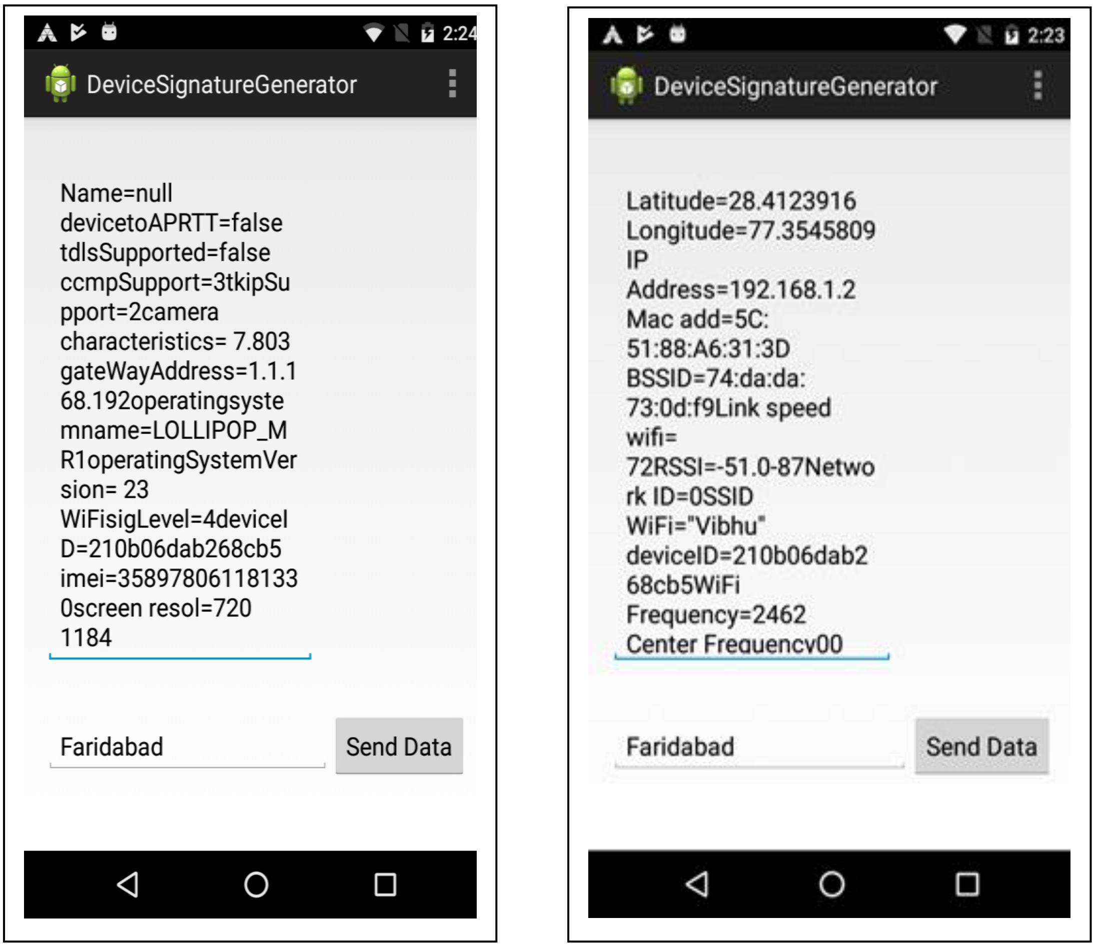
Algorithm 5.4 Algorithm for data transmission

5.3 IMPLEMENTATION AND ANALYSIS

Proposed device signature based authentication mechanism is implemented on Android platform. Mobile application extracts parameters of a device which helps in generation of signature of a device. This signature can be used in one form or another to authenticate the device on to the network. Figure 5.2 shows the mobile application screenshots used to capture the parameters needed to generate device signature.

Using this application, device signatures of multiple devices are collected as shown in Figure 5.3 and Figure 5.4.

These device signatures comprising of multiple device characteristics can be used to identify the device on the network in one form or another.



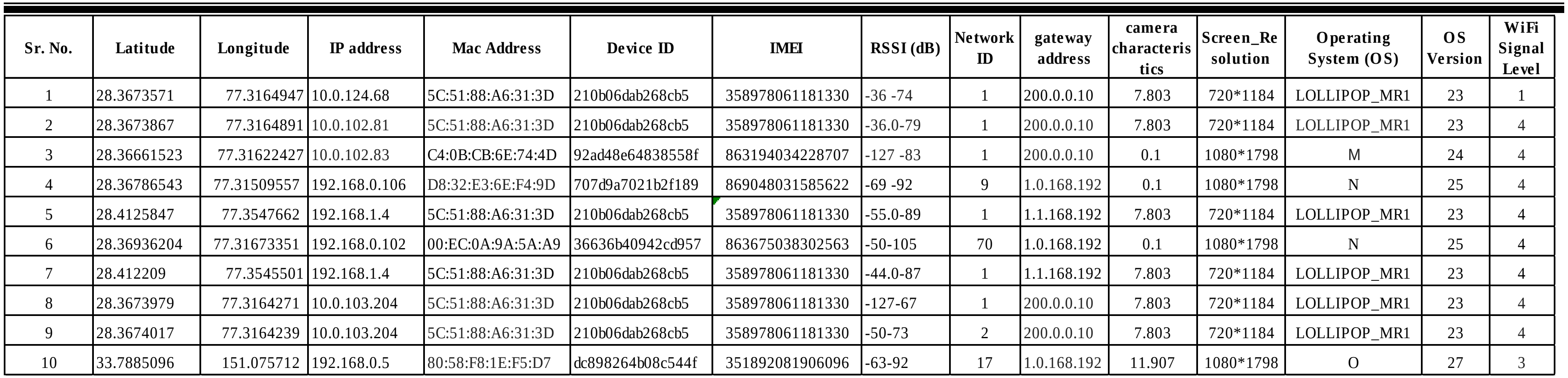
```
Name=null
devicetoAPRTT=false
tdlsSupported=false
ccmpSupport=3tkipSu
pport=2camera
characteristics= 7.803
gateWayAddress=1.1.1
68.192operatingsyste
mname=LOLLIPOP_M
R1operatingSystemVer
sion= 23
WiFisigLevel=4deviceI
D=210b06dab268cb5
imei=35897806118133
0screen resol=720
1184
```

Faridabad

```
Latitude=28.4123916
Longitude=77.3545809
IP
Address=192.168.1.2
Mac add=5C:
51:88:A6:31:3D
BSSID=74:da:da:
73:0d:f9Link speed
wifi=
72RSSI=-51.0-87Netwo
rk ID=0SSID
WiFi="Vibhu"
deviceID=210b06dab2
68cb5WiFi
Frequency=2462
Center Frequency00
```

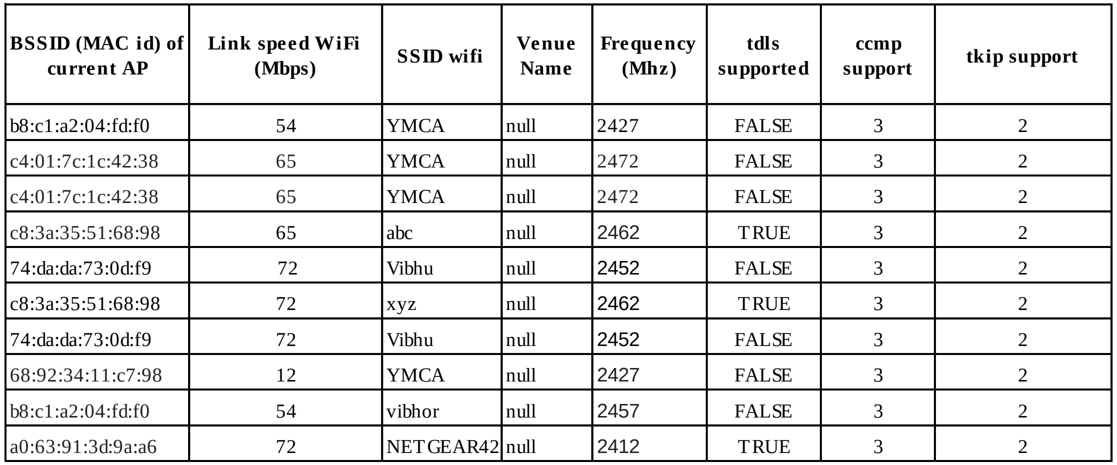
Faridabad

Figure 5.2: Application showing device signature parameters on Motorola device



Sr. No.	Latitude	Longitude	IP address	Mac Address	Device ID	IMEI	RSSI (dB)	Network ID	gateway address	camera characteristics	Screen_Resolution	Operating System (OS)	OS Version	WiFi Signal Level
1	28.3673571	77.3164947	10.0.124.68	5C:51:88:A6:31:3D	210b06dab268cb5	358978061181330	-36 -74	1	200.0.0.10	7.803	720*1184	LOLLIPOP_MR1	23	1
2	28.3673867	77.3164891	10.0.102.81	5C:51:88:A6:31:3D	210b06dab268cb5	358978061181330	-36.0-79	1	200.0.0.10	7.803	720*1184	LOLLIPOP_MR1	23	4
3	28.36661523	77.31622427	10.0.102.83	C4:0B:CB:6E:74:4D	92ad48e64838558f	863194034228707	-127 -83	1	200.0.0.10	0.1	1080*1798	M	24	4
4	28.36786543	77.31509557	192.168.0.106	D8:32:E3:6E:F4:9D	707d9a7021b2f189	869048031585622	-69 -92	9	1.0.168.192	0.1	1080*1798	N	25	4
5	28.4125847	77.3547662	192.168.1.4	5C:51:88:A6:31:3D	210b06dab268cb5	358978061181330	-55.0-89	1	1.1.168.192	7.803	720*1184	LOLLIPOP_MR1	23	4
6	28.36936204	77.31673351	192.168.0.102	00:EC:0A:9A:5A:A9	36636b40942cd957	863675038302563	-50-105	70	1.0.168.192	0.1	1080*1798	N	25	4
7	28.412209	77.3545501	192.168.1.4	5C:51:88:A6:31:3D	210b06dab268cb5	358978061181330	-44.0-87	1	1.1.168.192	7.803	720*1184	LOLLIPOP_MR1	23	4
8	28.3673979	77.3164271	10.0.103.204	5C:51:88:A6:31:3D	210b06dab268cb5	358978061181330	-127-67	1	200.0.0.10	7.803	720*1184	LOLLIPOP_MR1	23	4
9	28.3674017	77.3164239	10.0.103.204	5C:51:88:A6:31:3D	210b06dab268cb5	358978061181330	-50-73	2	200.0.0.10	7.803	720*1184	LOLLIPOP_MR1	23	4
10	33.7885096	151.075712	192.168.0.5	80:58:F8:1E:F5:D7	dc898264b08c544f	351892081906096	-63-92	17	1.0.168.192	11.907	1080*1798	O	27	3

Figure 5.3: Multiple Android Device Signatures (Device only)



BSSID (MAC id) of current AP	Link speed WiFi (Mbps)	SSID wifi	Venue Name	Frequency (Mhz)	tdls supported	ccmp support	tkip support
b8:c1:a2:04:fd:f0	54	YMCA	null	2427	FALSE	3	2
c4:01:7c:1c:42:38	65	YMCA	null	2472	FALSE	3	2
c4:01:7c:1c:42:38	65	YMCA	null	2472	FALSE	3	2
c8:3a:35:51:68:98	65	abc	null	2462	TRUE	3	2
74:da:da:73:0d:f9	72	Vibhu	null	2452	FALSE	3	2
c8:3a:35:51:68:98	72	xyz	null	2462	TRUE	3	2
74:da:da:73:0d:f9	72	Vibhu	null	2452	FALSE	3	2
68:92:34:11:c7:98	12	YMCA	null	2427	FALSE	3	2
b8:c1:a2:04:fd:f0	54	vibhor	null	2457	FALSE	3	2
a0:63:91:3d:9a:a6	72	NETGEAR42	null	2412	TRUE	3	2

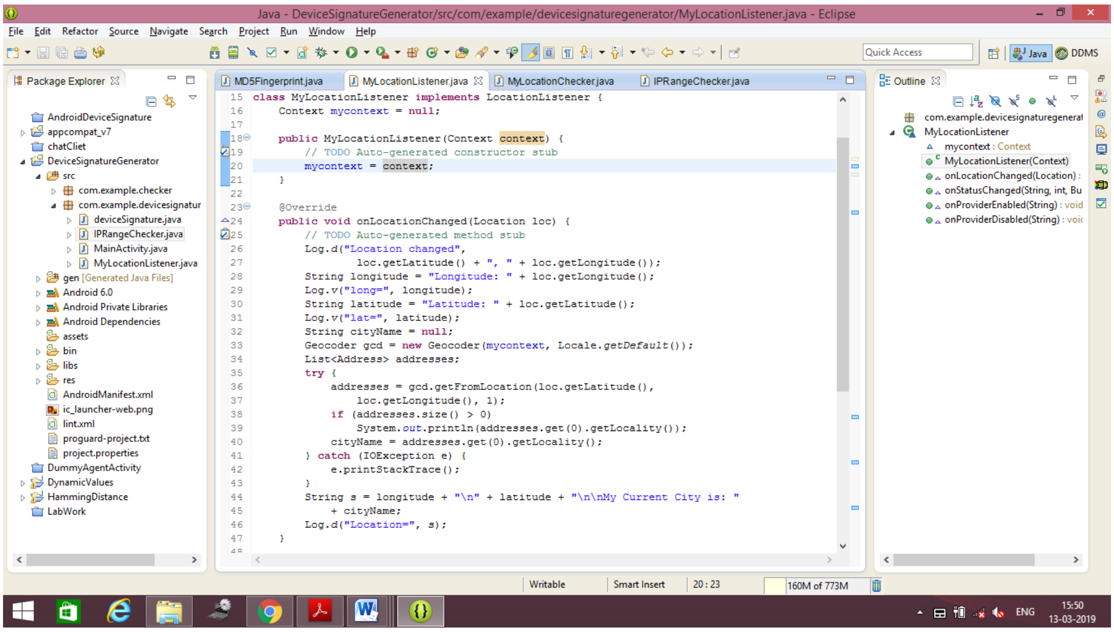
Figure 5.4: Multiple Android Device Signatures (Access Point)

5.3.1 AUTHENTICATION USING DEVICE SIGNATURE

Captured device signature parameters can be used in many forms for authentication in addition to the currently existing authentication protocols like EAP.

Some of the parameters usages are:

- ***Location***: Figure 5.5 shows the MyLocationListener class which is used to capture the current location of the mobile device. Location captured using this class can be used to check that whether a device is within the range of the Access Point or not.

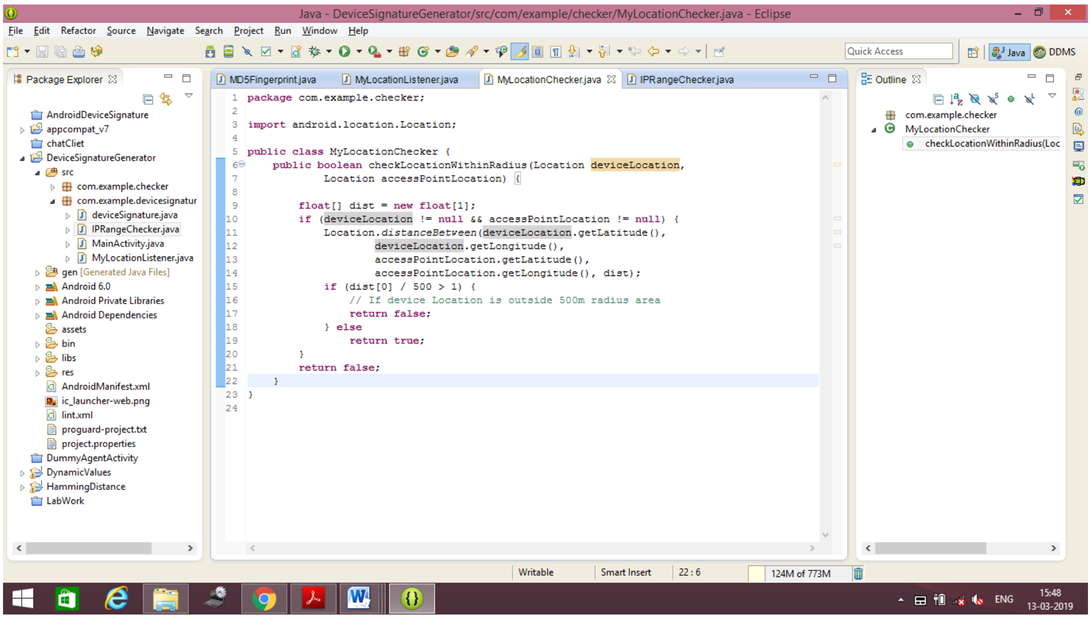


```
class MyLocationListener implements LocationListener {
    Context mycontext = null;

    public MyLocationListener(Context context) {
        // TODO Auto-generated constructor stub
        mycontext = context;
    }

    @Override
    public void onLocationChanged(Location loc) {
        // TODO Auto-generated method stub
        Log.d("Location changed",
                loc.getLatitude() + ", " + loc.getLongitude());
        String longitude = "Longitude: " + loc.getLongitude();
        Log.v("long=", longitude);
        String latitude = "Latitude: " + loc.getLatitude();
        Log.v("lat=", latitude);
        String cityName = null;
        Geocoder gcd = new Geocoder(mycontext, Locale.getDefault());
        List<Address> addresses;
        try {
            addresses = gcd.getFromLocation(loc.getLatitude(),
                loc.getLongitude(), 1);
            if (addresses.size() > 0)
                System.out.println(addresses.get(0).getLocality());
            cityName = addresses.get(0).getLocality();
        } catch (IOException e) {
            e.printStackTrace();
        }
        String s = longitude + "\n" + latitude + "\n\nMy Current City is: "
            + cityName;
        Log.d("Location=", s);
    }
```

Figure 5.5: Location Listener

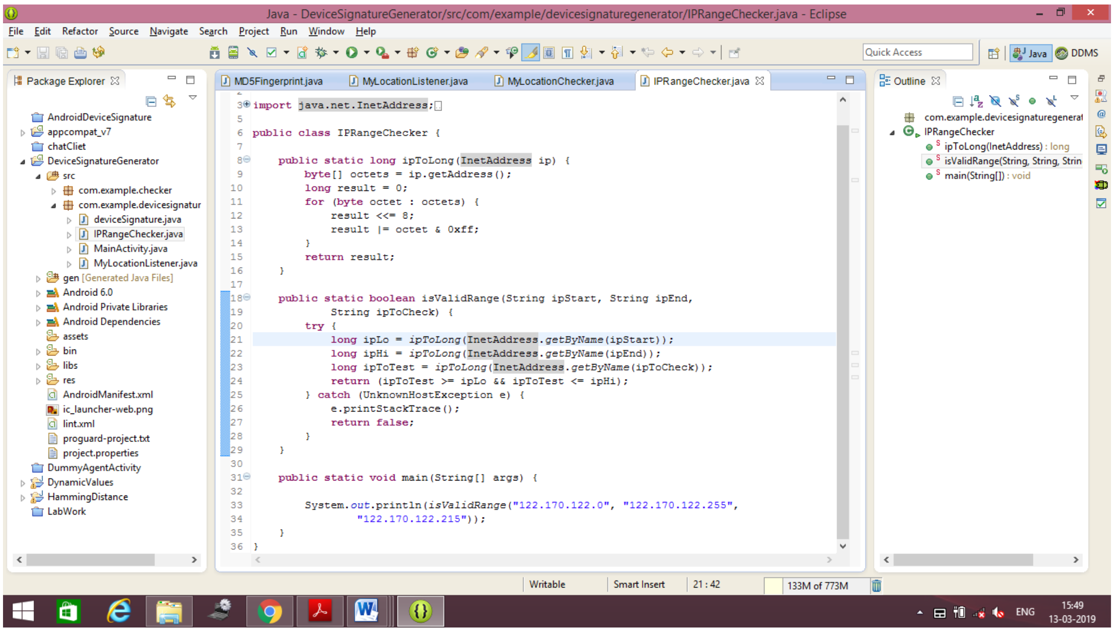
Figure 5.6 shows the snapshot of MyLocationChecker class which can be used to compare the location of the device and already stored location of the Access Point. This class compares the current device location that whether it is within the 500 meter radius of the access point or not. This check can be very useful for checking an unauthorized person or device pretending to be a valid one.

- ***IP Address*:** IP address of the device can also be one of the parameter for authentication. Whenever a network is created, a range of IP addresses are assigned to it. With the help of device IP address algorithm can check that whether or not the device IP address is within the valid range or not. If it is not with in the valid range then mobile device can be blocked. Class is shown in Figure 5.7.


Java - DeviceSignatureGenerator/src/com/example/checker/MyLocationChecker.java - Eclipse
File Edit Refactor Source Navigate Search Project Run Window Help
Package Explorer
MD5Fingerprint.java
MyLocationListener.java
MyLocationChecker.java
IPRangeChecker.java
Outline
Writable
Smart Insert
22 : 6
124M of 773M

```
package com.example.checker;

import android.location.Location;

public class MyLocationChecker {
    public boolean checkLocationWithinRadius(Location deviceLocation,
            Location accessPointLocation) {

        float[] dist = new float[1];
        if (deviceLocation != null && accessPointLocation != null) {
            Location.distanceBetween(deviceLocation.getLatitude(),
                    deviceLocation.getLongitude(),
                    accessPointLocation.getLatitude(),
                    accessPointLocation.getLongitude(), dist);
            if (dist[0] / 500 > 1) {
                // If device Location is outside 500m radius area
                return false;
            } else
                return true;
        }
        return false;
    }
}
```

Figure 5.6: MyLocationChecker class


Java - DeviceSignatureGenerator/src/com/example/devicesignaturegenerator/IPRangeChecker.java - Eclipse
File Edit Refactor Source Navigate Search Project Run Window Help
Package Explorer
MD5Fingerprint.java
MyLocationListener.java
MyLocationChecker.java
IPRangeChecker.java
Outline
Writable
Smart Insert
21 : 42
133M of 773M

```
import java.net.InetAddress;

public class IPRangeChecker {

    public static long ipToLong(InetAddress ip) {
        byte[] octets = ip.getAddress();
        long result = 0;
        for (byte octet : octets) {
            result <<= 8;
            result |= octet & 0xff;
        }
        return result;
    }

    public static boolean isValidRange(String ipStart, String ipEnd,
            String ipToCheck) {
        try {
            long ipLo = ipToLong(InetAddress.getByName(ipStart));
            long ipHi = ipToLong(InetAddress.getByName(ipEnd));
            long ipToTest = ipToLong(InetAddress.getByName(ipToCheck));
            return (ipToTest >= ipLo && ipToTest <= ipHi);
        } catch (UnknownHostException e) {
            e.printStackTrace();
            return false;
        }
    }

    public static void main(String[] args) {

        System.out.println(isValidRange("122.170.122.0", "122.170.122.255",
                "122.170.122.215"));
    }
}
```

Figure 5.7: IPRangeChecker Class

- ***MD5 Fingerprint*:** From the device signature static and dynamic values are extracted as shown in Table 5.1 and Table 5.2. Static values are the values which are not going to change throughout the life of the device. These static values of a device are taken to produce MD5 Fingerprint of the device.

Figure 5.8 shows the MD5Fingerprint class used to generate the fingerprint using the static parameters of the device. Similarly, static signature values for AP can be used as shown in Table 5.3 to produce the MD5 Fingerprint of the access point. This fingerprint for Access Point can be used to capture the fake access point into the network.

Table 5.1: Static parameters of device

Sr. No.	Static Parameters	
1	Mac Address	**Device**
2	Device ID	
3	IMEI	
4	Camera Characteristics	
5	Screen Resolution	
6	Operating System	
7	Operating System Version	

Table 5.2: Dynamic parameters of device

Sr. No.	Dynamic Parameters	
1	Latitude	**Device**
2	Longitude	
3	IP address	

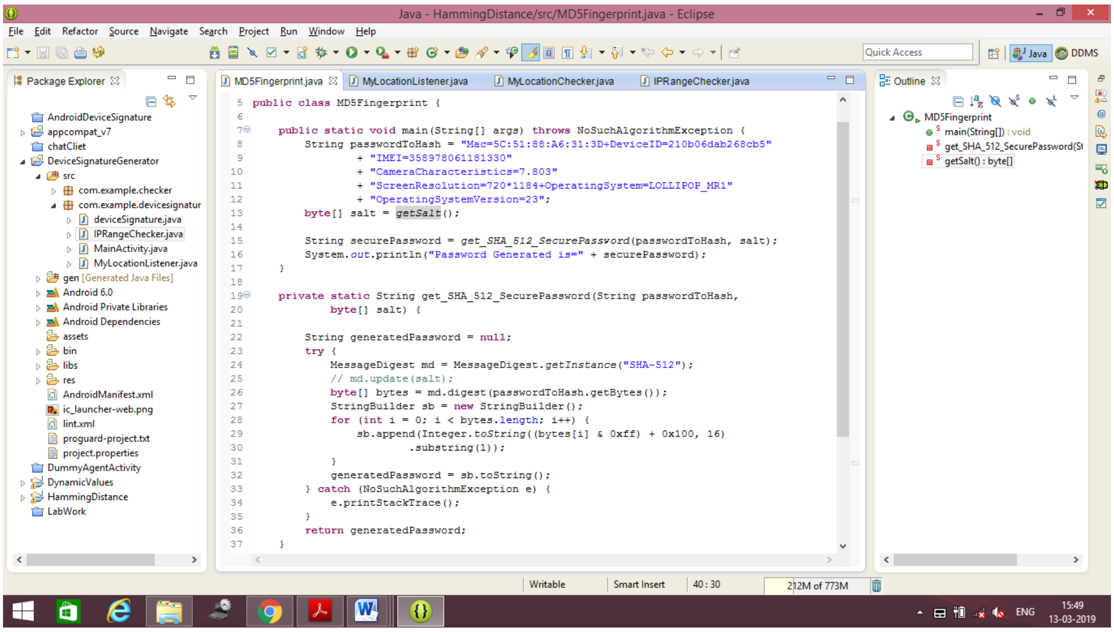
Table 5.3: Static parameters of access point

Sr. No.	Static Parameters	
1	BSSID	**AP**
2	SSID	
3	Frequency	
4	tdls supported	
5	ccmp support	
6	tkip support	

Advantage of creating MD5 Fingerprint is that, it is possible to create the MD5 Fingerprint from static device or access point parameters but reverse is not possible i.e. device or access point parameters cannot be extracted from the MD5 Fingerprint.

Apart from the MD5 Fingerprint, Hamming distance can be used to match and compare two mobile device signatures.

- ***Hamming Distance*:** Hamming distance of two device signatures also can be one of the mechanism for identification of mobile device. Distance between two signatures is measured using the hamming distance formula.



```java
public class MD5Fingerprint {

    public static void main(String[] args) throws NoSuchAlgorithmException {
        String passwordToHash = "Mac=5C:51:88:A6:31:3D+DeviceID=210b06dab268cb5"
                + "IMEI=358978061181330"
                + "CameraCharacteristics=7.803"
                + "ScreenResolution=720*1184+OperatingSystem=LOLLIPOP_MR1"
                + "OperatingSystemVersion=23";
        byte[] salt = getSalt();

        String securePassword = get_SHA_512_SecurePassword(passwordToHash, salt);
        System.out.println("Password Generated is=" + securePassword);
    }

    private static String get_SHA_512_SecurePassword(String passwordToHash,
            byte[] salt) {

        String generatedPassword = null;
        try {
            MessageDigest md = MessageDigest.getInstance("SHA-512");
            // md.update(salt);
            byte[] bytes = md.digest(passwordToHash.getBytes());
            StringBuilder sb = new StringBuilder();
            for (int i = 0; i < bytes.length; i++) {
                sb.append(Integer.toString((bytes[i] & 0xff) + 0x100, 16)
                        .substring(1));
            }
            generatedPassword = sb.toString();
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        return generatedPassword;
    }
```

Figure 5.8: MD5Fingerprint Class

Hamming distance gives minimum number that is required for a string to be similar with other. For device signature matching following hamming distance based formula is used as shown in equation 5.1.

$$d(x,y) = \frac{\sum_{i=1}^{F}(x_i != y_i)}{F}, \qquad 0 \le d(x,y) \le 1 \qquad \dots(5.1)$$

Where, F is the number of features and x_i and y_i represents ith feature of the device signature stored and extracted respectively. If x_i equals y_i, then it contributes 1 to the summation.

If the device signature parameters are different, only then they will contribute to distance. In the Table 1, there are 21 device signature parameters.

Let’s compare two signatures of same device using the equation 5.1.

$$x = \{x_1, x_2, x_3, \dots..x_n\}$$

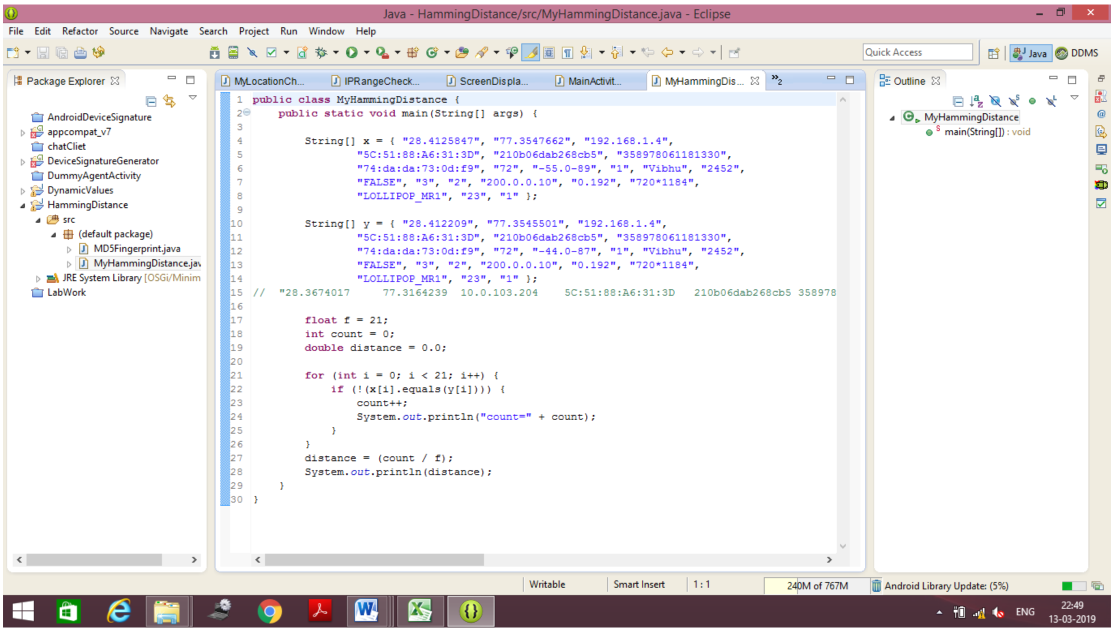
$$y = \{y_1, y_2, y_3, \ldots\ldots y_n\}$$

Where x represents device signature stored in the database during registration and y represents the signature supplied by the mobile agent when it tries to connect to the network.

x = {28.4125847, 77.3547662, 192.168.1.4, 5C:51:88:A6:31:3D, 210b06dab268cb5, 358978061181330, 74:da:da:73:0d:f9, 72, -55.0-89, 1,Vibhu, 2452, FALSE, 3, 2, 200.0.0.10, 0.192, 720*1184, LOLLIPOP_MR1, 23, 1}

y = {28.412209, 77.3545501, 192.168.1.4, 5C:51:88:A6:31:3D, 210b06dab268cb5, 358978061181330, 74:da:da:73:0d:f9, 72, -44.0-87, 1, Vibhu, 2452, FALSE, 3, 2, 200.0.0.10, 0.192, 720*1184, LOLLIPOP_MR1, 23, 1}

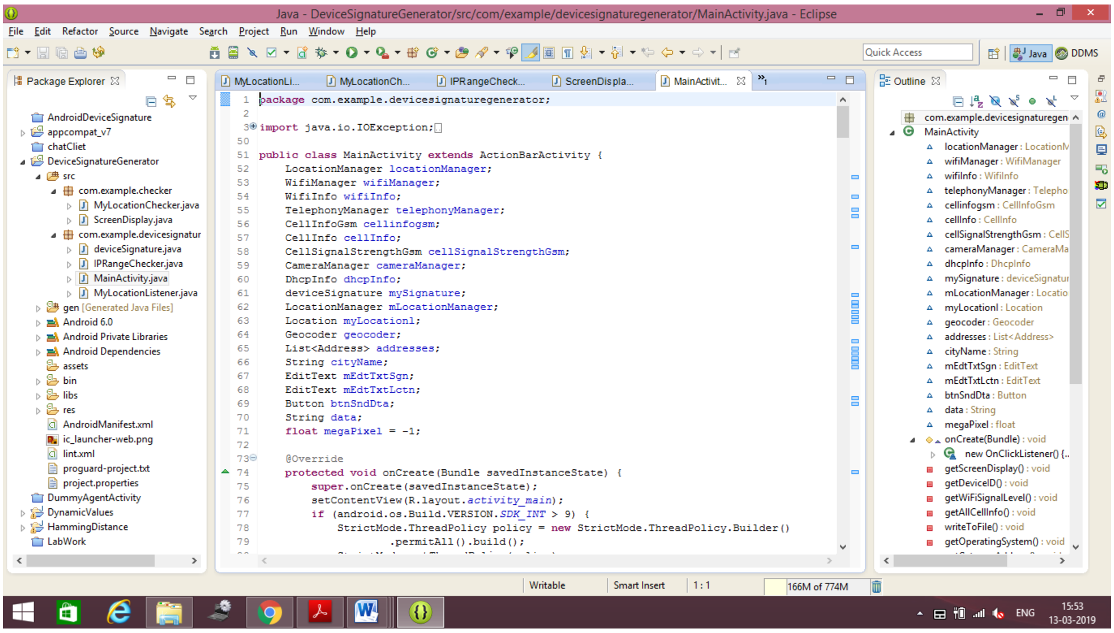
Using the equation 5.1 hamming distance of the signatures x and y is calculated. The Java code for the same is shown below in Figure 5.9. Output of the same i.e. the hamming distance is 0.1428571492433548. This hamming distance tells about the mismatch of the two signatures. Larger the hamming distance value, more the mismatch is. This hamming distance can be used to identify the signature of mobile device.



```java
public class MyHammingDistance {
    public static void main(String[] args) {

        String[] x = { "28.4125847", "77.3547662", "192.168.1.4",
                "5C:51:88:A6:31:3D", "210b06dab268cb5", "358978061181330",
                "74:da:da:73:0d:f9", "72", "-55.0-89", "1", "Vibhu", "2452",
                "FALSE", "3", "2", "200.0.0.10", "0.192", "720*1184",
                "LOLLIPOP_MR1", "23", "1" };

        String[] y = { "28.412209", "77.3545501", "192.168.1.4",
                "5C:51:88:A6:31:3D", "210b06dab268cb5", "358978061181330",
                "74:da:da:73:0d:f9", "72", "-44.0-87", "1", "Vibhu", "2452",
                "FALSE", "3", "2", "200.0.0.10", "0.192", "720*1184",
                "LOLLIPOP_MR1", "23", "1" };
//  "28.3674017     77.3164239  10.0.103.204    5C:51:88:A6:31:3D   210b06dab268cb5 358978

        float f = 21;
        int count = 0;
        double distance = 0.0;

        for (int i = 0; i < 21; i++) {
            if (!(x[i].equals(y[i]))) {
                count++;
                System.out.println("count=" + count);
            }
        }
        distance = (count / f);
        System.out.println(distance);
    }
}
```

Figure 5.9: Calculation of hamming distance

Figure 5.10 shows the MainActivity of the Android based mobile application.



```
package com.example.devicesignaturegenerator;

import java.io.IOException;

public class MainActivity extends ActionBarActivity {
    LocationManager locationManager;
    WifiManager wifiManager;
    WifiInfo wifiInfo;
    TelephonyManager telephonyManager;
    CellInfoGsm cellinfogsm;
    CellInfo cellInfo;
    CellSignalStrengthGsm cellSignalStrengthGsm;
    CameraManager cameraManager;
    DhcpInfo dhcpInfo;
    deviceSignature mySignature;
    LocationManager mLocationManager;
    Location myLocationl;
    Geocoder geocoder;
    List<Address> addresses;
    String cityName;
    EditText mEdtTxtSgn;
    EditText mEdtTxtLctn;
    Button btnSndDta;
    String data;
    float megaPixel = -1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        if (android.os.Build.VERSION.SDK_INT > 9) {
            StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder()
                    .permitAll().build();
```

Java - DeviceSignatureGenerator/src/com/example/devicesignaturegenerator/MainActivity.java - Eclipse
File Edit Refactor Source Navigate Search Project Run Window Help
Package Explorer
Outline
MainActivity
Writable
Smart Insert
1 : 1
166M of 774M

Figure 5.10: MainActivity of the Mobile Application

Figure 5.11 shows the method to run the application on the Android mobile device connected to the development environment.

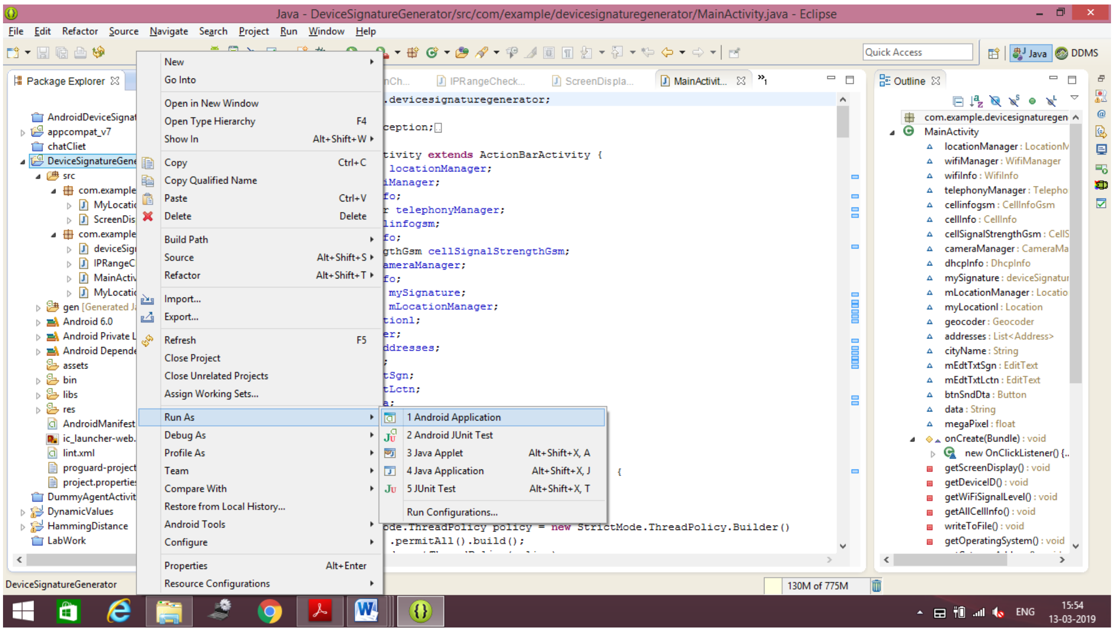


Figure 5.11: Application Run on Android Device

5.4 SUMMARY

In this chapter device fingerprint and agent based device authentication mechanism has been proposed. Mechanism uses various parameters for device signature calculation. Proposed device signature based authentication mechanism includes device fingerprint generation and comparison with the stored signature. Signature can be used in many ways including location checking of the device with the stored Access Point location. IP address of the device can be checked with the range of IP addresses available in to the network. Static parameters of the signature can be used to produce MD5 Fingerprint of the device which can be matched with the stored MD5 Fingerprint for authentication. Benefit of using MD5 approach is that it is possible to produce the signature from the static parameters but reverse is not possible i.e. device parameters cannot be generated from the MD5 Fingerprint.

CHAPTER 6

BIG DATA PROCESSING AND AUTHENTICATION OF DEVICES USING MOBILE AGENT

6.1 INTRODUCTION

Device Fingerprint based authentication mechanism described in the previous chapter can also be used in BigData processing which includes multiple distributed nodes for processing of large scale data. BigData is the term used to refer the data set which is captured, generated and processed at a tremendous rate. So, BigData refers to the data set which cannot be handled using the conventional **R**elational **D**atabase **M**anagement **S**ystem (RDBMS). In case of BigData processing input data is relatively very large and is distributed along multiple machines. The processing of this BigData involves hundreds or even thousands of machines which works in parallel to perform the task in a reasonable amount of time. This chapter introduces a new **A**gent **B**ased **M**ap**R**educe (ABMR) technique for processing BigData using MapReduce network. ABMR uses mobile agent for device registration and data transmission. Node which underperforms as compared to other participating nodes is known as straggler. The advantage of using mobile agent in this scenario is that if a machine becomes straggler then it is very easy to relocate the task to different machine. Mobile agents can resume its processing from the same point where it left. Proposed work is also compared with the Hadoop native scheduler against number of stragglers and different split sizes of the input file.

6.2 MOBILE AGENTS BASED MAPREDUCE (ABMR) FOR BIGDATA PROCESSING

Proposed ABMR algorithm makes use of mobile agent for task splitting and distribution. All the devices present in the network registers at the scheduler with the help of device signature. Device Signature will help in preventing unauthorized node to intentionally become straggler. Each mobile agent performs its task in a distributed manner and responds to the scheduler in the form of **P**erformance **S**core (P_{score}).

P_{score} will tell about the progress of the task which is assigned to the device. Scheduler calculates the average P_{score} and maintains a list of slower machines and faster machines.

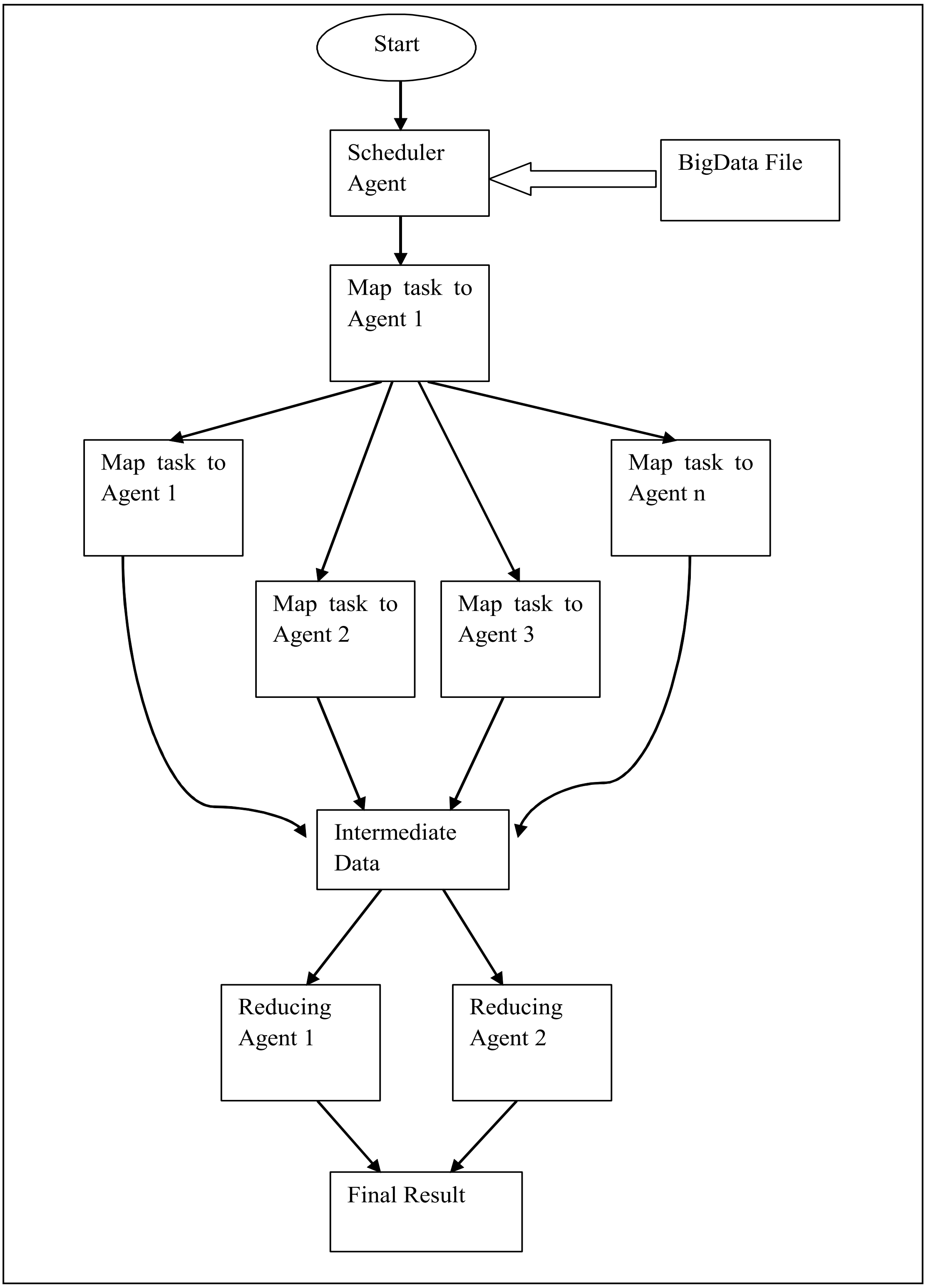


Figure 6.1: Flow Chart for the Proposed Solution

Scheduler analyses the P_{score} of each mobile agent and reschedules the slower machine mobile agent to faster machine. After each mobile agent has finished its job, scheduler recollects the result and forwards it to the original user. Figure 6.1 shows the complete flow chart of MapReduce technique used.

6.2.1 MOBILE AGENT BASED MAPREDUCE (ABMR) FOR BIGDATA PROCESSING

Some of the important assumptions in the algorithm that have been made to clearly specify the applicability of the technique are as follows:

- Each of the machines should have JADE [41]; the agent execution environment, which provides efficient mobile agent development and execution environment for parallel data processing. In the execution environment, a mobile agent can hide its code, data integrity and its owner identification.
- The Big Data file should be splitted evenly so that each agent gets an equal share of the load, and hence the P_{score} computed is even in all.
- Each MA gets only a single split of the file, i.e., the number of agents are equal to the number of the splits of the task in the network.
- An intermediate location will be defined to store the intermediate data from the various MA's.
- Scheduler is having BigData file which is then used for MapReduce

Methodology: During the literature survey, it was found that the methods were using some kind of heuristics for straggler detection and in parallel executing them at two different locations at the same time in order to mitigate their effect. Proposed solution, for processing BigData is using a scheduler agent, which supports mobile agent creation and rescheduling of MA's based on threshold value of P_{score}. Threshold is calculated based on the average P_{score}. Algorithm works in following phases:

- ***Registration Phase:*** Every device, who wants to be the part of the computing grid, should register itself with the main host where the application has been running. This registration process is carried out for a fixed-time period. Each node sends its IP Address and device signature to the main application so that it can further communicate with these nodes. These IP addresses are stored in the array and a corresponding mobile agent container is launched in the JADE runtime environment for each of the mobile agents. This agent container controls all operations to be performed by a particular mobile agent. Each node registering itself admits that it meets all the required conditions and is ready to offer its services to the parallel processing network.

- ***File Fetching and Scheduler Agent Creation:*** In this phase, the address of the Big Data file is fetched, and the scheduler agent is created along with the file address. Array of agent containers is also maintained. The scheduler then contacts all the containers which are used for the creation of agents. The scheduler in the meantime splits the file among multiple pieces based on the size of the split, which it takes from the user.

- ***Agent Score Calculation and Submission:*** Each agent calculates the score based on the number of words processed in unit time. Formula is given in equation 6.1.

$$\boxed{P_{score} = \left(\frac{no.of\ words\ processed}{total\ no\ of\ words\ to\ be\ processed}\right) \qquad \ldots\ (6.1)}$$

Unit time to process single word will be assumed as a constant. Each agent submits its partially calculated score to the scheduler.

- ***Agent Relocation Phase:*** Slow agents, which may be stragglers for the execution to complete, are identified using the performance score. Slow agents (straggler), are relocated to the new location based on the performance score. Scheduler takes the decision about relocation and message is sent to the slow agent for relocation.

- ***Agent Result Submission Phase:*** In this phase mobile agent submits result to the application, which then merges all the results into main result and then directs the final result to the user. The final result is in the form of a file, which contains the overall result. Application then sends the result back to the user.

6.2.2 INFORMATION FLOW DIAGRAM

Figure 6.2 shows the detailed information flow diagram of the proposed algorithm. In step 1 all devices on the network registers itself to the scheduler with their device signatures. In step 2, user supplies the BigData file which is to be processed to the main host. This file is used by the scheduler for processing. In step 3, scheduler splits the file according to the algorithm. Scheduler creates mobile agents and loads them with each split of the user file and dispatches it to the different machines.

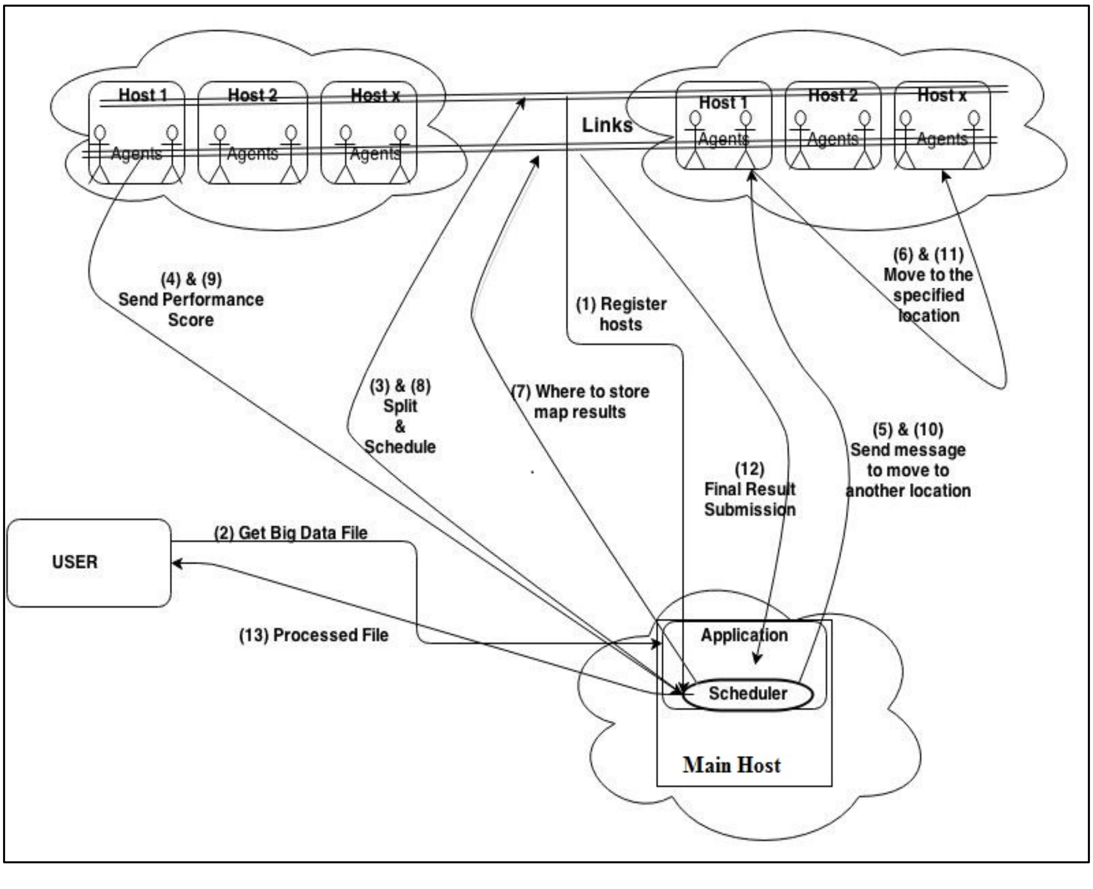

Host 1
Host 2
Host x
Agents
Agents
Agents
Links
Host 1
Host 2
Host x
Agents
Agents
Agents
(4) & (9) Send Performance Score
(3) & (8) Split & Schedule
(7) Where to store map results
(1) Register hosts
(6) & (11) Move to the specified location
(5) & (10) Send message to move to another location
(12) Final Result Submission
USER
(2) Get Big Data File
(13) Processed File
Application
Scheduler
Main Host

Figure 6.2: Information flow ABMR

These different machines in step 4 sends the P_{score} to the scheduler. Depending upon P_{score} scheduler reschedules the mobile agents running on slower machines to the faster machine and sends the message to the mobile agent about the same in step 5. In step 6 agent moves to a new location as scheduled by the scheduler. Steps from 4 to 11 are repeated until final results are available. In step 7, scheduler directs nodes about where to map results. In step 12, mobile agent submits final result information to the scheduler. As soon as scheduler has receives the final result, it is forwarded to the user.

6.2.3 ALGORITHM AND FLOW DIAGRAM

(a) Algorithm for Scheduler

1. A node which wants to participate on the network registers at the scheduler with their respective device signature.
2. Multiple containers are created using the IP address of the multiple hosts.
3. A Scheduler agent is created in by the main application which takes file name as argument and finds all the agent containers running on the home platform.
4. Scheduler agent splits the file and creates multiple mobile agents and assigns a split to each agent.
5. Each agent sends its performance score to the scheduler after every ‘s’ seconds.

$$score_{agent[i]} = \left(\frac{Total\_executed}{Total\_to\_be\_executed}\right) * s \qquad \dots (6.2)$$

6. Average score of all the containers is calculated.
7. Compute slow_containers and fast_containers depending upon the average score.
8. Agents from slow_containers are moved to fast_containers.
9. Meanwhile, when the agent moves from one location to another, it saves its current state so that they can start from the very same place where they stopped the execution.
10. After performing all the operations, the agent submits the result back to the scheduler.
11. Accumulation of results from all the agents is done by scheduler and presents it to the user.

Algorithm for scheduler module at main host is shown in Algorithm 6.1.

Algorithm: Scheduler Module (at the main host)

Input: file, container

Output: processed file to user

```
calculate ratio as
  ratio=file.length/container.length
initialize number_of_agents = 0 and score[ ]=0
for i=1, …, container_length do
    for  j=1, ..., ratio
        container[i].createAgent(file[j],container[i],ratio)
        number_of_agents = number_of_agents+1
    end for
end for
for k=1, …, container.length do
    score[k]= (receive(msg[k]))        //score of each container
    totalscore = totalscore + score[k]
end for
average_score = totalscore/number_of_agents
i=0
for m=1, …, container.length do
    if (score[m]<average_score)
        slowContainers[i++]=container[m]
    else
        fastContainers[i++]=container[m]
    end if
end for
for n=1, …, slowContainers.length
    agent[n].move(fastContainer[n]);
end for
```

Algorithm 6.1 Algorithm for scheduler agent at the main host

6.3 IMPLEMENTATION AND RESULT ANALYSIS

The implementation of proposed ABMR scheme is done in java JDK 1.7 using JADE Agent Development Environment for development of agents. An Intel i5 laptop with 6GB RAM is used for the execution of the program for both scenarios with a 450MB text file. The Map and Reduce operations are defined for word count procedures. JADE is used for creating mobile agents on various machines. **J**ava **A**gent **De**velopment Framework (JADE) is framework to implement mobile agents. This framework is developed in java language. This framework provides a middleware, with the help of that agents can easily be developed and deployed. This framework follows the **F**oundation for **I**ntelligent **P**hysical **A**gents (FIPA) guidelines. FIPA is the organization that helps in providing development guidelines and also helps in promoting the mobile agent technology. JADE provides a GUI based environment where agents can be easily developed and migrated to different machine.

The complete code for the technique is divided into three modules. First, for creating the mobile agents, second for mobile agent itself and third for scheduling mobile agents. All these 3 modules are called and controlled by one main module. The main module is also responsible for user inputs, graph reading from files and splitting the files. After the work assignment, every registered device calculates its performance score according to equation 6.2. Some of the important methods in the implementation are:

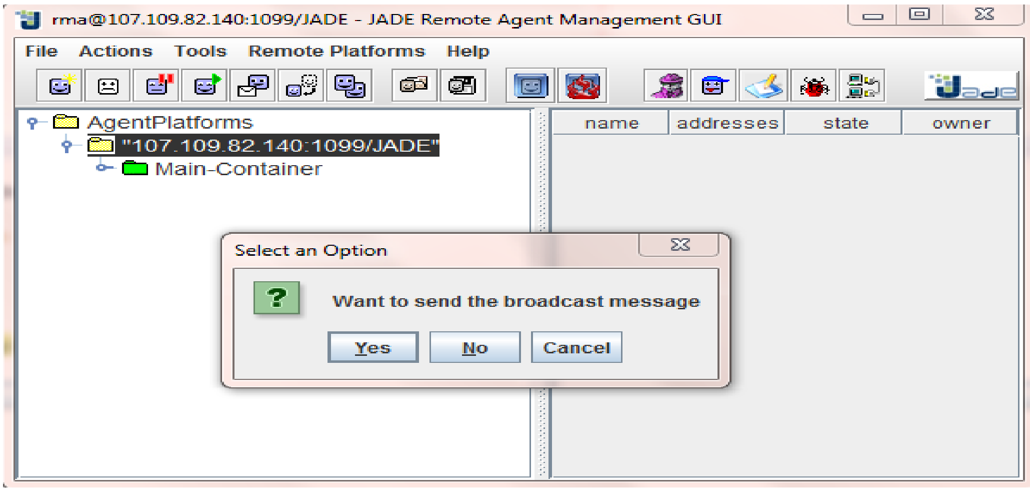
- ***send_message(Location_of_Scheduler):*** This method sends the performance score to the location provided to it which is fixed for a task i.e. the location of the scheduler agent. This method is defined in the MainAgent class and travels every location the agent visits.
- ***calc_Location(agent,faster_Array,current_location):*** This method finds the location where the agent from its current location will be relocated to available faster locations. Here *agent* is the mobile agent for which scheduler calls this function, *faster_Array* contains the locations for the faster machines and *current_location* is the mobile *agent*'s current location.

- ***kill(Location_Scheduler):*** This is a method that kills the mobile agent if the *attribute_list* contains null which is set when the map/reduce operations are finished. Otherwise it sends the agent to the scheduler agent at the main host for handling the incomplete task.

- ***move(location):*** This method is defined in the *MainAgent* and is called whenever the scheduler communicates a new *location* for mobile agent to move. This method saves the current state of the process along with its attributes and continues execution on the next machine where it has to move.

- ***schedule(agents_Array, location_Array):*** This method initially schedules all available mobile agents in the *agent_Array* to any of the location available in the location_Array. All locations are loaded equally. While rescheduling the agents, the *schedule* function takes care of the performance score sent to the scheduler and if the score is less than half of the average score then a new location is searched for this mobile agent.

- ***registerMe(location):*** This method registers various containers to the main module of the application. These locations are stored in the *location_Array.*



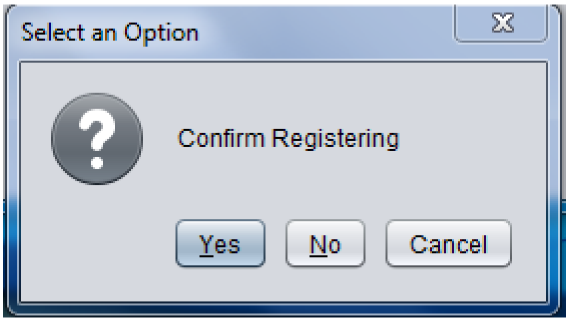
Want to send the broadcast message

Figure 6.3: Broadcast message to all host

In the implementation, firstly the main host is created. Main host will act as the server where the scheduler agent is created. Main host will broadcast the message to all nodes who wants to be the part of distributed data processing. Figure 6.3 shows the snapshot where main host initiates the broadcast message.

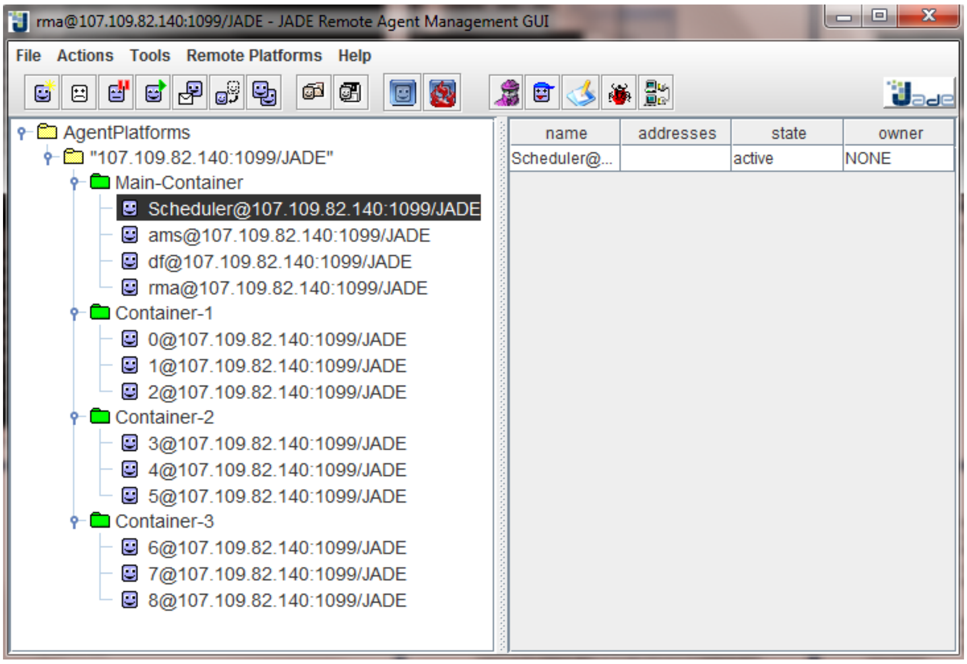
As soon as the broadcast message is initiated by the main host, all hosts will receive a prompt message as shown in Figure 6.4. All users who wants to participate will respond with the Yes message otherwise No message.

Figure 6.5 shows the main host after multiple containers are created and each container contains multiple agents loaded with file split and task to be performed. Each mobile agent will perform the task and will respond to the scheduler after certain time with the performance score. Scheduler will receive the performance score and will create the slow containers and fast containers list which will be updated periodically.



Confirm Registering

Figure 6.4: Confirm Registration

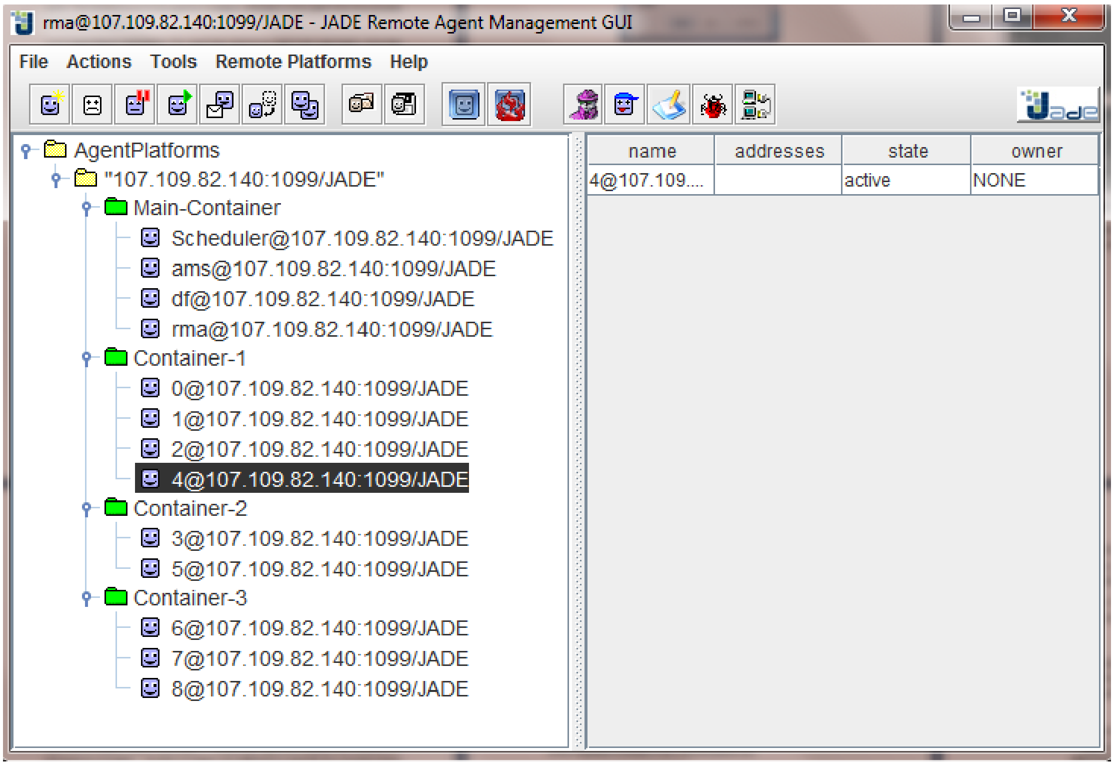

rma@107.109.82.140:1099/JADE - JADE Remote Agent Management GUI
File Actions Tools Remote Platforms Help

```
AgentPlatforms
  "107.109.82.140:1099/JADE"
    Main-Container
      Scheduler@107.109.82.140:1099/JADE
      ams@107.109.82.140:1099/JADE
      df@107.109.82.140:1099/JADE
      rma@107.109.82.140:1099/JADE
    Container-1
      0@107.109.82.140:1099/JADE
      1@107.109.82.140:1099/JADE
      2@107.109.82.140:1099/JADE
    Container-2
      3@107.109.82.140:1099/JADE
      4@107.109.82.140:1099/JADE
      5@107.109.82.140:1099/JADE
    Container-3
      6@107.109.82.140:1099/JADE
      7@107.109.82.140:1099/JADE
      8@107.109.82.140:1099/JADE
```

name	addresses	state	owner
Scheduler@...		active	NONE

Figure 6.5: Main host with multiple containers

Scheduler will identify the straggler with the help of performance score and will send Relocate message to the mobile agent from slower machine to the faster machine. Figure 6.6 shows the relocation process. Here the mobile agent number 4 is moved from slower container number 2 to faster container number 1.


rma@107.109.82.140:1099/JADE - JADE Remote Agent Management GUI
File Actions Tools Remote Platforms Help

```
AgentPlatforms
  "107.109.82.140:1099/JADE"
    Main-Container
      Scheduler@107.109.82.140:1099/JADE
      ams@107.109.82.140:1099/JADE
      df@107.109.82.140:1099/JADE
      rma@107.109.82.140:1099/JADE
    Container-1
      0@107.109.82.140:1099/JADE
      1@107.109.82.140:1099/JADE
      2@107.109.82.140:1099/JADE
      4@107.109.82.140:1099/JADE
    Container-2
      3@107.109.82.140:1099/JADE
      5@107.109.82.140:1099/JADE
    Container-3
      6@107.109.82.140:1099/JADE
      7@107.109.82.140:1099/JADE
      8@107.109.82.140:1099/JADE
```

name	addresses	state	owner
4@107.109....		active	NONE

Figure 6.6: Relocation Process

6.3.1 TESTING

The proposed algorithm is tested on two scenarios i.e. for different split sizes and for different number of stragglers. This section describes results collected by experiments and provides the comparison with results of HADOOP native scheduler. The implementation of the proposed algorithm is done on an Intel i5 laptop with 6 GB RAM and 450 MB text file. The Map and Reduce operations are defined for word count procedures. For implementation, Java Agent Development Framework (JADE) is used. Parameter values used for the algorithm are shown in Table 6.1.

Table 6.1: Parameter values

Parameter	**Value**
Size of Big Data file	450 MB
Agent containers per main container	2-3
Agents per agent container	3-10
Mapping Technique	Word Count

The proposed algorithm is compared with HADOOP Native Scheduler and results are shown in Table 6.2. This shows that the proposed technique gives good results when number of straggler percentage increases in to the system. It simply saves the extra time HADOOP takes to look for the HADOOP Distributed File System (DFS) for MapReduce.

Table 6.2: Execution overview on straggler %

Straggler %	**HADOOP Native Scheduler Time(ms)**	**Mobile Agent Based MapReduce Time(ms)**
10%	48587	51237
20%	58548	54893
40%	87373	74120
50%	120252	99365

The proposed algorithm is compared with HADOOP Native Scheduler for its straggler detection and mitigation technique also. Figure 6.7 clearly shows that the proposed technique outperforms HADOOP's technique in execution time for randomly generated graphs for different straggler percentage.

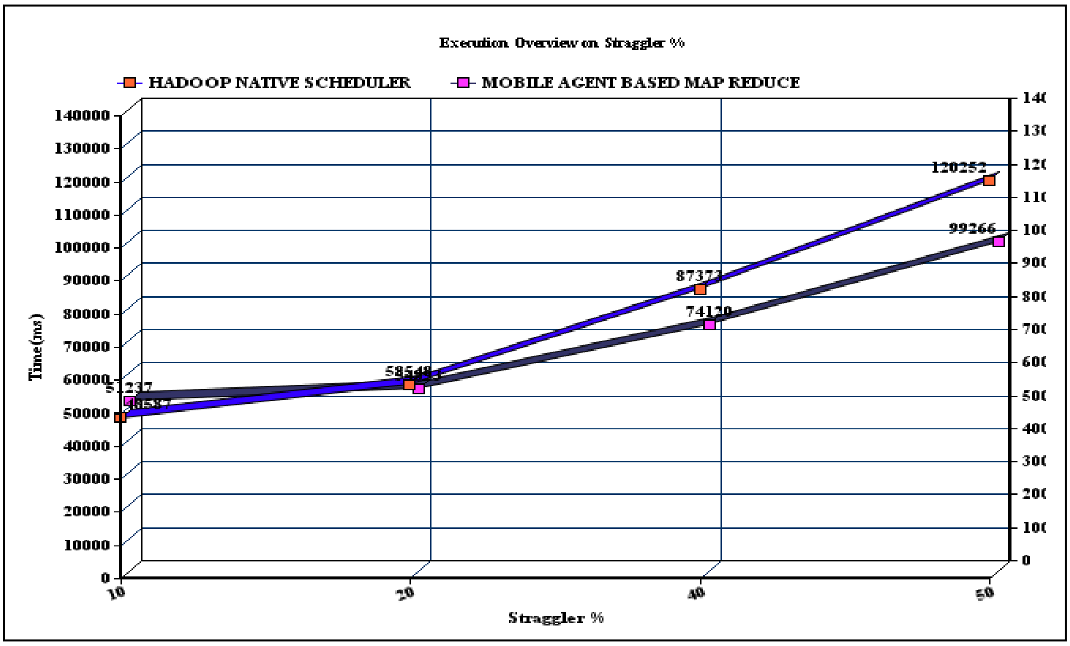


Figure. 6.7 Graph showing performance comparison with HADOOP based on split sizes

For a particular network speed, number of splits can affect delay in MapReduce operation. This can increase the overall execution time if network conditions are not good. From the previous section, it is evident that the MBMR Algorithm performs well on a single machine as well as for a network of machines.

Table 6.3 shows execution overview on the basis of various split sizes. The significant insights from the result of the implementation of MBMR are as follows:

- Stragglers are detected and mobile agents are then rescheduled to other machine without wasting a single machine cycle.
- There is a significant improvement in the execution time as compared to HADOOP's native algorithms, i.e., about 10–12%.

Table 6.3: Execution overview on the basis of various split sizes

Setup Scenario Size*Container*Agents/ Container	**HADOOP Native Scheduler Time(ms)**	**Mobile Agent Based MapReduce Time(ms)**
50*3*3	45	48
45*2*5	47	48
30*3*5	50	49
15*3*10	55	55

Rescheduling process can be delayed in case a node has stuck in some useful work. In this case, it is not marked as straggler and can continue its work.

6.4 SUMMARY

Straggler detection and mitigation approaches have attracted a lot of attention of researchers in recent years and there is a considerable increase in the number of algorithms published for solving issue as it has applications in various domains like big data processing and parallel computation. Proposed approach tried to find a new approach for straggler detection and mitigation along with their rescheduling. The main goal was to come up with a technique which is better than the current state of the art solutions. The proposed technique of Mobile Agent based MapReduce for Big Data Processing and straggler detection is described. Tests are also performed and compared with the Hadoop native scheduler. The proposed technique performs well as compared to the classical algorithms and the current state of art algorithms.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

Proposed authentication framework provides a robust authentication mechanism and over this, different authentication protocols can be incorporated like Extensible Authentication Protocol (EAP). Mobile agents in the proposed approach help in greatly reducing the traffic in the network and at the authentication server. Key exchange with authentication of the KDC and client node is also proposed. This method makes use of the biometric data for the authentication and key generation. Device Signature is also one of the approach for authentication and a new Android operating system based device signature based authentication mechanism has been developed. After authentication mobile agent based MapReduce framework for BigData processing is also proposed which promises for the authentication, efficient transmission and execution of task in the distributed environment.

7.2 CONTRIBUTIONS

The main aim of the proposed research is to maximize the security of the existing security protocols. In the area of key exchange the contributions are:

- *Key Exchange with Biometric Authentication:* Proposed work in this area provides a unique method for generation of keys and authentication of KDC and communication parties with the biometric information.
- *Traffic reduction:* Proposed KDFBA protocol greatly reduces the traffic at the KDC to a great extent. Inadvertently the traffic on the network is also reduced to a great extent.
- *Protection against the black hole attack:* Proposed key exchange model protects the nodes against the black hole attack. Result analysis clearly shows that model greatly decreases the vulnerability to the attack.

In the area of authentication, a mobile agent based framework is proposed for wireless authentication. The key contributions of this model are:

- *Dynamic change in underlying algorithm:* Proposed framework for authentication can incorporate authentication algorithm which can be changed dynamically without change in the underlying architecture.
- *Traffic Reduction:* Number of message exchanges to the authentication server is greatly reduced so, number of users supported by the authentication server will be increased to a great number.

In the area of authentication again we worked on the device fingerprint based authentication mechanism which provides a robust authentication mechanism for protecting the network from unauthorized user access. The key contributions of this model are:

- *Unique Device Fingerprint:* A unique device fingerprint is generated which can be used for the authentication of the user on the network.
- *Android Application:* An android application which generates the unique fingerprint of the device is developed.

In the area of mapreduce a mobile agent based approach is developed and provides an improvement over current state of the art techniques in terms of:

Throughput: Throughput is increased in case of mobile agent based approach compared with the Hadoop native scheduler to a good extent.

7.3 FUTURE SCOPE

Proposed work can be extended in some other more interesting scenarios which can be explored in the future. Some of these are:

- **Mobile Agent Itinerary**

  Itinerary is the path that mobile agent follows from one node to another during its lifetime. In the proposed framework, key exchange and BigData processing itinerary plays an important role. Itinerary can be static and dynamic. Itinerary in proposed mobile agent framework and key exchange mechanism can be worked upon in future.

- **Fault Tolerance**

  As wireless network are adhoc in nature and user can move from one location to another from access point to another so, if an access point fails or is overloaded, then work can be done in this field also.

- **Internet of Things (IoT)**

  For continuous availability and diagnosis purpose, concept of Internet of Things (IoT) can be integrated into mobile agent based authentication framework. Mobile agent based proposed approach can be tested on IoT concept also in terms of authentication, key exchange and data aggregation.

- **Cloud Computing**

  Cloud computing can be beneficial in mobile agent based approaches as it provides ample storage and computing resources. In wireless security cloud computing can be helpful for storage of authentication related data like username, password and authentication algorithm processing can be done at cloud itself.

- **Network Mobility**

  Wireless network supports node mobility as user can move from one location to another in no time. So suitable network mobility and proper handover in case of mobile agent based approach can also be explored in future.

REFERENCES

[1] T. S. Rappaport, “Wireless Communications: Principles and Practice,” Upper Saddle River, New Jersey, Prentice Hall, 2002.

[2] Naftali Herscovici, Christos Christodoulou, “Wireless Communications and Networking: An Overview,” *IEEE Antenna’s and Propagation Magazine*, vol. 44, no. 1, 2002, pp. 185- 193.

[3] P. Gupta and P. Kumar, “The Capacity of Wireless Networks,” *IEEE Transactions on Information Theory*, vol. 46, no. 2, May 2000, pp. 388-404.

[4] Lee, Jin-Shyan, Yu-Wei Su, and Chung-Chou Shen, “A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi,” *Industrial electronics society*, vol. 5, 2007, pp. 46-51.

[5] Lusignan, Bruce, “Cellular data system,” U.S. Patent no. 4, 914, 651, 3 Apr. 1990.

[6] Doshi, Bharat Tarachand, “Cellular system architectures supporting data services,” U.S. Patent no. 5,729,536, 1998.

[7] WW. Wu, E.F. Miller, W.L Pritchard, R.L. Pickholtz, “Mobile satellite communications,” vol. 82, no. 9, 1994, pp. 1431-1448.

[8] Willig, Andreas, Kirsten Matheus, Adam Wolisz, “Wireless technology in industrial networks,” *Proceedings of the IEEE*, vol. 93, no. 6, 2005, pp. 1130-1151.

[9] Gomez, Carles, Joaquim Oller, Josep Paradells, “Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology,” *Sensors, vol.* 12, no. 9, 2012, pp. 11734-11753.

[10] http://csrc.nist.gov/wireless.

[11] http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html.

[12] Ignacio Velasquez, Angelica Caro, Alfonso Rodriguez, “Authentication schemes and methods: A systematic literature review,” *Information and Software Technology*, vol. 94, 2018, pp. 30-37.

[13] Jayson E. Street, Kent Nabors, Brian Baskin, Marcus Carey, *Dissecting the hack.* vol. 1, 2010, pp. 195-227.

[14] Ramesh Yegireddi, RKiran Kumar, “A survey on Conventional Encryption Algorithms of Cryptography*,” International Conference on ICT in Business Industry & Government (ICTBIG),* 2016.

[15] M. Abdalla, D. Pointcheval, “Simple Password-based Encrypted Key Exchange Protocols,” *in Topics in Cryptology-CT-RSA 2005,* Spring-Verlag, 2005, pp. 191-208.

[16] J. Baumann, F. Hohl, K. Rothermel, M. Straber, “Mole–Concepts of a mobile agent system,” *World Wide Web,* vol. 1, no. 3, 1998, pp. 123-137.

[17] D. Gavalas, D. Greenwood, M. Ghanbari, M. O. Mahony, “Advanced network monitoring applications based on mobile/intelligent agent technology,” *Computer Communications,* vol. 23, no. 8, pp. 720-730.

[18] Yulong Zou, Jia Zhu, Xianbin Wang, Lajos Hanzo, “A survey on wireless security: Technical challenges, recent advances, and future trends,” *Proceedings of the IEEE,* 2016, vol. 104, no. 9, pp. 1727-65.

[19] Okan Can, Ozgur Koray Sahingoz, “A survey of intrusion detection systems in wireless sensor networks,” *6th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO),* 2015, pp. 1-6.

[20] Yulong Zou, Jia Zhu, Xianbin Wang, and Lajos Hanzo, “A survey on wireless security: Technical challenges, recent advances, and future trends,” *Proceedings of the IEEE,* vol. 104, no. 9, 2016, pp. 1727-1765.

[21] Xu Lan, “Analysis and research of several network traffic prediction models,” *Chinese Automation Congress (CAC),* 2013, pp. 894-899. doi:10.1109/CAC.2013.6775859

[22] Xiangyun Zhou, Behrouz Maham, Are Hjorungnes, “Pilot contamination for active eavesdropping,” *IEEE Transactions on Wireless Communications*, vol. 11, no. 3, 2012, pp. 903–907.

[23] Sherin Hijazi, Mohammad S. Obaidat, “A New Detection and Prevention System for ARP Attacks Using Static Entry,” *IEEE Systems Journal,* vol. 13, no. 03, 2018, pp. 2732-2738.
doi:10.1109/jsyst.2018.2880229

[24] M. Conti, N. Dragoni, and V. Lesyk, ‘‘A survey of man in the middle attacks,’’ *IEEE Communications. Surveys and Tutorials,* vol. 18, no. 3, 2016, pp. 2027–2051.

[25] Yisroel Mirsky, Naor Kalbo, Yuval Elovici, Asaf Shabtai, “Vesper: Using Echo Analysis to Detect Man-in-the-Middle Attacks in LANs,” *IEEE Transactions on Information Forensics and Security,* vol. 14, no. 6, 2016, pp. 1638-1653.

[26] Qiao Hu and Gerhard. Petrus Hancke, ‘‘A session hijacking attack on physical layer key generation agreement,’’ *IEEE International Conference on Industrial Technology*, 2017, pp. 1418–1423.

[27] Jing, C., Wang, C., & Yan, C., “Replay Attack: A Prevalent Pattern of Fraudulent Online Transactions,” *5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)*, pp. 75-82, 2018. doi:10.1109/cscloud/edgecom.2018.00

[28] Zi Feng, Jianxia Ning, Ioannis Broustis, Konstantinos Pelechrinis, Srikanth V. Krishnamurthy, and Michalis Faloutsos, "Coping with packet replay attacks in wireless networks," *8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2011, pp. 368-376.

[29] Hong Huang, Nihal Ahmed, Pappu Karthik, “On a New Type of Denial of Service Attack in Wireless Network: The Distributed Jammer Network,” *IEEE Transactions on Wireless Communications*, vol.10, no.7, 2011, pp. 2316–2324.
doi:10.1109/twc.2011.052311.101613

[30] Mallikarjunan, K. Narasimha, K. Muthupriya, S. Mercy Shalinie "A survey of distributed denial of service attack," *2016 10th International Conference on Intelligent Systems and Control (ISCO)*. IEEE 2016, pp. 1-6.

[31] Tatli Emin Islam, ‘‘Cracking more password hashes with patterns,’’ *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 8, 2015, pp. 1656–1665.

[32] Ding K, Pantic N, Lu Y, Manna S, Husain MI, “Towards building a word similarity dictionary for personality bias classification of phishing email contents,” *9th International Conference on Semantic Computing (IEEE ICSC 2015)*, 2015, pp. 252-259.

[33] Berger, Yigael, Avishai Wool, and Arie Yeredor, “Dictionary attacks using keyboard acoustic emanations,” *Proceedings of the 13th ACM conference on Computer and communications security*, 2006, pp. 245-254.

[34] Qiuwei Yang, Xiaogang Zhu, Hongjuan Fu, Xiqiang Che, “Survey of security technologies on wireless sensor networks,” *Journal of sensors*, vol. 2015, pp. 1-9.

[35] Jitender Grover, Shikha Sharma, “Security issues in wireless sensor network—a review,” *2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, IEEE, 2016, pp. 397-401.

[36] M. Vandenwauver, R. Govaerts, J. Vandewalle, "Overview of Authentication Protocols: Kerberos and SESAME,” *Proceedings 31st Annual IEEE Camahan Conference on Security Technology*, 1997, pp. 108-113.

doi:10.1109/ccst.1997.626248

[37] Şeyma Bat, Didem Gözüpek, “Joint Optimization of Cash Management and Routing for New-Generation Automated Teller Machine Networks,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017, pp.1-15.

[38] R.L. Zunkei, “Hand geometry based verifcation, ” *A.K. Jain, R. Bolle, S. Pankanti (Eds.),* Biometrics: Personal Identifcation in Networks Society, Kluwer Academic Publishers, Dordrecht, 1999, pp. 87–101.

[39] D S. Stienne, Nathan Clarke and Paul Reynolds, Strong Authentication for Web Services using Smartcards, *7th Australian Information Security Management Conference*, 2013, pp. 55-62.

[40] B. Lloyd and W. Simpson, “PPP Authentication Protocol,” Internet Engineering Task Force (IETF) RFC 4017 October 1992.

[41] J. Galka, M. Masior, M. Salasa, "Voice authentication embedded solution for secured access control", *Consumer Electronics IEEE Transactions on Consumer Electronics*, vol. 60, no. 4, 2014, pp. 653-661.

[42] C. Shyamala Kumari and M. Deepa Rani, “Hacking Resistance Protocol For Securing Passwords Using Personal Device,” *7th International Conference on ISCO*, 2013, pp. 458-463.

[43] L. Latha, M. Pabitha and S. Thangasamy, "A Novel Method for Person Authentication using Retinal Images,” *International Conference on Innovative Computing Technologies (ICICT),* 2010, pp. 1-6.

[44] Chowdhury M., Gao J., Islam R., “Biometric Authentication Using Facial Recognition, ” *In: Deng R., Weng J., Ren K., Yegneswaran V. (eds) Security and Privacy in Communication Networks. SecureComm 2016*, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 198. Springer, 2017, pp. 287-295.

[45] Shallen Giroux, Renata Smolikova-Wachowiak, P. Wachowiak, Mark, "Keystroke-Based Authentication by Key Press Intervals as a Complementary Behavioral Biometric,” *Proceedings of the IEEE International Conference on Systems Man and Cybernetics*, 2009, pp. 80-85.

[46] Jyh-Chen and Yu-Ping Wang, “Extensible Authentication Protocol (EAP) and IEEE 802.1x: Tutorial and Empirical Experience,” *IEEE Communications Magazine*, Dec 2005, pp. 26-32.

[47] W. Wang, X. Liu, J. Vicente, and P. Mohapatra, "Integration gain of heterogeneous WiFi/WiMAX networks, " *IEEE Transactions Mobile Computing*, vol. 10, no. 8, 2011, pp. 1131-1143.

[48] C.-I. Fan, Y.-H. Lin, R.-H. Hsu, "Complete EAP method: User efficient and forward secure authentication protocol for IEEE 802.11 wireless LANs", *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 4, 2013, pp. 672-680.

[49] S. Willens, "Remote Authentication Dial In User Service (RADIUS)", *RFC 2865*, June 2000.

[50] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., & Levkowetz, H., “Extensible authentication protocol (EAP),” *RFC 3748*, June 2004.

[51] Cisco Wireless LAN Security by Andrew Balinsky, Darrin Miller, Krishna Sankar, Sri Sundaralingam Publisher: Cisco Press, 2004.

[52] R. Rivest, The MD5 Message-Digest Algorithm, RFC Editor, 1992.

[53] D. Stanley, J. Walker, B. Aboba, "Extensible Authentication Protocol (EAP) Method Requirements for Wireless LANs", *RFC 4017*, Mar. 2005.

[54] Sencun Zhu, Sanjeev Setia, Sushil Jajodia, “LEAP: efficient security mechanisms for large-scale distributed sensor networks,” *Proceedings of the 10th ACM conference on Computer and communications security*, Washington D.C., USA, October 27-30, 2003.

[55] Umesh Kumar, Parveen Kumar and Sapna Gambhir, “Analysis and Literature Review of IEEE 802.1x Authentication Protocols,” *International Journal of Engineering and Advanced Technology*, vol. 03, no. 05, 2014, pp. 163-168.

[56] Erik Tews , Martin Beck, “Practical attacks against WEP and WPA,” *Proceedings of the second ACM conference on Wireless network security*, Zurich, Switzerland, March 2009, pp. 79-86.

[57] Zorn, G. and S. Cobb, “Microsoft PPP CHAP Extensions”, *RFC 2433*, October 1998.

[58] Anjali K. Rai, Shivendu Mishra and Vimal Kumar, Strong Password Based EAP-TLS Authentication Protocol for WiMAX, *International Journal on Computer Science and Engineering*, vol. 02, no. 08, 2010, pp. 2736-2741.

[59] Aboba, B. and D. Simon, "PPP EAP TLS Authentication Protocol," *RFC 2716*, October 1999.

[60] W. Chou, "Inside SSL: Accelerating Secure Transactions", *IT Professional*, vol. 4, no. 4, 2002, pp. 37-41.

[61] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," *RFC 5280* (Proposed Standard), May 2008. Updated by RFC 6818.

[62] Umesh Kumar, Sapna Gambhir, "A Literature Review of Security Threats to Wireless Networks", *International Journal of Future Generation Communication and Networking*, vol. 7, no. 4, 2014, pp. 25-34.

[63] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., & Levkowetz, H., "Extensible authentication protocol (EAP)," *RFC 3748*, June 2004.

[64] Bakytbek Eshmurzaev and Gokhan Dalkilic, "Analysis of EAP-FAST Protocol," *34th International Conference on Information Technology Interfaces*, Cavtat, Croatia June 2012, pp. 417-422.

[65] Bahareh Shojaie, Iman Saberi , Mazleena Salleh, "Improving EAP-TLS Performance Using Cryptographic Methods," International Conference on Computer & Information Science, June 2012, 760-764.

[66] David Q. Liu , Mark Coslow, "Extensible authentication protocols for IEEE standards 802.11 and 802.160," *Proceedings of the International Conference on Mobile Technology, Applications, and Systems*, September 10-12, 2008, Yilan, Taiwan.

[67] Soo-Cheol Kim, Jung-Sik Cho and Sung Kwon Kim, "Agent Tag Based User Authentication Protocol for Mobile IPTV Service," *Proc. International Conference on Consumer Electronics (ICCE)* vol. 3, 2010, pp. 325-333.

[68] Ya-ling Zhang, JianBai, "The Design of Interactive Authentication for Virtual Enterprise Based on Mobile Agent," *2nd International Symposium on Information Engineering and Electronic Commerce*, 2010, pp. 277-280.

[69] Sandeep K.Sood, Anil K.Sarje, Kuldip Singh, “A secure dynamic identity based authentication protocol for multi-server architecture,” *Journal of Network and Computer Applications*, vol. 34, 2011, pp. 609-618.

[70] Y.P. Liao, S.S. Wang, “A secure dynamic id-based remote user authentication scheme for multi-server environment,” *Computer Standards & Interface*, vol. 31, no. 1, 2009, pp. 24-29.

[71] H.C. Hsiang, W.K. Shih, “Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment,” *Computer Standards & Interface*, vol. 31, no. 6, 2009, pp. 1118-1123.

[72] Chin-Ling Chen, Chien-Hung Chen , Chih-Cheng Chen, “ Cryptanalysis of a Secure Dynamic Identity Based Authentication Protocol for Multi-server Architecture,” *IEEE International Symposium on Computer, Consumer and Control*, 2012.

[73] Xiumei Liu, Junjiang Liu , Guiran Chang, “nPAKE: An Improved Group PAKE Protocol,” *IEEE Ninth Web Information Systems and Applications Conference*, 2012.

[74] Gang Yao, Hongji Wang, Dengguo Feng, “A Group PAKE Protocol Using Different Passwords,” *International Conference on Networks Security, Wireless Communications and Trusted Computing*, 2009.

[75] Z. Liu, J. Ma, Q. Huang, "Asymmetric key pre-distribution scheme for sensor networks", *IEEE Transactions on Wireless Communication.*, vol. 8, no. 3, 2009, pp. 1366-1372.

[76] U. Chatterjee, R. S. Chakraborty, D. Mukhopadhyay, "A PUF-Based Secure Communication Protocol for IoT", ACM Transactions on Embedded Computing Systems (TECS), vol. 16, no. 3, Apr. 2017, pp. 1-25.

[77] Wen-Shenq Juang, “Efficient multi-server password authenticated key agreement using smart cards”, *IEEE Transactions on Consumer Electronics*, vol. 50 no. 1, February 2004, pp.251-255.

[78] Seung-Hyun. Seo, Jongho Won, Salmin Sultana, Elisa Bertino, "Effective key management in dynamic wireless sensor networks", *IEEE Transaction on Information Forensics Security*, vol. 10, no. 2, Feb. 2015, pp. 371-383.

[79] M. R. Alagheband and M. R. Aref, “Dynamic and secure key management model for hierarchical heterogeneous sensor networks,” *IET Information Security*, vol. 6, no. 4, 2012, pp. 271-280.

[80] Sattam S. Al-Riyami, Kenneth G. Paterson, "Certificate less public key cryptography" *in Asiacrypt, Berlin, Germany: Springer*, vol. 2894, 2003, pp. 452-473.

[81] Rai A.K., Mishra S., Tripathi P.N., “An Improved Secure Authentication Protocol for WiMAX with Formal Verification,” *In: Abraham A., Lloret Mauri J., Buford J.F., Suzuki J., Thampi S.M. (eds) Advances in Computing and Communications*, vol. 191, pp. 407-416.

[82] Babak Daghighi, Laiha Mat Kiah, Salman Iqbal, Muhammad Habib Ur Rehman, Keith Martin, “Host mobility key management in dynamic secure group communication, ” *Wireless Networks*, vol. 24, no.8, 2018, pp. 3009-3027.

[83] T. Kohno, A. Broido, K.C. Claffy, "Remote physical device fingerprinting", IEEE Transactions on Dependable Secure Computing, vol. 2, no. 2, 2005, pp. 93-108.

[84] Shimshon Berkovits , Joshua D. Guttman, Vipin Swarup, “Authentication for Mobile Agents,” *In: Vigna G. (eds) Mobile Agents and Security. Lecture Notes in Computer Science*, vol. 1419, 1998, pp. 114-136.

[85] W. Fang, C. Zhang, Z. Shi, Q. Zhao and L. Shan, “BTRES: beta-based trust and reputation evaluation system for wireless sensor networks”, *Journal of Network and Computer Applications*, vol. 59, no. 1, pp. 88-94, 2016.

[86] G. P. Gupta, M. Misra, K. Garg, "Energy and trust aware mobile agent migration protocol for data aggregation in wireless sensor networks,” *Journal of Network and Computer Applications*, vol. 41, May 2014, pp. 300-311.

[87] N. Sae-Bae and N. Memon, Fellow, IEEE, “Online Signature Verification on Mobile Devices,” *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 6, 2014, pp. 933 – 947.

[88] G. Starnberger, L. Froihofer and K. M. Goeschka, “QR-TAN: Secure mobile transaction authentication,” *IEEE Computer Society, ARES '09 The Fourth International Conference on Availability, Reliability and Security*, 2009, pp. 578-583.

[89] G. Geetha and C. Jayakumar, “Implementation of Trust and Reputation Management for Free-Roaming Mobile Agent Security,” *IEEE Systems Journal*, vol. 9, no. 2, 2015, pp. 556-56.

[90] Bhavin Shah and Bhushan H. Trivedi, “Improving Performance of Mobile Agent Based Intrusion Detection System,” *Fifth International Conference on Advanced Computing & Communication Technologies*, 2015, pp. 425-430.

[91] D. Prasad Sharma, “Mobile Agent-Based Authentication: A Model for User Authentication in a Distributed System,” *International Journal of Computer Applications*, vol. 112, no. 13, 2015, pp. 20-25.

[92] M. Dong, K. Ota, L. T. Yang, S. Chang, H. Zhu, Z. Zhou, "Mobile agent-based energy-aware and user-centric data collection in wireless sensor networks", *Computer Network*, vol. 74, Dec. 2014, pp. 58-70.

[93] M. Riecker, S. Biedermann, M. Hollick, "Lightweight energy consumption based intrusion detection system for wireless sensor networks,” International Journal of Information Security, vol. 14, no. 2, 2015, pp. 155-167.

[94] B. Lampson, M. Abadi, M. Burrows and E. Wobber, “Authentication in distributed systems: Theory and practice”, *ACM Transactions on Computer Systems*, vol. 10, no. 4, 1992, pp. 265-310.

[95] Weidong Fang, Chuanei Zhang, Zhidong Shi, Quing Zhao and Lianhai Shan, “BTRES: beta-based trust and reputation evaluation system for wireless sensor networks,” *Journal of Network and Computer Applications*, vol. 59, no. 1, 2016, pp. 88-94.

[96] N. Sae-Bae and N. Memon, Fellow, IEEE, “Online Signature Verification on Mobile Devices,” *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 6, 2014, pp. 933 – 947.

[97] Guenther Starnberger, Lorenz Froihofer and Karl M. Goeschka, “QR-TAN: Secure mobile transaction authentication,” *IEEE Computer Society, ARES '09 The Fourth International Conference on Availability, Reliability and Security*, 2009, pp. 578-583.

[98] G. Geetha and C. Jayakumar, “Implementation of Trust and Reputation Management for Free-Roaming Mobile Agent Security”, *IEEE Systems Journal*, vol. 9, no. 2, 2015, pp. 556-56.

[99] B. Shah and B. H. Trivedi, “Improving Performance of Mobile Agent Based Intrusion Detection System,” *Fifth International Conference on Advanced Computing & Communication Technologies*, 2015, pp. 425-430.

[100] D. Prasad Sharma, “Mobile Agent-Based Authentication: A Model for User Authentication in a Distributed System”, *International Journal of Computer Applications*, vol. 112, no. 13, 2015, pp. 20-25.

[101] Laura Wilber, Steve Mills, and Bill Perlowitz, “Demystifying Big Data,” *Notices of TA Foundation*, 2009.

[102] R. Elmasri, S.B. Navathe, “Fundamentals of Database Systems (7th edition),” *Pearson/Addison-Wesley*, 2015.

[103] S. Singh and N. Singh, "Big Data Analytics,” *2012 International Conference on Communication, Information & Computing Technology*, IEEE, October 2011.

[104] Jeffrey Dean, Sanjay Ghemawat, “MapReduce: simplified data processing on large clusters,” *Communications of the ACM 50*, vol. 51, no. 1, 2008, pp. 107–113.

[105] Ganesh Ananthanarayanan, Ali Ghodsi, Scott Shenker, Ion Stoica, “Effective Straggler Mitigation: Attack of the Clones,” *nsdi'13 Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation*, 2008, pp. 185-198.

[106] Jia Li, Changjian Wang, Dongsheng-Li, Zhen Huang, "Partial Clones for Stragglers in MapReduce,” *International Conference of Young Computer Scientists, Engineers and Educators*, vol. 503, 2015, pp. 109-116.

[107] G. Ananthanarayanan, S. Kandula, A. Greenberg, I. Stoica, E. Harris, and B. Saha, “Reining in the Outliers in Map-Reduce Clusters using Mantri,” In USENIX OSDI, 2010.

[108] Rohan Gandhi, Amit Sabne "Finding Stragglers in Hadoop", In Proceedings of the 8th USENIX conference on Operating systems design and implementation, OSDI, vol. 32, pp. 425-443, 2008.

[109] William Glenn and Wei Yu, “Cyber Attacks on MapReduce Computation Time in a Hadoop Cluster, ” Big Data Concepts, Theories, and Applications, 2016, pp. 257–279.

[110] HDFS architecture (2014) The Apache Software Foundation. [Online]. Available: http:// hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html. Accessed 2 May 2015

[111] The Apache Software Foundation. [Online]. Available: https://hadoop.apache.org/docs/r2.6.0/ hadoop-yarn/hadoop-yarn-common/yarn-default.xml.

[112] The Apache Software Foundation. [Online]. Available: http://hadoop.apache.org/docs/r2.6. 0/hadoop-mapreduce-client/hadoop-mapreduce-client-core/mapred-default.xml. Accessed 3 May 2015

[113] The Apache Software Foundation. [Online]. Available: http://hadoop.apache.org/docs/r2.6.0/ hadoop-project-dist/hadoop-common/core-default.xml. Accessed 3 May 2015

[114] The Apache Software Foundation. [Online]. Available: https://hadoop.apache.org/docs/r2.6.0/ hadoop-project-dist/hadoop-hdfs/hdfs-default.xml. Accessed 3 May 2015

[115] Wesley M. Eddy, Verizon Federal Network Systems, "Defenses Against TCP SYN Flooding Attacks," The Internet Protocol Journal, vol. 9, no. 4, Dec 2006.

[116] Gurjit Singh Bhathal, Amardeep Singh, "Big data: Hadoop framework vulnerabilities, security issues and attacks," Array, vol. 1-2, 100002, 2019.

[117] M. Zaharia, A. Konwinski, A. D. Joseph, R. Katz, and I. Stoica. "Effective Straggler Mitigation: Attack of the Clones". In Proceedings of the 8th USENIX conference on Operating systems design and implementation, OSDI'08, pp. 29–42, 2008.

[118] G. Ananthanarayanan, S. Kandula, A. Greenberg, I. Stoica, Y. Lu, B. Saha, and E. Harris, "Reining in the Outliers in Map-Reduce Clusters using Mantri, " In Proc. of USENIX OSDI, 2010.

[119] Sufian Hameed and Usman Ali, "HADEC: Hadoop-based live DDoS detection framework," *EURASIP Journal on Information Security*, vol. 1, no. 11, 2018.

[120] Yeonhee Lee, Youngseok Lee, "Detecting DDoS attacks with Hadoop," *Proceedings of The ACM CoNEXT Student Workshop on - CoNEXT '11 Student*, no. 7, 2011, pp. 1-20.

[121] Yoon-Su Jeong, Yong-Tae Kim, "A token-based authentication security scheme for Hadoop distributed file system using elliptic curve cryptography," *Journal of Computer Virology and Hacking Techniques*, vol. 11, no. 3, 2015, pp. 137-142.

[122] Durbadal Chattaraj, Monalisa Sarma, Ashok Kumar Das, Neeraj Kumar, J. J. P. C. Rodrigues, and Youngho Park, "HEAP: An Efficient and Fault-Tolerant Authentication and Key Exchange Protocol for Hadoop-Assisted Big Data Platform," *IEEE Access*, vol. 6, 2018, pp. 75342–75382.

[123] Z. Shen, L. Li, F. Yan, and X. Wu, ''Cloud computing system based on trusted computing platform,'' *Proc. Int. Conf. Intell. Comput. Technol. Automat. (ICICTA)*, vol. 1, 2010, pp. 942–945.

[124] J. Baker, S. Savino, "The role of client/server computing technology in the management of global enterprises," *Innovation in Technology Management. The Key to Global Leadership,* July 1997.

[125] https://www.statista.com/topics/2157/internet-usage-in-india/

[126] Umesh Kumar, Sapna Gambhir, "Mobile Agent Based MapReduce Framework for Big Data Processing," In: Aggarwal V., Bhatnagar V., Mishra D. (eds) Big Data Analytics. Advances in Intelligent Systems and Computing, vol. 654, 2017, pp. 391-402.

[127] C. Konstantopoulos, A. Mpitziopoulos, D. Gavalas, G. Pantziou, "Effective determination of mobile agent itineraries for data aggregation on sensor networks", *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 12, 2010, pp. 1679-1693.

[128] Mohamed El Fissaoui, Abderrahim Beni-hssane, Mostafa Saadi, "Multi-mobile agent itinerary planning-based energy and fault aware data aggregation in wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking,* 2018.

[129] Wang, X.; Chen, M.; Kwon, T.; Chao, H.C., "Multiple mobile agents' itinerary planning in wireless sensor networks: Survey and evaluation," *IET Communication*, vol. 5, 2011, pp. 1769–1776.

[130] Huthiafa Q Qadori, Zuriati A Zulkarnain, Zurina Mohd Hanapi and Shamala Subramaniam, "Multi-mobile agent itinerary planning algorithms for data gathering in wireless sensor networks: A review paper," *International Journal of Distributed Sensor Network,* vol. 13, no. 2, 2017, pp. 1-13.

[131] William Stallings, "Cryptography and Network Security (4th Edition)," Upper Saddle River, NJ, USA: Prentice-Hall, Inc.2005.

[132] Williams, H., "A modification of the RSA public-key encryption procedure," *IEEE Transactions on Information Theory*, vol. 26, no. 6, 1980, pp. 726–729.

https://doi.org/10.1109/TIT.1980.1056264

[133] Nan Li, “Research on Diffie-Hellman key exchange protocol,” *2nd International Conference on Computer Engineering and Technology*, 2010, pp. 634-637.
https://doi.org/10.1109/ICCET.2010.5485276

[134] Du, X., Guizani, M., Xiao, Y., & Chen, H.-H., “Transactions papers a routing-driven Elliptic Curve Cryptography based key management scheme for Heterogeneous Sensor Networks,” *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, 2009, pp. 1223–1229.

[135] Dong, J., Zheng, F., Cheng, J., Lin, J., Pan, W., & Wang, Z., “Towards High-performance X25519/448 Key Agreement in General Purpose GPUs,” *2018 IEEE Conference on Communications and Network Security (CNS)*, 2018.
https://doi.org/10.1109/cns.2018.8433161

[136] Zhang, W., Lin, D., Zhang, H., Zhou, X., and Gao, Y., “A Lightweight FourQ Primitive on ARM Cortex-M0,” *17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications*,2018, pp. 699-704.
https://doi.org/10.1109/TrustCom/BigDataSE.2018.00102

[137] Cimato, S., Cresti, A., D’Arco, P., “A unified model for unconditionally secure key distribution,” Journal of Computer Security, vol. 14, no. 1, 2006, pp. 45–64.
https://doi.org/10.3233/JCS-2006-14102

[138] Blundo, C., De Santis, A., Herzberg, A., Kutten, S., Vaccaro, U., Yung, M., “Perfectly Secure Key Distribution for Dynamic Conferences,” *Information and Computation*, vol. 146, no. 1, 1998, pp. 1–23.

[139] Chatterjee K., De A., Gupta D., “An Improved ID-Based Key Management Scheme in Wireless Sensor Network,” *In: Tan Y., Shi Y., Ji Z. (eds) Advances in Swarm Intelligence. ICSI 2012. Lecture Notes in Computer Science*, Springer, vol. 7332, 2012, pp.351-359.

[140] Jin Z., Jin Teoh A.B., Ong T.S., Tee C., “Secure Minutiae-Based Fingerprint Templates Using Random Triangle Hashing,” *In: Badioze Zaman H., Robinson P., Petrou M., Olivier P., Schröder H., Shih T.K. (eds) Visual Informatics: Bridging Research and Practice, Lecture Notes in Computer Science*, vol. 5857, 2009, pp. 521-531.

List of Published Papers in Journals

S.No.	Title of Paper	Name of Journal where published	No.	Volume and Issue	Year	Pages	Indexing
1.	KDFBA: Key Distribution through Fingerprint based Authentication using Mobile Agent	Multimedia Tools and Applications	ISSN: 1573-7721	Vol. 79 No. 19	2020	13891-13918	Science Citation Index Expanded
2.	Device Fingerprint and Mobile Agent based Authentication Technique in Wireless Networks	International Journal of Future Generation Communication and Networking	ISSN: 2233-7857	Vol. 11 No.3	2018	33-48	ESCI, Google Scholar
3.	Secured Authentication Method for Wireless Networks	IOSR Journal of Computer Engineering	ISSN: 2278-0661	Vol. 1 No. 1	2015	01-11	UGC, Google Scholar
4.	A Literature Review of Security Threats to Wireless Networks	International Journal of Future Generation Communication and Networking	ISSN: 2233-7857	Vol. 7 No. 4	2014	25-34	ESCI, Google Scholar
5.	Analysis and Literature Review of IEEE 802.1x (Authentication) Protocols	International Journal of Engineering and Advanced Technology (IJEAT)	ISSN: 2249-8958	Vol. 5 No. 3	2014	163-168	UGC, Google Scholar

List of Published papers in Conferences

S.No	Title of Paper	Name of Conference	Indexing	Pages	Year
1.	MABFWA: Mobile Agent Based Framework for Wireless Authentication	8th International Conference on Cloud Computing, Data Science & Engineering (Confluence)	Scopus, Google Scholar	219-224	2018
2.	A Novel Approach for Key Distribution through Fingerprint based Authentication using Mobile Agent	3rd International Conference on Computing for Sustainable Global Development (INDIACom)	Scopus, Google Scholar	3441-3445	2016
3.	Mobile Agent Based MapReduce Framework for Big Data Processing	Aggarwal V., Bhatnagar V., Mishra D. (eds) Big Data Analytics. Advances in Intelligent Systems and Computing	DBLP, SCOPUS, Google Scholar and Springerlink	391-402	2018
4.	Mobile Agent Technology: Possible Attacks and Security Requirements	National Conference on Emerging Trends in Computer Science and Information Technology		13-16	2015
5.	Study of Various Wireless Security Protocols and Design of Enhanced- WPA2 (e-WPA2)	6th International Conference on Advanced Computing and Communications Technologies (ICACCT)		3-9	2012

BRIEF BIODATA OF THE RESEARCH SCHOLAR

Umesh Kumar has completed his B.E. (Computer Engineering) from Deenbandhu Chhotu Ram University of Science and Technology (DCRUST), Murthal, Sonepat in 2007. He has completed his M.Tech. in Computer Engineering from J.C. Bose University of Science and Technology, YMCA, Faridabad and pursuing his Ph.D (Computer Engineering) from the same University under the supervision of Dr. Sapna Gambhir, Assistant Professor in Department of Computer Engineering. Umesh has worked as Assistant System Engineer (ASE) in Tata Consultancy Services (TCS) for around two years (2010-2012). Presently, he is working as an Assistant Professor in Department of Computer Engineering since April 2012. He is having around 8 years of teaching experience. He has published more than 13 research papers in International/National journals and conferences. His area of interest includes Mobile Agent, Wireless Security, Authentication and Wireless Communication techniques.