GENERIC LIGHT WEIGHT CERTIFICATE MANAGEMENT PROTOCOL

BY

Muhammad Asif

2008-NUST-MS-CCS-02

Supervisor

Dr. Abdul Ghafoor (DoC)

A thesis submitted in partial fulfillment of the requirements for the degree
Of Masters of Science in Computer & Communication Security (MS-CCS)
IN
School of Electrical Engineering and Computer Science (SEECS),
National University of Sciences and Technology (NUST),
Islamabad, Pakistan.
(April 2012)

Approval

It is certified that the contents and form of the thesis entitled **"Generic Light Weight Certificate Management Protocol"** submitted by Muhammad Asif have been found satisfactory for the requirements of the degree.

Advisor: Dr. Abdul Ghafoor (DoC)

Signature: _____

Date: _____

Committee Member 1: Dr. Fauzan Mirza

Signature: _____

Date: _____

Committee Member 2: Dr. Hamid Mukhtar

Signature: _____

Date: _____

Committee Member 3: Mr. Qasim Rajpoot

Signature: _____

Date: _____

Abstract

Mobile devices and digital gadgets are very popular and commonly used in daily life. Research community increased its processing power and designed new advanced applications to attract business community to use it for e-commerce, business activities, sharing of valuable documents and many other sensitive activities. There are many problems in the development of secure applications for mobile devices. First, most of the users are not technical enough to configure security parameters and even already developed libraries do not support extended security features like transparent handling of certificates, verification of identities, and distribution of certificates. It is also observed that already developed security libraries are very difficult to use and integrate with existing applications to provide security features.

In order to solve above problems, a Generic Light Weight Certificate Management Protocol (GLCMP) is designed which is based on holistic approach in order to solve complex certificate management task. In order to achieve desired objectives, proxy based architecture has been adopted to offload computational intensive operations from mobile devices. In GLCMP, the trust between mobile device and proxy server is developed without exchanging any secret information on network. In addition, GLCMP designed and developed by using the concept of generic security objects.

The claimed security properties, authentication, confidentiality and non-repudiation of the protocol are formally verified by employing formal ***Z-Notation modeling.*** In Z-Notation modeling, different attacks are formalized on messages exchange between components and discussed all possible scenarios in which an attacker can attack the protocol. After verification, it is concluded that the designed protocol resists against most of the attacks launched on registration and verification process such as impersonation, man-in-the-middle and replay. Furthermore, for the proof of the concepts, the GLCMP is implemented and evaluate its result. Computed Authentication latency is 0.394 sec which is less than its nearest competitors NSI (4.7), PKI (5.01), and PKASSO (5.19 delegation time + 0.082 authentication times). Moreover, our design is also providing secure registration and certificate management.

Dedication

I dedicate my work to all my fellows.

Certificate of Originality

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by an-other person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at School of Electrical Engineering & Computer Science (SEECS) of National University of Sciences & Technology (NUST) or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at SEECS, NUST or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Author Name: **Muhammad Asif**

Signature: _____

Acknowledgment

First and foremost, I am immensely thankful to Almighty Allah for letting me pursue and fulfill my dreams. Nothing could have been possible without His blessings.

I would like to thank my parents for their support throughout my educational career, especially in the last year of my Master degree. They have always supported and encouraged me to do my best in all matters of life. I dedicate this work to them. I would also like to thank my brothers and sisters for their love and prayers for the successful completion of this work.

My cordial thanks are to my committee members, Dr. Fauzan Mirza, Dr. Hamid Mukhtar, Mr. Qasim Rajpoot, and all others who had contributed in any way towards the successful completion of this thesis.

Finally, this thesis would not have been possible without the expert guidance of my advisor, Dr. Abdul Ghafoor Abbasi, who has been a great source of inspiration for me during these years of research. Despite all the assistance provided by Dr. Abdul Ghafoor Abbasi and others, I alone remain responsible for any errors or omissions which may unwittingly remain.

Muhammad Asif

Table of Contents

List of Figures

List of Tables

List of Abbreviations

DN_U	Distinguish Name of mobile user
PU_U	Public Key of mobile user
PR_U	Private Key mobile user
NO_U	Nonce mobile user
R/V Server	Registration and Verification Server
IDMS	Identity Management System
PKASSO	PKI Based Single Sign-on protocol
WPKI	Wireless Public Key Infrastructure
LPKI	Light Weight Public key Infrastructure
SaPKI	Server Aided Public key Infrastructure
AVISPA	Automated Verification of Internet Security Protocol Analyzer
PCL	Protocol Communication logic
HOL	Higher Order Logic
CR	Certificate Request
H	Hash
E, ENC	Encryption
D, DEC	Decryption

CHAPTER 1

Introduction and Objectives

1.1 Introduction

Mobile devices and digital gadgets have become integral part of our daily life. Research community increased its processing power and designed new advanced applications to attract business community to use it for e-commerce, business activities, sharing of valuable documents and many other sensitive activities. Due to immense increase in its use a lot of security concerns have been highlighted like authentication, authorization, confidentiality and integrity. Public key Infrastructure (PKI) [1] is being used for achieving end-to-end security in desktop environment. But PKI is not recommended in mobile devices due to memory, battery and processing speed constraints. There are many problems in the development of secure applications for mobile devices. First, most of the users are not technical enough to configure security parameters and even already developed libraries do not support extended security features like transparent handling of certificates, verification of identities, and distribution of certificates. It is also observed that already developed security libraries are very difficult to use and integrate with existing applications to provide security features.

Second, certificate handling and execution of security functions are computationally intensive operations.

In order to solve above problems, a Generic Light Weight Certificate Management Protocol (GLCMP) is designed which is based on holistic approach in order to solve complex certificate management task. This protocol is further divided into three sub-protocols. These protocols transparently verifies users, generates, certifies and manages certificates for mobile devices based on well-established standards i.e. PKCS#10, Signed PKCS#7. The sub-protocols of GLCMP are identity registration protocol, verification and certificate management protocol. In order to achieve desired objectives, proxy based architecture has been adopted to offload computational intensive operations from mobile devices. In GLCMP, the trust between mobile

device and proxy server is developed without exchanging any secret information on network. In addition, GLCMP designed and developed by using the concept of generic security objects. These objects are easy to use by software engineers, easy to extend with new features and provide complete functions and features about a specific aspect.

1.2 Problem Statement

To design and implement a light weight certificate management that enables Public Key Infrastructure (PKI) in mobile devices to generate and manage certificates for trusted and verified mobile users, so they will be able to perform strong authentication, reliable communication and protection of resources.

1.3 Research Objectives

As discussed above in section 1.1 that the usage of mobile devices and digital gadgets is increasing in business activities, so devices are attractive victims of attacks.

Ethical hackers and security researchers have also gear up to find generic and dynamic ways for defeating the hackers.

The main objective of this research is to design a protocol that enables mobile devices to efficiently manage certificates with PKI to secure applications stored on mobile devices.

There are many problems in the development of secure applications for mobile devices. First, most of the users are not technical enough to configure security parameters and even already developed libraries do not support extended security features like transparent handling of certificates, verification of identities, and distribution of certificates. Secondly, observed that already developed security libraries are very difficult to use and integrate with existing applications to provide security features.

Thirdly, targeted low power mobile devices (have internet browsing ability) cannot appropriately perform computation intensive operations involved in Public key infrastructure.

In order to solve above problems, a Generic Light Weight Certificate Management Protocol (GLCMP) is designed which is based on holistic approach in order to solve complex certificate management task.

1.4 Research Methodology

Research is a structured enquiry that uses acceptable scientific methodologies to solve problems and create new knowledge that is generally applicable. Scientific methods comprises of systematic observation, classification and interpretation of data.

The approach we have adopted to conduct this research is briefly explained below.

Considering the limitations and constrains in mobile devices and digital gadgets regarding processing power, memory, battery power and security concerns, we propose a complete light weight certificate management protocol which provides secure registration of users, verification of identities, generate and manage certificates for mobile devices.

A thorough literature survey is done to understand the area and already proposed solutions. After that we have observed and analyze the results of the previously proposed security mechanisms and found some issues regarding authentication latency, extended security features like transparent handling of certificate, its management and distribution and complexity. To overcome all these issues, a security mechanism "Generic Light Weight Certificate Management protocol" is proposed. In GLCMP, computational intensive operations like creation of certificate request in PKCS#10 standard formats are offloaded to a domain level proxy server. Moreover, generic security objects have been used for developing the protocol.

Before implementing the protocol, we have formally verified the protocol by Z Notation modeling. In Z modeling, different attacks have been formalized on the exchanged messages between protocol agents and discuss all possible scenarios in which an attacker can attack the protocol.

After formal verification we have implemented the protocol and get the results. We have evaluated the results and compare them with previously claimed results of the protocol.

1.5 Overview of the Protocol

PKI is the most reliable infrastructure to ensure authentication, confidentiality and non-repudiation. Digital Certificates are exchanged for authentication because it is digitally signed by a reliable certificate Authority.

Creation of Certificate Request in PKCS#10 formats is very resource intensive because of involved cryptographic operations so it is not affordable for mobile devices. So certificate management is not efficiently being used in mobile devices. To overcome this issue, a Generic Light Weight Certificate Management protocol is proposed.

In this protocol, resource intensive operations are offloaded to a domain level certificate management server (proxy Server). Proxy server will authenticate the user by verification server.

The proposed protocol GLCMP is divided into three sub-protocols, Registration, Verification and Certificate Management.

Registration service is SSL secured and only domain users can access this service by terminal or its mobile device. User will register its information and its own secure password. When any user needs to generate a digital certificate, it will send its user ID to registration server. The registration server will authenticate and send distinguish name to the user.

The user generates its asymmetric key pair, concatenates its distinguish name with public key, takes hash of the concatenated message and sends to proxy server along with nonce.

In Verification sub-protocol, the proxy server transfers the same received message to Verifier server for verification. The Verifier server retrieves credentials from the IDMS register against the corresponding user identity. If verification is successful, Verifier sends ***"Accept"*** tag encrypted receiving nonce, its identity with user's password and sends to the Proxy server. On successful verification, Proxy server generates PKCS#10 with the collaboration of the mobile user.

In Certificate Management protocol, mobile user authenticate proxy server and sign hash of encoded Req. Info object with its private key. Proxy server integrates this signature with certificate request and sends to trusted CA server for issuing digital certificate. CA server verifies the user, sign the certificate and sends to proxy server. Proxy server sends the certificate to mobile user.

1.6 Research Contribution

- Proposed GLCMP protocol is light weight because it offloads computation intensive operations to proxy server.
- In GLCMP, the trust between mobile device and proxy server is developed without exchanging any secret information on network.
- Authentication latency of proposed GLCMP is 0.394 sec which is 91%, 92% less than NSI and PKI respectively. As for as PKASSO is concerned, If we include 5.19 sec delegation time, our result is 93% efficient but if we do not include it then our authentication latency is 79% greater as shown in Table 3, but here we are also providing secure registration.

- In addition, GLCMP is designed and developed by using the concept of generic security objects. These objects are easy to use by software engineers, easy to extend with new features and provide complete functions and features about a specific aspect.
- It also provides complete secure registration and verification and resists against most of the attacks launched on registration and verification process such as impersonation, man-in-the-middle and replay.

1.7 Thesis Organization

In chapter 1, brief introduction of research problem, overview of its proposed solution, research contribution, research objectives and research methodology are discussed.

Chapter 2 is dedicated to background of the proposal and literature review of the related area. Background of Formal verification, the methods logics used to formally verify the security properties of the protocol, is also presented in this chapter.

Motivation and scope of the research is covered in chapter 3.

Chapter 4 is covers the main components and abstract architecture of the proposed solution. It also covers the main function of each component and their integration with each other. Proposed protocol is explained in detail along with exchanged messages during each transition.

Chapter 5 is dedicated to design and analysis of GLCMP. Comparison of the results regarding authentication latency with previous proposed protocols is also depicted by table and graph.

In chapter 6, formal verification of security properties by using Z-Notation modeling and formalizing possible attacks such as man-in-the-middle and impersonation are discussed.

Conclusion and Future work is presented in chapter 7.

CHAPTER 2

Background and Literature Review

Due to the use of mobile devices in banking transactions and business, individuals and companies became more conscious for security. A lot of protocols and mechanisms have been proposed and implemented for achieving confidentiality, integrity, authentication and integrity in wireless communication. There are mainly two methods for achieving security services.

One is symmetric key cryptography (shared secret key) and other is asymmetric key cryptography (Public, Private Key) mechanism. We will mainly focus on asymmetric key cryptography.

2.1 Background

2.1.1 Public Key Infrastructure

Public Key Infrastructure (PKI) is a set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificate. In cryptography, a PKI is an arrangement that binds public keys with respective user identities by means of a certificate authority (CA). The user identity must be unique within each CA domain .The Public-Key Cryptography Standards are specifications produced by RSA Laboratories in cooperation with secure systems developers worldwide for the purpose of accelerating the deployment of public-key cryptography. First published in 1991 as a result of meetings with a small group of early adopters of public-key technology, the PKCS documents have become widely referenced and implemented. Contributions from the PKCS series have become part of many formal and de facto standards, including ANSI X9 documents, PKIX, SET, S/MIME, and SSL.

2.1.2 Digital Certificates

Certificate is a data structure that binds public key of an entity with Distinguish name and other optional attributes. Challenging password may be included in optional attributes which is used

later for certificate revocation. A third independent party signs this data structure. This certificate will be trusted by the entities have certificate from the same certificate authority. In certificate, it is guaranteed that specified public key is belonging to the certificate holding entity. The other information specified in the certificate is a unique certificate serial number, the name of the certificate issuer, name of the certificate owner, algorithm use to calculate the signature, a validity period and some extensions which are optional. Validity period is there to consider security issues.

2.1.3 Pkcs# 10

This standard describes syntax for a Certificate Request [3]. It comprises of a public key, distinguish name, possibly a set of optional attributes and collectively signed by the certificate requesting entity. This certificate request is sent to the Certificate Authority which after some verification transforms this request into an X.509 digital certificate. The optional attributes are very useful in finding other information of the entity like challenge password which may be used while certificate revocation. It may also require for the certificate Authority to ask for non-electronic certificate request and send non-electronic reply.

2.1.4 Kerberos

Kerberos is a network authentication protocol. A key Distribution Server (KDC) is employed for the management of secret keys. Kerberos uses shared secret key between corresponding servers and totally relies on KDC. The technique behind secret key is that both sender and receiver share a secret which is used for encrypting and decrypting the exchanged messages and data transferred in their result. Kerberos is recently being used in Microsoft Windows 2000 and onward for native network authentication. Kerberos is basically a mutual authentication protocol. A ticket called TGT is assigned to the requesting server after authentication. A session key is generated by the KDC to be served from the service server or Application Server.

2.1.5 Wireless Transport Layer Security

PKI is being used in mobile environment as it is being used in fixed network by introducing some modifications to cope with limitations being faced like low processing capability, low battery power and memory. The PKI enables secure m-commerce transactions via wireless devices and the provision of non-repudiation, which is often a requirement for m-commerce [4].

End-to-End security is being achieved by using application layer protocol like WAP (Wireless Application protocol). WAP v1.2 uses WTLS (Wireless Transport Layer Security) to protect messages exchange between wireless device and WAP gateway. Contents must be decrypt and encrypt while passing through the WAP gateway because WAP and WTLS are incompatible. This requires that there must be a chain of trust between the client and WAP gateway and similarly from originating server to WAP gateway. Because of this chain end-to-end security cannot be achieved from client to web server. WAP 2.0 is enhanced version of WAP1.2 which do not use WAP gateway because it support network and transport protocols. It means that mobile devices support TLS to web server without need of decryption at WAP gateway.

2.1.6 Wireless Public Key Infrastructure

Wireless public key Cryptography (WPKI) is an improved version of PKI in wireless communication. The exchange process in Mobile Electronic Commerce requires confidentiality, data integrity, authentication and non-repudiation [5]. It is only use for WTLS security requirements. WPKI infrastructure comprises of the same components as of PKI like end-Entity (EE), Registration Authority (RA), Certificate Authority (CA) and certificate repository. However, EE and RA implementation is different and a new entity PKI portal is introduced. In WPKI EE is deployed on WAP device but function is same.PKI portal work as registration authority and translate requests from WAP clients to certificate authority. WPKI also introduces optimized format of certificate, crypto algorithms along with keys.

2.1.7 The mSign Approach

The mobile Electronic Signature association [6] of companies concern with mobile security issues has published mSign protocol. This association intends to make this protocol a standard for interoperable mobile signatures. In this publication, a lot of message types, formats and security levels have been defined. Since crypto capabilities of different mobile clients are different so a new idea is introduced in mSign protocol to generate signature according with. Now it is possible to generate a signature on the client or by trusted party on behalf of original user.

2.2 Related Work

2.2.1 PKI-based Single Sign-On Protocol

Ki-Woong park et al, have presented a security infrastructure called PKASSO which consists of five main components. PKINIT is one of the main components, enhanced form of Kerberos, which encompasses Kerberos, LDAP and CA servers. A service requesting entity, mobile user, for authentication, authorization and accounting should have smart card like device called PANDA equipped with low powered Zigbee [7] for intercommunication and location sensing. Third component is a service device with Zigbee for communication with the users and Ubiquitous Fashionable Computer (UFC) [10]. Fourth component is a delegation server for maintain the entire proxy certificate along with public private keys delegated and signed by users. In start user delegates it's all authentication process to delegation server according to RFC3820 [8]. Delegation server performs its responsibilities till certificates remain valid. The last and the most important component is referee server which is assigned the duty of investigating authentication messages and binding these messages with users. This server provides the functionality of non-repudiation. According to this mechanism, service device spread its beacon, challenge message, to service required User. User perform mutual authentication by using PKI and then delegate other authentication operation to delegation server. For this, User create proxy certificate with public key sent by Delegation server, sign it by its private key and send to Delegation server. After successful delegation, user encrypt received challenge message from service device with AES twice and send it to Delegation server. Delegation server verifies this message from referee server and after successful verification performs PKINIT [9] operations. In PKINIT operations, delegation server gets TGT, Ticket Granting Ticket, by PKI and SGT over Kerberos. Delegation server wills response message to Service device and User will send confirm message to Service device. This protocol achieve, Single Sign-on, Digital Signature, Authentication, Non-repudiation and secure key distribution on the cost of 0.082 sec authentication latency.

2.2.2 Server aided PKI

Every mobile node has to process intensive computational cryptographic operations. To offload work from clients, Server-aided PKI infrastructure service (SaPKI) is deployed in GSM and CDMA networks. SaPKI is based on Modadugu's key generation protocol [11] and Asokan's S^3

[12] protocol. Modadugu et al. enhance efficiency of RSA key generation for low power handheld devices even with support of untrusted server.

SaPKI combines the encryption and key exchange capabilities of Modadugu's protocol and digital signature generation capability of the S^3 protocol thus it provides the full PKI infrastructure service for mobile device applications. SaPKI is useful for mobiles in two ways. First SaPKI have access to hardware use for cryptography that is capable of performing single cryptographic function faster than client. Second, because of offloading cryptographic operations, client is free for other operations.

There are three key interfaces to offload the intensive computation from mobile nodes to SaPKI server.

1- ISaPKI_KeyGen () help mobile client generate the key used for key exchange and key generation.
2- ISAPKI_Cert () provide help to initialize client key used or generation digital signature.
3- ISAPKI_sign () provide help for signing the message.

Architecture of SaPKI is comprises of three main components, CA, admin utilities related to CA like policy Server responsible for system-wide security and Billing Server for accounting and client. Each SaPKI service serves many clients in turn. The designation of client to SaPKI is offline by system security administrator. A client may also be served by many servers at the same time.

Libraries present in client component support SaPKI operations. The architecture of SaPKI is designed for cell phone networks such as GSM/CDMA. It is only useable for cell phones when in touch, via a nearby base station. Phone-call need communication with infrastructure can be used to carry messages require for SaPKI protocol.

2.2.3 Light Weight Public key Infrastructure (LPKI)

LPKI introduces secure infrastructure for computation-constrained platforms like mobile devices and digital gadgets [13]. It minimizes computation cost and communication overhead by using Elliptic Curve Cryptography (ECC) and signcryption. The *Elliptic Curve Cryptography* (ECC) is usually deemed as a suitable solution for the resource-constrained devices [14]. As an example, it is believed that a 160-bit key in an elliptic curve-based system provides the same level of security as that of a 1024-bit key in an RSA-based system [15].

LPKI delegates the validation computation intensive operations to validation authority. LPKI consists of Registration Authority, certification Authority, OCSP server, Validation Authority, Key Generation Server, Timestamp server and End-Entities. All the exchanged messages between servers are encrypted with symmetric key encryption by using AES and secret key is generated by using public key of each entity. Foe exchanging keys, LPKI takes the advantages of HMQV key exchange protocol [16] because of its efficiency, standardization and a lot of provided security attributes. Now a day's both smart cards (e.g. SIM) and hand held devices have suitable capacity so these is no problem to manage X.509 certificates especially in a case where one certificate is associated with one subscriber.

Any public key will only be trustworthy if it is verified by the certificate and all certificate validation is delegated to VA server to offload intensive operations. OCSP server will keep update VA server about the expiry status of the certificates.

2.2.4 MP-PKINIT

M-PKINIT is a lighter version of PKINIT introduce minor modification and used in mobile devices. It considers that the client knows the public key of KDC, so client generates session key and encrypt it with KDC's public key. In this scenario there is one problem that KDC may reject the session key if it does not meet the KDC's key policy. Client will do the same operation for building trust relationship with proxy server which is an additional node in this protocol for caching client's certificate chain. This technique eradicates the need of sending certificate on wireless network.

2.2.5 Towards the Efficient PKI for Restricted Mobile Devices

Jalali-Sohi and P. Ebinger solve the mobile node's issue of searching and verifying digital certificates, which was not possible due to computation power, battery and memory constraints, by using PKI-Server in the authentication infrastructure. Mobile node delegates its responsibility to PKI-server which is defined in [17]. PKI-Server provides some type of APIs for clients which help clients to be unaware from most of the involved complex PKI operations. In this method, minimum achieved authentication latency is 4.75sec. This protocol is not feasible because its authentication latency is much higher and main PKI operations are performed on mobile nodes.

2.2.6 Analysis of the Related Work

- WPKI is not suitable for mobile devices because of processor intensive jobs like certification path verification, computational cost and storage capacity. Moreover, it uses two pair of private/public keys.
- In Charon, Client is requesting to Proxy for TGT from KDC. In next phase Client is requesting to Proxy a session key to building trust relationship with same Proxy from TGS (Ticket Granting Server). If there is no trust build with Proxy then how it request to Proxy for TGT and Session keys. Moreover, Proxy is just middle man between Client and Kerberos.
- In NSI, authentication latency is 4.75sec which is not appropriate.
- To overcome the limitations of NSI, PKASSO is proposed. But PKASSO is not feasible for mobile devices because of its complexity and implementation problems. Moreover certificate which is signed by mobile user does not contain user's public key but of delegation server.
- M-PKINIT does not provide digital signature and non-repudiation which is a great security lap.

A comparison regarding authentication latency is shown below [2].

Sr.#	Protocols	Security Services			
		Authentication	Non-Repudiation	Digital Signature	Authentication Latency
1	Kerberos	Yes	No	No	0.19 Sec
2	PKINIT	Yes	No	No	1.21Sec
3	M-PKINIT	Yes	No	No	o.74 Sec
4	NSI	Yes	Yes	Yes	4.70 Sec
5	PKI	Yes	Yes	Yes	5.01 Sec
6	PKASSO	Yes	Yes	Yes	0.082 Sec

Table 1: Security Services and Authentication Latencies of the protocols

- In SaPKI, RSA algorithm is being used for key generation and signing. Moreover, it provide three interfaces, ISaPKI_KeyGen ( ), ISaPKI_Cert ( ) and ISaPKI_Sign ( ) for each function. In this approach all the computation intensive operations are being performed on servers but for establishing trust client has to generate a session key and encrypt the message by server's public key using RSA which is again intensive operation.

- LPKI is the counterpart of PKI; Elliptic Curve Cryptography (ECC) is used rather than modular exponentiation-based counterpart RSA which has great computational advantage. It is basically a totally different infrastructure parallel to PKI [26].

Mobile Device	**Signature generation**		**Signature Verification**	
	ECDSA	RSA	ECDSA	RSA
Nokia 6610	2.294	74.682	4.382	2.825
Siemens S55	18.963	883.602	35.277	30.094

Table 2: A Comparison between execution times (ms)

In LPKI, asymmetric encryption for signing, symmetric encryption with PKC for generating session keys and signcryption whenever both digital signature and encryption simultaneously required are used which do not seem as light weight.

2.3 Formal Verification Methods

Formal verification is the way of testing correctness and properties of the protocol what it is claiming. More exact and detailed information is required for the verification of security protocols than normal communication Protocols. Security protocols needs to be verified regarding their security, efficiency and claiming characteristics before its deployment because a minor flaw may cause unacceptable results. The bases of these procedural verifications are formal methods, procedural analytical rules to prove whether the protocol is secure or not.

Usually Informal standard notion is employed for describing security protocol [20]. However, this notation remains ambiguous in isolated interpretation and provides no reasoning and semantic principles. Contrary to this, formal methods provide exact specification and verification of computer systems. Therefore, numerous efforts have been undertaken for promoting formal methods for verifying correctness of security protocols [21]. But non-functional concepts of security like confidentiality and authentication have been found difficult to analyze and prove. Formal verification of security protocols is very important before its implementation because several, considered secure protocol have been found later vulnerable to attacks [22].

2.3.1 Classifications of the Formal Methods

There are four main classes of formal methods employ to formally verify the security protocols.

Inference-constructed

This method bases on logic and belief. After application of postulates and employing logical rules, we infer some beliefs which show agreement and disagreement with protocol's initial specification. BAN Logic, its variant GYN Logic, Higher Order Logic Theory (HOL), SvO Logic (Combination of BNa, & GNY, VO Logics) etc. are the common examples of this class.

Proof Construction

In this method the considering protocol is defined in reliable and unreliable principles. Roles and beliefs of each participant is precisely separated which lead to prove or disprove the secrecy of the protocol. This method gives human-readable results at the end.

Attack Constructed

Attack constructed is the third type of formal methods in which provable attacks are constructed from the algebraic properties of the protocol. In this technique all possible states of the protocol are verified with attack point of view which is also a disadvantage due to a large number of possible states.

Expert Systems

Expert systems are basically independent of protocol. The testing protocol is first described in a specific language. NRL protocol analyzer and Interrogator are good examples of this method. However formal procedures of techniques are different to prove a protocol secure or insecure.

2.3.2 Process of Formal Methods

It is the process of verifying system behavior of security protocols by using formal semantics. The process of formal methods comprises of following three tasks.

Formal Specification – descriptive

Formal Specification summarizes system into high level mathematical model. It is used to catch ambiguous and inconsistent specification in high level designs. It can be different types such as model oriented property oriented, and logical.

Formal Synthesis-layered approach

It is incremental top-down refining process to split large verification tasks into smaller tasks. Since each refined task is verified individually by powerful verification algorithms and allows more, so it is costly.

Formal Verification – analysis

In Formal Verification – analysis, given specification are analyzed and provide minimum changes in the development process. Since a full verification is difficult, so it requires simple and focus on partial specification like type safety, functional uniformity of two systems.

2.3.3 Formal Verification and Simulation

It is a good approach to simulate a model of actual or theoretical system before executing it in production environment. The output of the simulated model can be analyzed for acquiring solid results of actual systems. There are three primary sub-fields in simulation: model design, model execution and model analysis.

2.3.4 HERMES

It is the dedicated secrecy verification tool for cryptography protocol. It uses special purpose front-end language called EVA. First of all protocol is defined in standard formal notation and then transformed into EVA scripts keeping assumptions on all exchanges to be verified as security property. In HERMES, it is to be ensured that no message should be public. If a message is deduce from a set of intercepted messages known to intruder or publically accessible then a message will be consider public.

HERMES is based on protector messages to guarantee protection of the secrets. These messages are encrypted so intruder cannot access them. Suppose there is a message S and hypothesis H. On one hand, a set of messages is calculated which provides security to normal messages during its execution. On the other hand attack messages are constructed. HERMES provides a set of secret messages S and protected messages H at the end of execution. These two sets provide help in enlightening secrets without any danger.

There are four parts in EVA scripts, declarations, Initial Knowledge, Message Knowledge and Session & Secrets. Declaration section contains variables, algorithms, key types, principals and Initial Knowledge. Last section consists of sessions and goals that lists all the secrets and sessions respect to all principle and values.

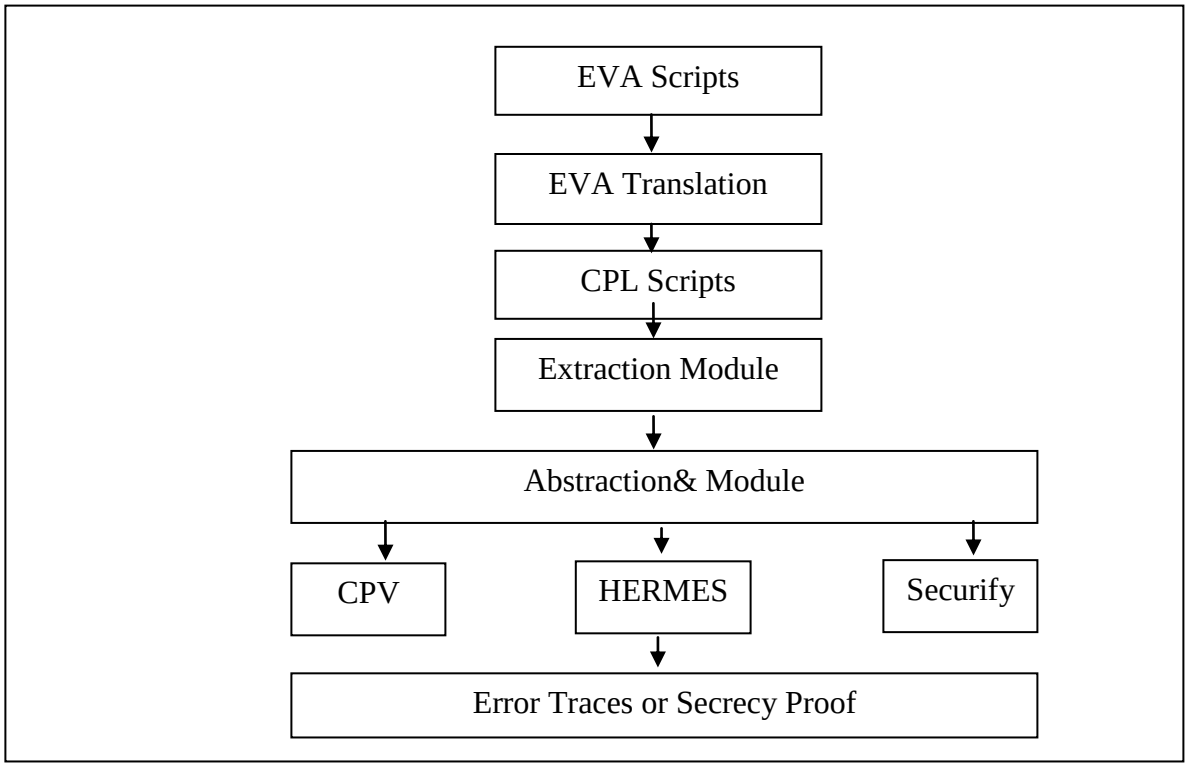

Figure 1: Functional Architecture of HERMES

2.3.7 Z Notation

It is an intuitive based notation used to formalizing and verifying the correctness of security protocols. It divides the whole verifying process in phases. Boyd and Kearney [21] explored protocol animation using Z for fair exchange protocols. For the purpose of modeling fair exchange protocols, each agent considers the other to be the intruder.

Z based notation systematically translate the informal standard into model of Z. It actually captures all features of the protocol.

2.3.8 NRL Protocol Analyzer

US naval research Laboratory is the founder of this tool. This tool was developed for analyzing the security of the protocols. It is special-purpose prototype verification tool that is written in prolog and developed for verification and analysis of the cryptographic protocols. NRL protocol analyzer can also be used for finding security flaws in the protocol.

2.3.9 AVISPA

Automatic Verification of Internet Security Protocol and Applications is a validation tool for Internet Security Protocols. It operates as automatic push button. It is developed as a funded project by European Union. It encompasses first five OSI layers for all security protocols and provides about twenty security services and mechanisms. It also covers about 85% of IETF security specification that is verifiable by it. AVISPA library is available online which is verified with code by hundred problems of more than two dozen protocols. Key Management Architecture for Hierarchical Group Protocols has been verified using AVISPA tool in [22] and the Cross-Layer Verification of Type Flaw Attacks on Security Protocols was also detected as in [23].

A high level protocol specification language (HLPSL) is used in AVISPA to feed a protocol in it. HLPSL is an extremely expressive and intuitive language for modeling of a protocol. A state is a system behavior in HLPSL. There are variables for each state that are responsible for each transition; that is when a variable changes, the state turn to new form.

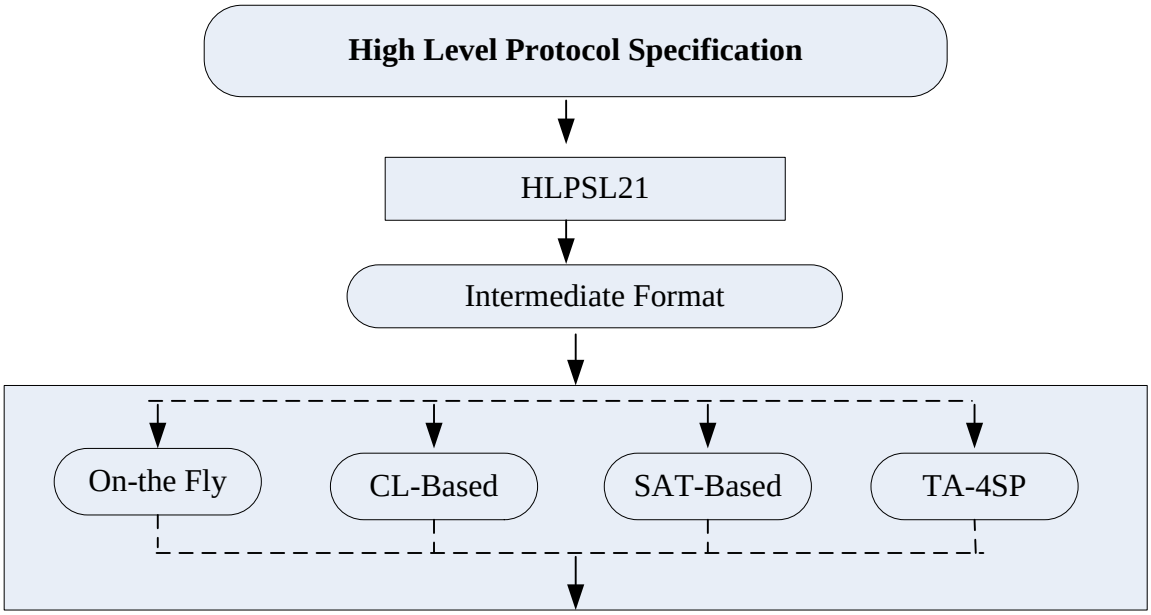

Figure 2: AVISPA Tool Architecture

2.3.10 Protocol Composition Logic (PCL)

Network security protocols, such as key-exchange and key-management protocols, are difficult to design and debug. Protocol Composition Logic (PCL) is a logic that is used to verify security properties of the protocols such as key-exchange and key management protocols. It is developed

for the protocols that are difficult to design and debug. For example, the 802.11 Wired Equivalent Privacy (WEP) protocol, used to protect link-layer communications from eavesdropping and other attacks, has several serious security flaws [24]. A lot of predicates, such as Send(X, t), Receive (X, t), New (X, t), D encrypt (X, t), Verify (X, t) are used in PCL.

A → B: m

B → A: n, SIGB {|n, m, A|}

A → B: SIGA {|n, m, B|}

CHAPTER 3

Motivation and Scope

3.1 Research Motivation

Mobile devices and digital gadgets are very popular and commonly used in daily life.

E-commerce, business activities, sharing of valuable documents, information and many other sensitive activities have started to be performed on mobile devices. The more use of mobile devices in communication the more number of attacks.

Emerging ratio of security issues in mobile communication like authentication, confidentiality, non-repudiation become motivation to find a way to evade from such issues. So that, trust of mobile users can increase and business on mobiles become viable.

3.1.1 Why PKI?

Symmetric and Asymmetric key cryptography are two ways for achieving authentication. Asymmetric key (Public / Private Key) is used in public key infrastructure (PKI). It is the most essential mechanism for open and popular systems.

A PKI (public key infrastructure) enables users of a basically unsecure public network such as the Internet to securely and privately exchange data and money through the use of a public and a private cryptographic key pair that is obtained and shared through a trusted authority [19].

PKI is used in digital certificates which are issued by a trusted party so that individuals can build reliable trust before communication. An individual can authenticate the other by verifying the whole chain of certificate authority.

In addition, the frequency of user authentication requests increases rapidly in proportion to the number of the service devices that require mutual authentication [25].

PKI is efficiently being used in desktop environment but facing some problems in mobile devices because it involve resource intensive public key cryptography operations. Since mobile

devices have less processing power, memory and battery power, so its response time is unaffordable. This delay is not acceptable in business environment.

3.1.2 Generic and Light Weight Protocol

As explained above the facing problems for deployment of PKI in mobile devices, it is need to design a security infrastructure that cope with all said issues. This need entice to design GLCMP that is generic and light weight and can be used in mobile gadgets having less memory, computation and battery power.

The previous proposed solutions mostly use Kerberos for establishing trust relationship. Some protocols use PKINIT, a variant of Kerberos use public/private keys for authentication and confidentiality. MP-PKINIT is the lighter version of PKINIT which is specially designed for mobile devices. But this protocol does not provide non-repudiation.

While, In GLCMP, the trust between mobile device and proxy server is developed without exchanging any secret information on network. In addition, GLCMP designed and developed by using the concept of generic security objects. These objects are easy to use by software engineers, easy to extend with new features and provide complete functions and features about a specific aspect.

3.2 Research Scope

It is necessary to define the scope and limitations of the research before starting. It keeps researcher focused and streamline. This scope and limitations depends upon various parameters like time, demand with respect to society and organization, strength and funds. For ease of understanding, we categorized our scope into three parts.

3.2.1 Mobile User Authentication

The main barrier in mobile business communication is slow response time and authentication latency. So, it is necessary to design a protocol which comes up with solutions to address above said issues. In GLCMP, we tried to achieve reliable authentication on the cost of affordable authentication latency and response time. We adopted a holistic approach in which no public / private keys or any credential is sent on network. Secret password is used to build trust which securely registered during registration process.

3.2.2 Creation of PKCS#10

As discussed above the importance of certificate usage in communication, it is need to implement efficient use of certificates. In certificate request process, creation of PKCS#10 is very resource intensive due to cryptographic operations and encodings. So, the next goal of research is to adopt a mechanism which offloads such intensive operation from mobile device. For this purpose, proxy based architecture is adopted. Many proposed solutions used this approach but for authentication not to offload heavy computations.

3.2.3 Management of the Certificate

Management of the certificates was also discussed in research work. Normally mobile devices do not allocate a special container for storing and using certificates. It is part of our research to find a secure way for storing and exchanging the certificates. Certificate path verification and certificate revocation and causes of on-demand revocation is not in our research scope. These goals can be considered in future work.

3.3 Aims of Proposed Solution

- Light weightiness.
- Generic in nature to be used for multiple platforms.
- Design and develop by using the concept of generic security objects.
- Offloading of the computation intensive operations.
- Authentication without exposing any credential.
- Use minimum servers to obtained maximum results.

CHAPTER 4

Proposed Generic Light Weight Certificate Management Protocol

4.1 Abstract Architecture of the System

The purpose behind to propose this protocol is security, easy to implement and offloading the processor and memory intensive functions involve during certificate request generation.

4.1.1 Components of the System

Mobile Device

A mobile node needs to be authenticated for generating and managing certificate. Mobile can also be used for registering information by accessing SSL secured registration service but in this case there will be a security policy on registration server to authenticate the mobile user that is domain credentials.

Management Terminal

A terminal used for inserting identities and other information of clients manually. This terminal is basically part of same network as secure domain and only domain members can access secure registration service on this terminal.

Registration/Verification Server

An interface (service) used for registering information of the user and verifying the user whose request is received from proxy server. This service is secured by implementing SSL and is also domain member. IDMS (Identity Management System) is directly accessible to this service. Only domain members can access this secure service. There is a trust relationship between Domain Level Certificate management Server (Proxy) and Verification Service. Secure password of the user is only share between this service and the user itself.

IDMS

A database contains the identities and demographic information of the clients. This database is only accessible to Registration/Verification Service.

Domain Level certificate Management Server

This is just like proxy server used to offload processor intensive functions from Client. It has no credentials of the mobile users. There is a trust between proxy and Registration/Verification server. Proxy server totally depends on verification server regarding verification.

4.1.2 Diagrammatical Overview of the System

The components of the proposed system interact with each other as shown in the figure.

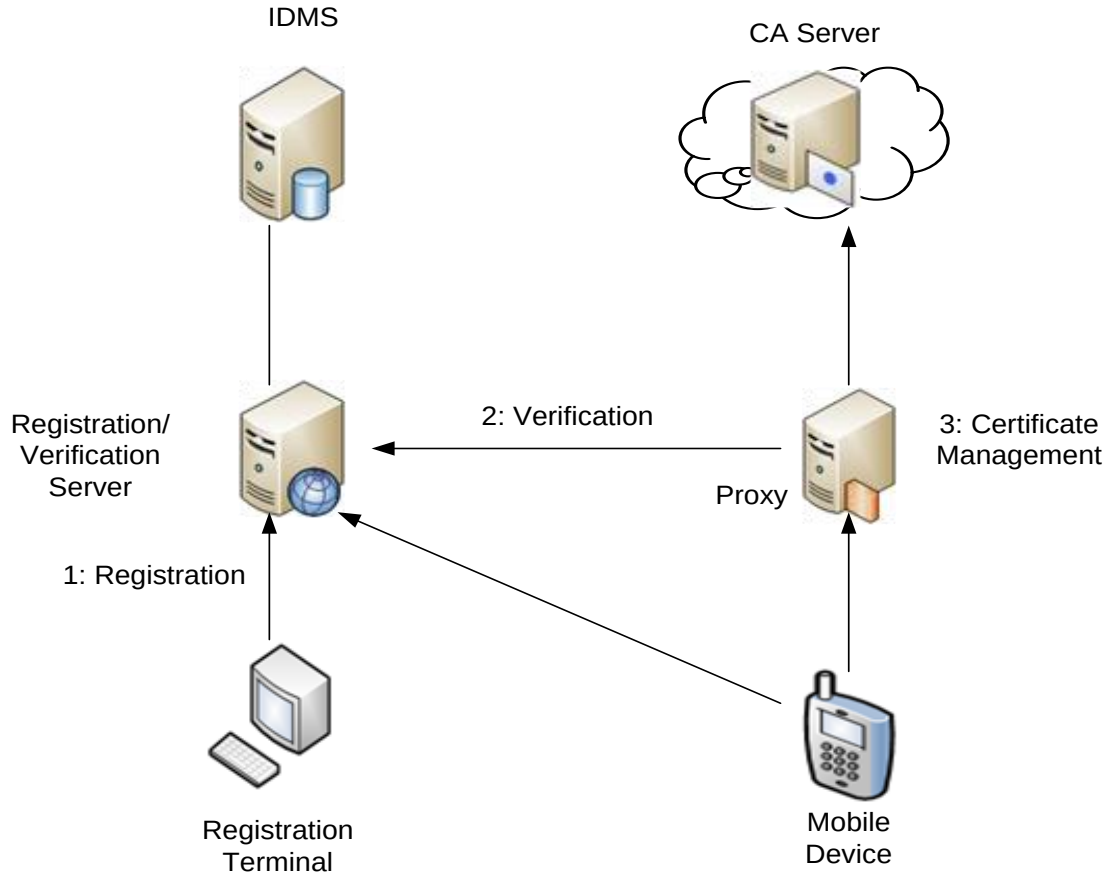

Figure 3: Abstract Architecture of the system

To reduce complexity we divide our protocol into three sub-subprotocols, Registration, Verification and Certificate Management.

4.3 Components of the Protocol

The proposed protocol is divided into three components, Registration, Verification and Certificate management, due to its complexity because it is being designed for light weight mobile devices.

U: Mobile user

R/V Server: Registration/Verification Server

P: Domain Level Certificate Mgt. Server (Proxy)

C: CA Server

I: IDMS

PU_U: Public Key of Mobile user

PR_U: Private Key of Mobile user

SP_U: Secret Password of Mobile user

H: Hash Value

DN_U: Distinguish Name of Mobile user

E: Encryption

D: Decryption

4.3.1 Registration

Assumptions:

IDMS, Registration/Verification Server, Registration Terminal and mobile users are members of the secure domain.

Only authenticated users of the domain can register. There are two ways of registration. One is manual registration by visiting the help desk team and second method is from mobile device by accessing secure web service. If a user tries to register by its mobile device, it must be first authenticated by using its domain credentials.

- SSL is implemented on the registration service.
- SSL client and server establish a secure SSL connection on the guarantees of following security properties.
- Private connection is established because each message is sent in encrypted form.
- Peer is authenticated by public/private keys.

- Integrity is confirmed by using message authentication code (MAC) computed by data and shared secret.

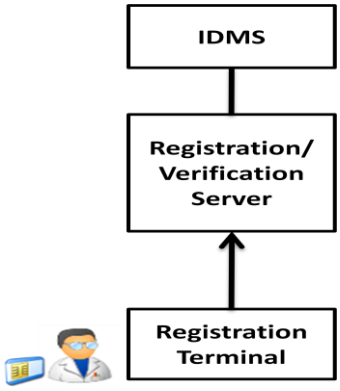

Figure 4: Registration of the mobile user.

User will access secure web interface of the service and register its information and secure password.

M1: $(Info)_U \mid SP_U$

M1 is the first message of the protocol and SP_U is secure password.

Mobile user sends request to fetch registration data from Registration/Verification server (R/V Server) that is require for certificate request and authentication. First user generate a onetime nonce, concatenate it with hash of its user identity, encrypt it with its secure password and concatenate encrypted message with plain text user identity ID_U and send to the R/V Server as shown in message M2 and Figure 5 (a).

M2: $ID_U \mid E\,[SP_U, (NO_U \mid H\,(ID_U)]$

R/V Server retrieve required password from IDMS against received user identity, decrypt the message. Then the Server calculates hash of user identity ID_U and compares it with received hash value. If both hash matches, integrity is confirmed. It is depicted in Figure 5 (b).

After successful integrity check, server creates distinguish name $DN_{U,}$ concatenate with its identity and received nonce of the user, encrypt it with user password. Server then concatenate this encrypted part with plaintext distinguish name DN_U and sends to the mobile user as shown in message M3 and figure 6.

M3: $DN_U \mid E\,[SP_U, (NO_U \mid ID_R \mid H\,(DN_U)]$

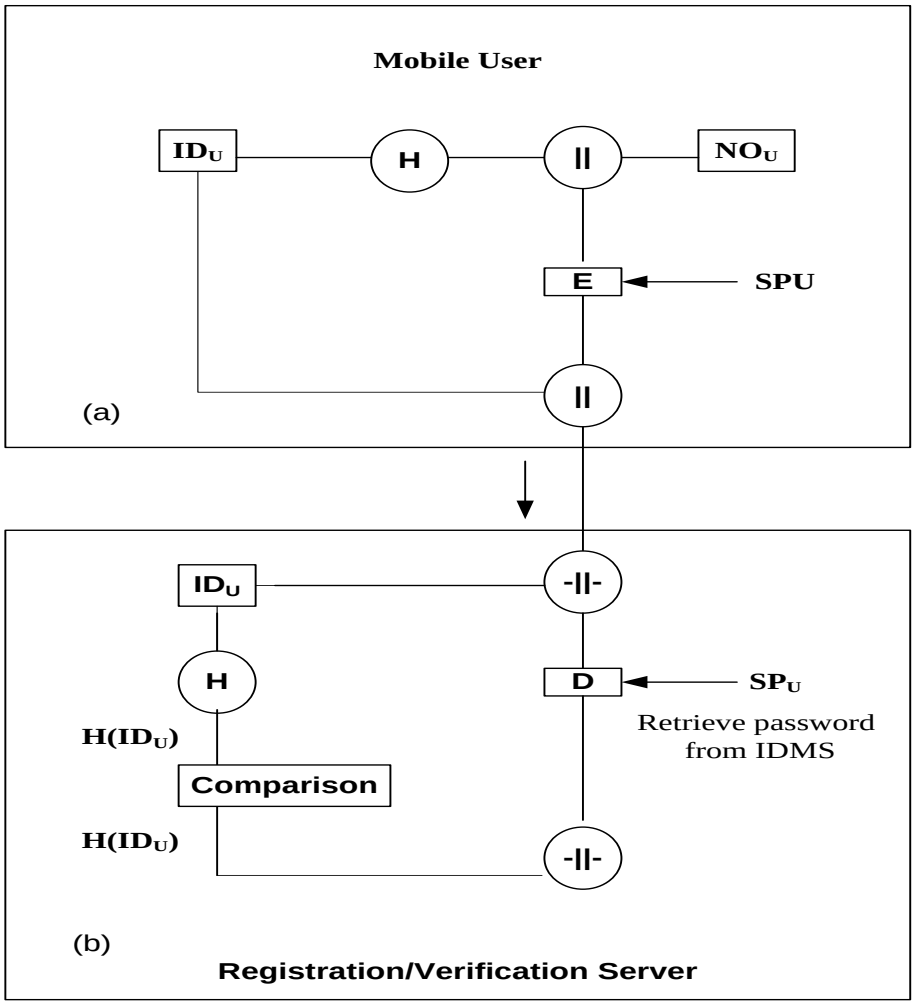

Figure 5: Request for fetching distinguish name from R/V server

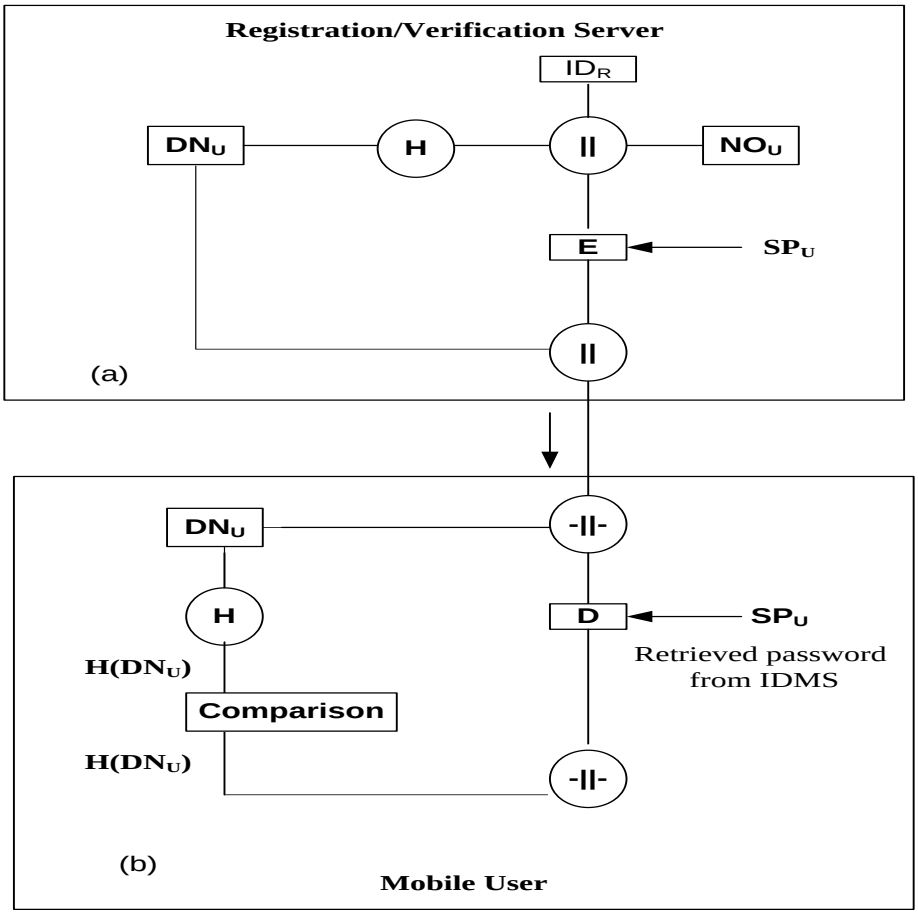

Figure 6: Process of Creating distinguish name

On receiving message M3 from the Server, Mobile User decrypts the message with its password compares nonce and hash with the calculated hash of DN_U to ensure integrity as shown in Figure 6 (b) and store the received ID_R in safe place. This ID_R will be used to avoid Proxy-Bypass attack.

5.1.2 Verification

If integrity is ensured, user creates asymmetric key pair of (1024) by using RSA, takes hash of its public key and distinguish name, concatenate it with nonce of User, encrypts it with password and sends to Domain Level Certificate Management Server (Proxy) to create certificate request as shown by the message M4 and Figure 7 (a).

M4: $(PU_U | DN_U) \mid E\,[SP_U, (NO_U | H\,(PU_U | DN_U)]$

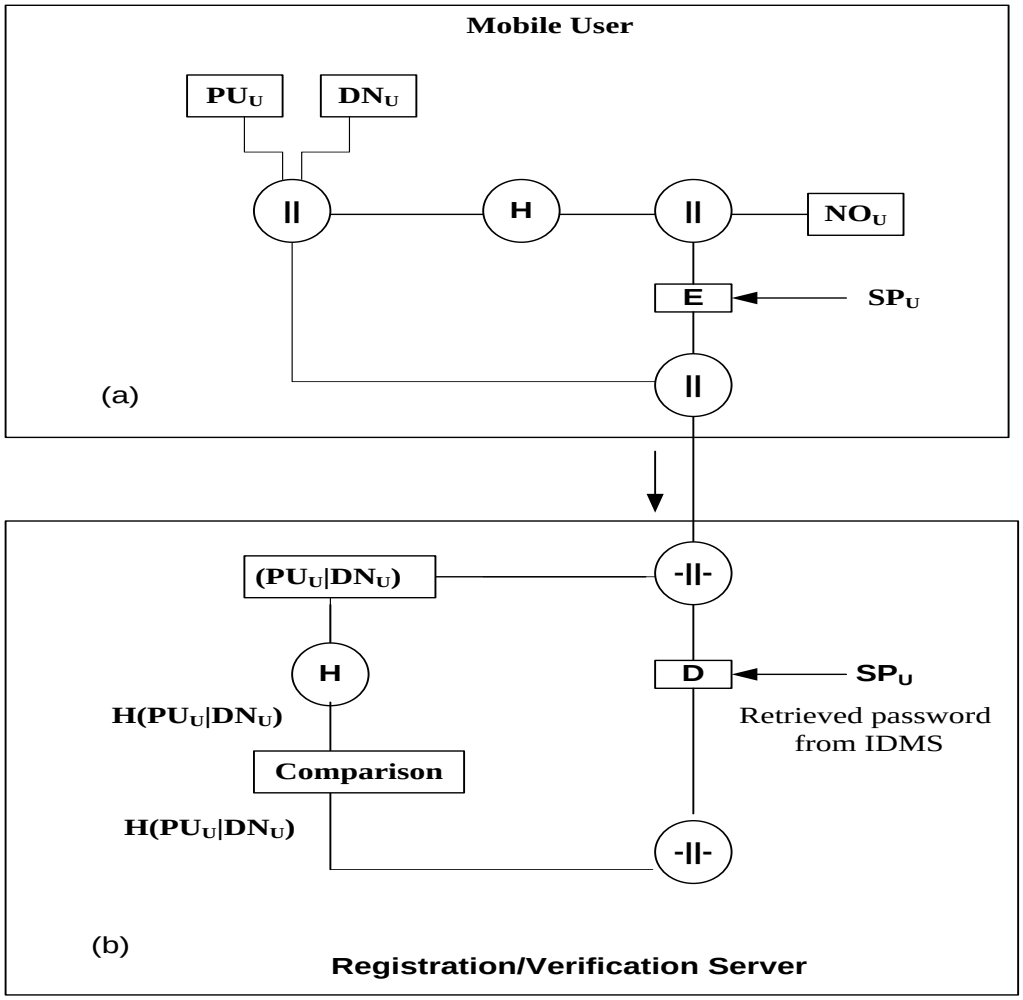

Figure 7: (a) Message M4 and (b) verification check at R/V Server

For verification, the Proxy Server forwards the same message to R/V Server as shown by message M5.

M5: $(PU_U| DN_U) | E [SP_U, (NO_U|H (PU_U| DN_U)]$

For verification, R/V Server take user identity from distinguish name, retrieve corresponding password from IDMS, decrypts the message, and compares the hash value with calculated hash of (PU_U| DN_U) for ensuring the integrity of the message and authentication. If authentication is successful, R/V server sends *Accept* tag along with encrypted message consist of received nonce, ID_R of registration server otherwise reject message to Proxy Server.

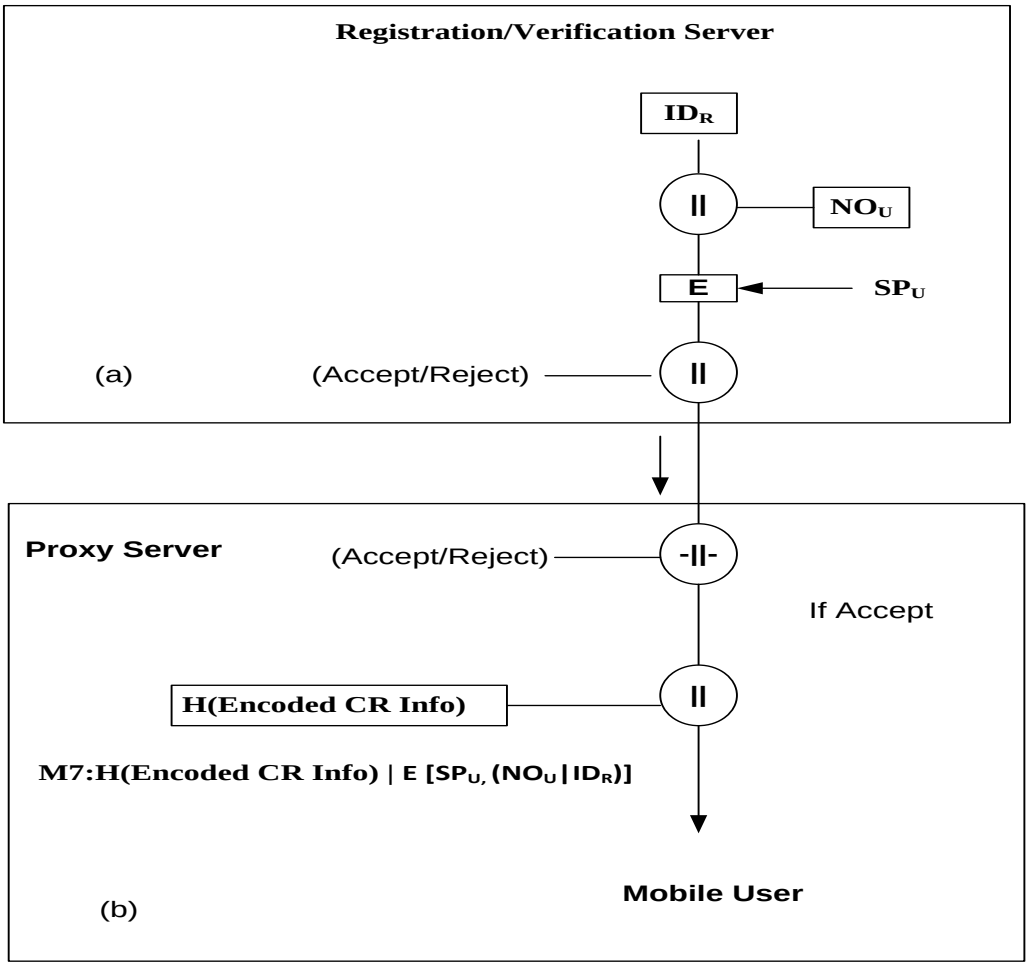

Figure 8: Verification of the mobile user

M6: [Accept | E [$SP_{U,}$ (NO_U | ID_R)]

After successful verification, Proxy Server integrate distinguish name (subject) with subject's public key, create Certificate Request Info (CR Info. Value) object in DER encoded form, takes its sha1 hash, concatenate it with encrypted message received from Verification server and sends to Mobile user for signing.

M7: [H (CR Info Value) | E [$SP_{U,}$ (NO_U | ID_R)]

User (Mobile Device) decrypts the message, and verifies its nonce and ID_R of verification server. If integrity is ensured successfully, User signs *H (CR Info object)* with its private key to avoid non-repudiation as shown in M8 and send to the proxy server for generating PKCS#10

M8: E [PR_U, H (CR Info object))]

5.1.3 Certificate Management

Proxy Server integrates the signature to form certificate request in PKCS#10 format and sends to CA Server for issuing digital certificate.

M9: [Signed (PKCS#10)]

CA server performs it formal operations and transforms the request into an X.509 public key certificate [10] by following the format standard PKCS#7 [11] and sends it to Proxy Server P.

M10: [Signed Digital Certificate by CA]

Proxy server forwards the certificate to mobile user.

M11: [Signed Digital Certificate by CA]

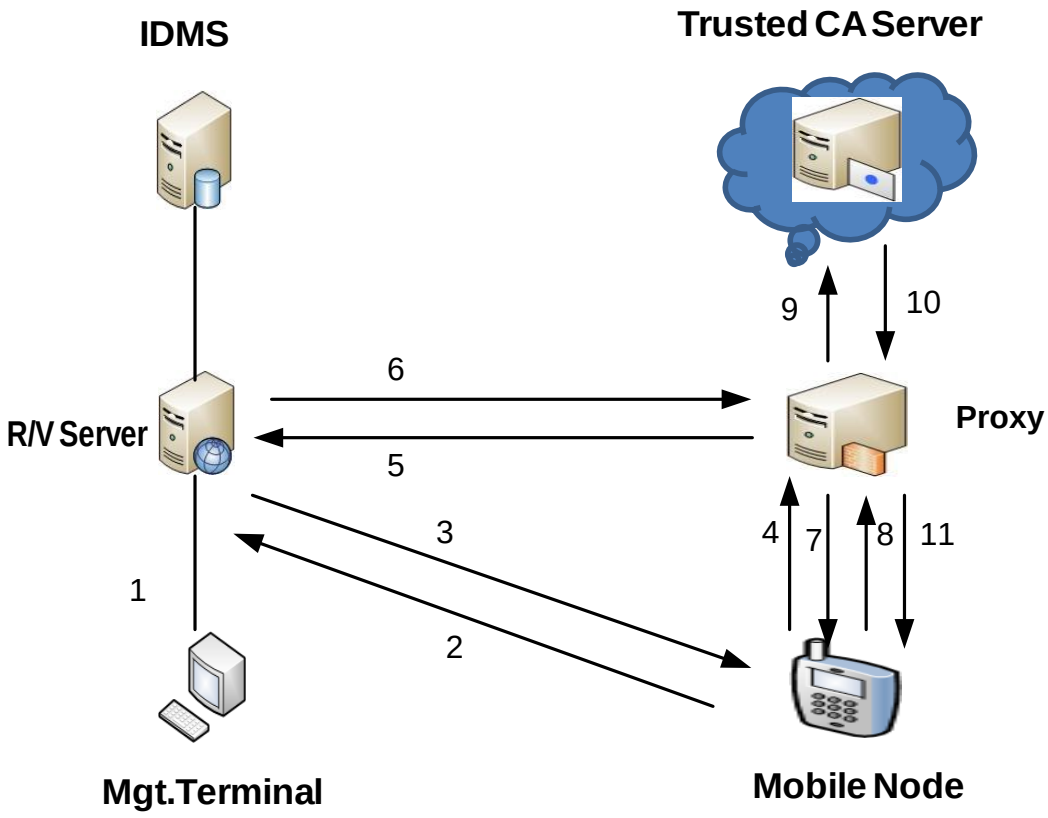

Figure 9: Overall System Architecture

CHAPTER 5

Design and Analysis

5.1 Design of the System

The proposed protocol GLCMP is designed for an organization to integrate with domain environment. It is designed by keeping in consideration light weightiness, security properties like authentication, authorization, confidentiality, integrity and non-repudiation. It is tried to maintain a balance between security level and light weightiness. Security is not compromised at any level. Security of each sub-protocol like Registration, Verification and certificate Management is taken into account separately. Standard Secure socket layer (SSL) is implemented on registration service. User can register by two ways. First is by visiting help desk terminal and second is by mobile device. During second method there will be domain authentication process to avoid any unknown user to register.

To eliminate dependencies on mobile phones to keep personal information, user fetch it's distinguish name from registration server for generating certificate. If mobile phone is lost or stolen user do not need to register again. He just needs to revoke prior certificate and create a new one.

In generating certificate, creation of certificate request in PKCS#10format is very difficult because of involved cryptographic computation intensive operations and complicated structure. To offload this task, domain level proxy server is deployed. But to avoid non-repudiation, hash is signed by the mobile user.

Each exchanged message is encrypted with user password by using DES encryption. Integrity is ensured after each message transmission. To avoid replay attacks and get reliable authentication, encrypted nonce is sent in each message. If received nonce does not match with the sent nonce, trust will not be established.

Fetching and Verification processes are interlinked to authenticate verifier and avoid proxy-bypass attack in the way that received ID_R in verification must be same ID_R received by proxy after verification.

5.2 Specifications of implementation Environment

Host Machine

CPU	1.6 GHz (Dual CPU)	**RAM**	2.5GB

- Registration/Verification server is implemented on netbeans.
- My SQL server 5.5 for IDMS
- Android emulator 2.3

Virtual Machine

CPU	Shared	**RAM**	512

5.3 Analysis with respect to Authentication Latency

Authentication latency and response time are of great importance in mobile communication. High authentication latency and response time are not affordable in mobile business and E-Commerce. To overcome such issues, protocol is designed to remain simple and light weight by achieving adequate security level.

In the table below, there is a comparison of authentication latency of previously proposed solutions and GLCMP along with claimed security properties.

Sr.#	Protocols	Security Services			
		Authentication	Non-Repudiation	Digital Signature	Authentication Latency
1	Kerberos	Yes	No	No	0.19 Sec
2	PKINIT	Yes	No	No	1.21Sec
3	M-PKINIT	Yes	No	No	o.74 Sec
4	NSI	Yes	Yes	Yes	4.70 Sec
5	PKI	Yes	Yes	Yes	5.01 Sec
6	PKASSO	Yes	Yes	Yes	0.082 Sec
7	GLCMP	Yes	Yes	Yes	0.394Sec

Table 3: Security Services and Authentication Latencies of the protocols

The first three proposed solutions only deal with authentication of the mobile clients. But they do not provide a comprehensive security mechanism to deal with the issues of non-repudiation and integrity mismatch. Their authentication latencies are near to affordable. Fourth and fifth proposed solutions are comprehensive but their resulted authentication latencies are pretty high which are not affordable in mobile business and e-commerce environment.

To overcome limitations of PKI, NSI and M-PKINIT, PKASSO was proposed. No doubt it may provide less authentication latency (0.082 Sec) but this time is consumed after delegation of authentication operations to delegation server. To enter security infrastructure and delegate the responsibilities, asymmetric key operations are performed which take 5.19 second. Moreover, it is very complex to implement and integrate for mobile devices because for delegation mechanism three keys are shared between the User, delegation and referee servers.

	Kerberos	PKINIT	M-PKINIT	NSI	PKI	PKASSO	GLCMP
Auth. Latency (Sec.)	0.19	1.21	0.74	4.70	5.01	0.082	0.394

Table 4: Comparison of protocols regarding Authentication Latencies

Authentication latency of proposed GLCMP is 0.394 sec which is 91%, 92% less than NSI and PKI respectively. As for as PKASSO is concerned, If we include 5.19 sec delegation time, our result is 93% efficient but if we do not include it then our authentication latency is 79% greater as shown in Table 3, but here we are also providing secure registration.

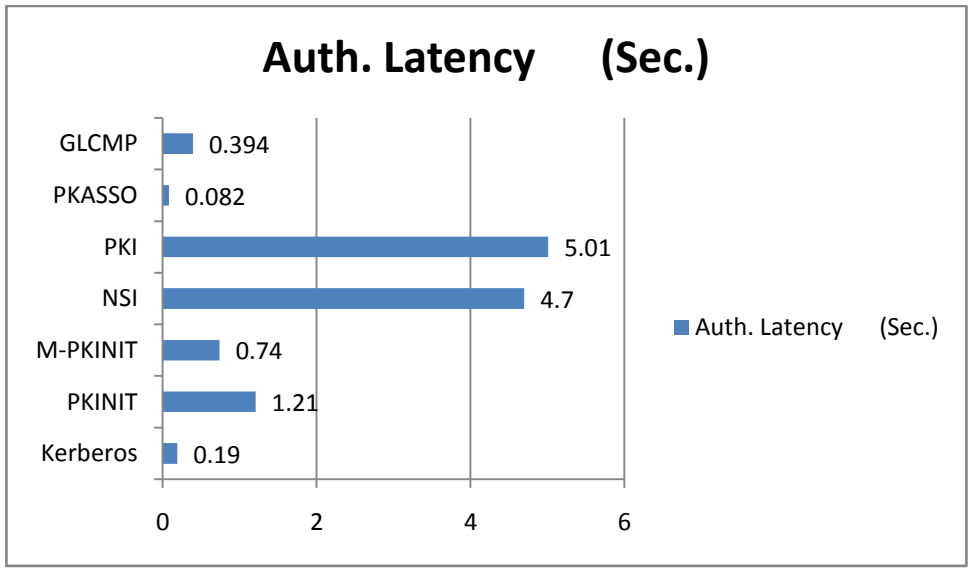

Figure 10: Authentication latency comparison

If we include time before delegation, then whole picture will become as shown below.

	Kerberos	PKINIT	M-PKINIT	NSI	PKI	PKASSO	GLCMP
Auth. Latency (Sec.)	0.19	1.21	0.74	4.70	5.01	5.272	0.394

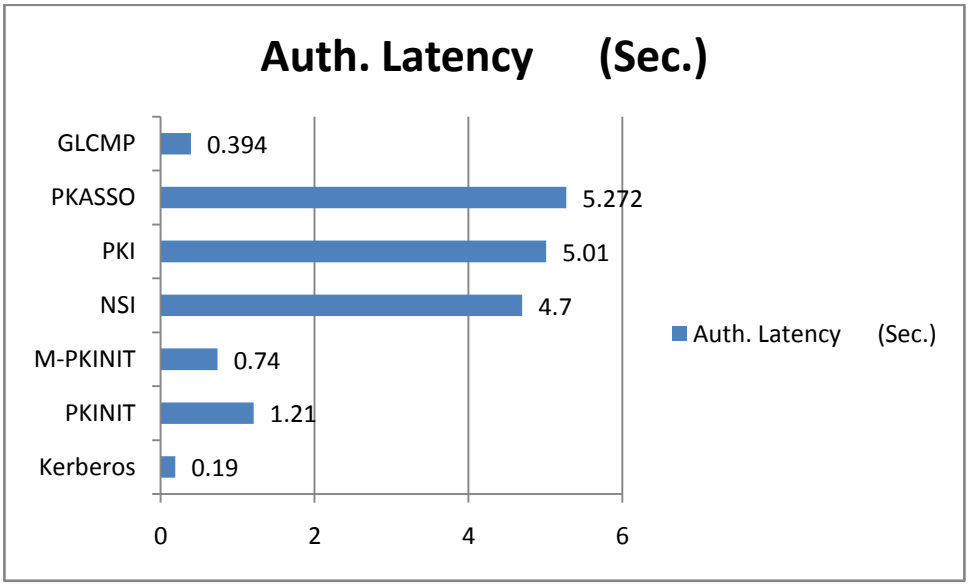

Figure 11: Authentication latency comparison including delegation time

	Kerberos	**M-PKINIT**	**NSI**	**PKASSO**	**GLCMP**
Operation time on Client (Sec)	0.024	0.518	4.726	0.066	2.78
Operation time on Server	0.036	0.333	0.51	3.253	0.9
Total	0.06	0.851	5.236	3.319	3.68

Table 5; Operation Time on different components of the system till verification

In our protocol, asymmetric key pair is generated on client side while in case of PKASSO, key pair is generated on delegation server.

Here is comparison of PKASSO and GLCMP regarding operation time on client and server.

(a) (b)

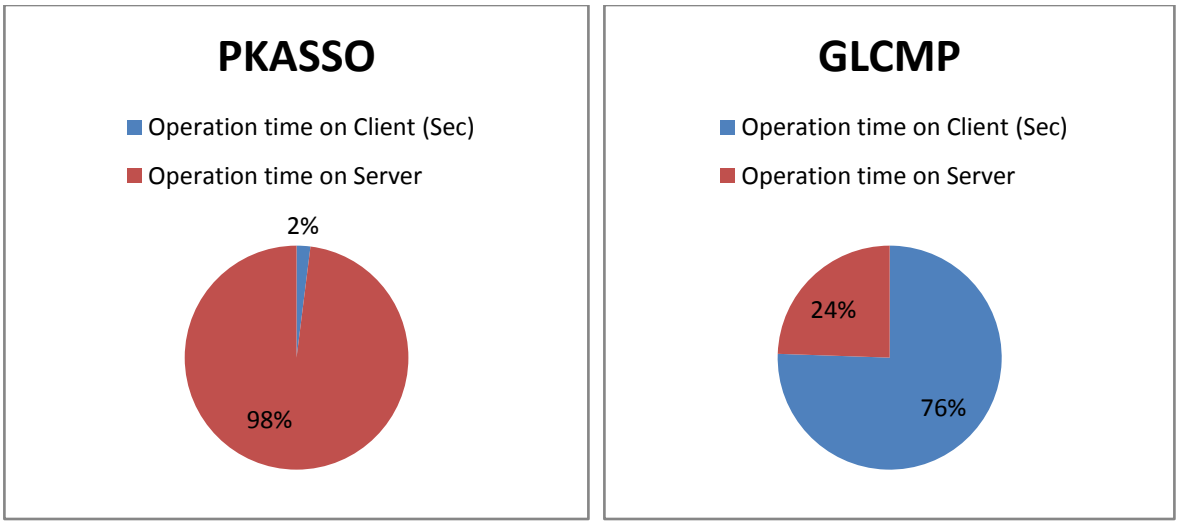

Figure 12: Comparison of the operation time on Client and Server: for (a) [2]

Same argument is here that delegation time is not included in the depicted operation time.

CHAPTER 6

Formal Verification of the Protocol

Formal verification of correctness of the protocols in generals and of security properties of security protocols especially is very essential before their deployment because there are chances that a correct protocol later found incorrect and a secure protocol later found insecure just like Needham-Schroeder Public Key Protocol.

6.1 Knowledge Modeling in Z Notation

Z based approach for formal verification and correctness of the protocols is very intuitive.

6.1.1 Phase 1: Suitable Data Types

In this phase suitable data types is assigned to agents and elements.

[AGENT]:= U | P | V | ⊥ |

This is the set of agents involved in the protocol

U : Mobile User	⊥ : No Agent
P : Proxy Server	V : Verification Server

ITEM: =DN_U| Enc | Dec| Hash | SP_U | NONCE| UID | KEY

Set of items used in verification.

MSG== Seq ITEM Messages are sequences of items.

Subset of ITEMS:

NONCE: ℙITEM Set of Nonce is a subset of set of items.

SP_U: ℙITEM User's secure Passwords

UID : ℙITEM Set of user IDs

ENC: ℙITEM

KEY: ℙITEM

Disjoint: ⟨DN, NONCE, SP_U, ENC, HASH, DEC, UID⟩

ITEM = DN **U** NON **U** SP_U **U** ADR **U** ENC **U** HASH **U** DEC **U** PKCS#10

Union of subsets is equal to superset ITEM.

6.1.2 Phase 2: Operations on the Message Components

In this phase Components of the messages are defined in form of operations. Each mobile user has its own public/private key pairs. Public key is used to bind distinguish name of an entity and private is to sign the certificate request. So we define sets of public and private keys as subset of KEY. Here in this table pair = pair ~ is bijective function to show one-to-one correspondence.

Possession of Keys Pairs
PU : ℙ KEY PR : ℙ KEY pair : KEY ⤖ KEY
disjoint < PU , PR > (PU ∪ PR) = KEY pair = pair ~ ∀ k : KEY • k ∈ PR ⇔ pair (k) ∈ PU

Declaration of Agent's Item Possession
PU : AGENT ↦ KEY PR : AGENT ↦ KEY NON : AGENT ↦ NON ID : AGENT ↦ UID
∀ G : AGENT • PU(G)=pair (PR(G)) In this declaration, the defined four functions map each agent uniquely.

6.1.3 Phase 3: Global State

This phase is used to define state space. In the analysis of security protocol, it is always assumed that instances are independent and one message is in transit at a time. InTransit represents the transition of the message and ⊥ represent no message in transit.

InTransit to : AGENT from : AGENT msg : MSG
Init InTransit ---------- to = ⊥

6.1.4 Phase 4: Dynamic Behavior as a set of Z Operations

This phase model the dynamic behavior of the protocol operations as a set of Z Notation. Here are four atomic operations corresponding to four steps modeled by standard notation.

Mobile User (U) -------> Proxy (P)

ΔInTransit
$to = \perp \wedge to' = P \wedge from' = U$
$msg' = (PU_U \mid DN_U) \frown \langle ENC[SP_U, (NO_U \mid H(PU_U \mid DN_U))] \rangle$

Proxy (P) -------- > Verifier (V)

Because Proxy server does not have any credential of mobile user to encrypt or decrypt so it will send the same message to verifier V for verification. In (X), represents the case if any unknown intruder intercepts the exchanged messages. Here X is any unknown.

Verifier will extract user ID_U from DN_U (Distinguish name of mobile user) and search in IDMS.

If $ID_U \in$ UID (set of registered users), Verifier retrieve the corresponding password, decrypt the message, calculate hash of (PU_U| DN_U) and compare it with received hash H (PU_U|DN_U).

ΔInTransit
$to = V \wedge from' = P$
$(\exists X: AGENT ; to' = X \wedge$
$msg = (PU_U \mid DN_U) \frown \langle ENC[SP_U, (NO_U \mid H(PU_U \mid DN_U))] \rangle \wedge$
$msg' = (PU_X \mid DN_X) \frown \langle ENC[SP_X, (NO_X \mid H(PU_X \mid DN_X))] \rangle$ …… (X)

Here (X) is the modeling of intruder.

$\langle DEC\ [SP_U,(NO_U|H\ (PU_U|DN_U\ )\ )]\rangle$

$= (NO_U|\ H\ (PU_U|DN_U))$ after Decryption.

$= H\ (PU_U|DN_U))$ after deconcatenation. …… (i)

Calculate hash of $(PU_U|DN_U)$.

$= H\ (PU_U|DN_U))$ ……(ii)

If (i) and (ii) are equal, integrity will be ensured and user will be verified successfully.

Verifier (V) ---------- > Proxy (P)

ΔInTransit
ΔInTransit
$to = V \wedge to' = P \wedge from' = V$
$msg' =$ (Accept or Reject) $\frown \langle ENC\ [SP_{U,}\ (NO_U|ID_R)]\rangle$

Proxy (P) --------- > User (U)

If verification is successful and accept message received to Proxy from Verifier, then proxy server create hash of "CR Info value" and sends to user along with encrypted received nonce and its ID_R.

ΔInTransit
$to = U \wedge from' = P$
$\exists H:Hash \bullet msg'=H$ (CR Info value) $\frown enc[SP_{U,}\ (NO_U|ID_R)]$

6.2 Formalizing the Attacks

1. U Sends to F (Fake Agent)
2. F Sends to P
3. P Forwards to V
4. V Reply to P
5. P Ack to F
6. F sends to U

U Sends to F (Fake Agent)

ΔInTransit
ΔInTransit

$to = \perp \wedge to' = F \wedge from' = U$
$msg' = (PU_U \mid DN_U) \frown \langle enc[SP_U, (NO_U \mid H(PU_U \mid DN_U))] \rangle$

F Sends to P

Fake user can intercept and modify the message. It can only change plaintext part $(PU_U|DN_U)$ and can do nothing with encrypted part except create its own.

If intruder is neither part of domain nor know any valid user ID, attack cannot be successful.

Lets intruder tamper distinguish name like $(PU_U|DN_U)$ with $(PU_U|DN_F)$ Then

ΔInTransit
ΔInTransit
$to = F \wedge to' = P \wedge from' = F$
$msg' = (PU_U \mid DN_F) \frown \langle enc[SP_U, (NO_U \mid H(PU_U \mid DN_U))] \rangle$

P proceed to V

Proxy server forwards the same message to verifier for verification.

$ID_F \notin UID$ (Set of registered User IDs)

ID_F is identity of intruder which takes from DF (Distinguish name of intruder).

So verification will be failed and Reject message will be sent to P and P will send this message to F. If intruder changer both public key and Distinguish name $(PU_U|DN_U)$ with fake $(PU_F|DN_F)$, and encrypted part as well, the result will be achieved because user id is not valid.

If Intruder is part of domain or knows any valid ID.

In this case IDF ∈ UID but corresponding password SP_F will remain unable to decrypt the message, so verification will be fail and attack will not be successful. ***So protocol is secure.***

If Intruder also take hash of $(PU_U|DN_F)$ concatenate its nonce NO_F and encrypt the message with its password.

ΔInTransit
ΔInTransit
$to = F \wedge to' = P \wedge from' = F$
$msg' = (PU_U \mid DN_F) \frown \langle enc[SP_F, (NO_F \mid H(PU_U \mid DN_F)] \rangle$

In this case, IDF ∈ UID, corresponding password SP_F will successfully decrypt the message, integrity will also be confirmed, and so verification will be successful.

$\langle Dec\ [SP_F, (NO_F|H\ (PU_U|DN_F))]\rangle$

$= (NO_F|H\ (PU_U|DN_F))$

$= H\ (PU_U|DN_F)$.which is equal to the calculated hash of $(PU_U|DN_F)$.

V Reply to P

ΔInTransit
ΔInTransit
$to = V \wedge to' = P \wedge from' = V$
$msg' = (Accept) \frown \langle enc\ [SP_F, (NO_F|ID_R)]\rangle$

P Reply to F

ΔInTransit
ΔInTransit
$to = P \wedge to' = F \wedge from' = P$
$msg' = Hash\ (CR\ Info\ value) \frown \langle enc\ [SP_F, (NO_F|ID_R)]\rangle$

F Sends to U

ΔInTransit
ΔInTransit
$to = F \wedge to' = U \wedge from' = F$
$msg' = H(CR\ Info\ value) \frown \langle enc\ [SP_F, (NO_F|ID_R)]\rangle$

Intruder will send the message to honest user U for signing. User will be unable to decrypt the message. So attack will not be successful and protocol is proved secure.

6.3. Bypassing the Verifier

If Intrude deploy its own Proxy which bypass the verification process

If Intruder intercepts the message and no matter modify the message or not and sends the hash of CR Info Value as the message is shown below.

ΔInTransit
ΔInTransit

$to = F \wedge to' = U \wedge from' = F$

$msg' = Hash(CR\ Info\ Value) \frown \langle ENC[SP_U, (NO_U | H(PU_U | DN_U))]\rangle$

Intruder cannot send any message encrypted with its key because decryption will be fail that result in failure of the attack. In this case honest agent will successfully decrypt the message, receive same nonce but did not receive identity of Verifier ID_R that receive in fetching process. Finally this attack will also be fail.

6.4. Verification of Registration Protocol

Assumptions:

- IDMS, Registration/Verification Server and Registration Terminal are members of the secure domain
- Only authenticated users of the domain can register
- SSL is implemented on the registration service

SSL client and server establish a secure SSL connection on the guarantees of following security properties.

a) Private connection is established because each message is sent in encrypted form.

b) Peer is authenticated by public/private keys

c) Integrity is confirmed by using message authentication code (MAC) computed by data and shared secret.

CHAPTER 7

Conclusion and Future Work

7.1 Conclusion

Due to mobility and easiness, mobile devices and digital gadgets have also been started being used in business and e-commerce. Research community increased its processing power and designed new advanced applications to attract business community to use it for e-commerce, business activities, sharing of valuable documents and many other sensitive activities. But still corporate community hesitates to adopt it properly because of some security concerns like authentication, authorization, confidentiality, tempering and non-repudiation. Moreover, there are also some problems like user is not enough technical to configure security properties, already developed libraries do not provide extended security features, high response time and authentication latency which is not affordable in mobile communication for business.

Considering all said security issues, researchers proposed different security mechanisms like WPKI, PKINIT, LPKI, NSI, SaPKI and PKASSO. But some mechanisms are very complex like PKASSO and LPKI to implement, some do not provide suitable response time and authentication latency like NSI and PKINIT and some mechanism are designed for GSM, CDMA networks but not for Wi-Fi like SaPKI.

The proposed protocol (GLCMP) is designed by keeping in view its practical implementation light weightiness and security concerns. It does not only provide reliable authentication of clients but also secure registration, identity verification and offloads intensive computational processing to Proxy Server that is involved in generation of certificate request in PKCS#10 standard formats. It is also implemented and concludes that its authentication latency 0.394 sec which is less than its nearest competitors NSI (4.7), PKI (5.01), and PKASSO (5.19 delegation time + 0.082 authentication times). Certificate management like transparent handling, distribution of certificates is other salient features of GLCMP. In our designed protocol, trust between mobile device and proxy server is established without exchanging any secret information on network. In

addition, we designed and developed our protocol using the concept of generic security objects which are easy to use by software engineers, easy to extend with new features and provide complete functions and features about a specific aspect.

Before the deployment, GLCMP has also been formally verified for the claimed security properties by Z notation modeling. For man in the middle attack, we discuss different possible scenarios like if the intruder is member of the secure domain or know any valid user ID from UID (Set of valid mobile user identities) and if intruder is not member of secure domain or does not know any valid user ID. We have also formally verified that intruder remain unable to bypass the proxy server and verification server. After verification we conclude that GLCMP ensures protection against replay, man in the middle, impersonation and non-repudiation attacks.

7.2 Future Directions

An interesting area of future research would be to integrate GLCMP with security applications like secure email client application for Android. The future work may also include finding out the ways to integrate SIM card with GLCMP to store certificates and to perform SIM based cryptographic functions.

Bibliography

[1] http://en.wikipedia.org/wiki/Public_key_infrastructure

[2] Ki-Woong Park, Sang Seok Lim and Kyu Ho Park, "Computationally Efficient PKI-Based Single Sign-On Protocol PKASSO for Mobile Devices", TRANSACTIONS ON COMPUTERS, VOL. 57, NO. 6, JUNE 2008.

[3] ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-10/pkcs-10v1_7.pdf

[4] SCHWIDERSKI-GROSCHE, S. and KNOSPE, H, "Secure m-commerce", *Electron. Commun. Eng. J.*, October 2002, 14, (5), pp.228–238.

[5] Fang LIU, Qi YANG, "Study and Analysis of the E-Commerce Security Based on WPKI", DOI, IITA Workshop, IEEE, 2008.

[6] MSign: see http://www.msign.org.

[7] Zigbee Specification v1.0. Zigbee Alliance Board of Directors, 2005.

[8] Tung, B., et al., Public Key Cryptography for Initial Authentication in Kerberos, 2001: http://www.ietf.org /internet-drafts/draft-ietf-cat-kerberos-pk-init-12.txt.

[9] L. Zhu and B. Tung, RFC 4556: Public Key Cryptography for Initial Authentication Kerberos (PKINIT). IETF Network Working Group, 2006.

[10] J. Lee, S.-H. Lim, J.-W. Yoo, K.-W. Park, H.-J. Choi, and K.H. Park, "A Ubiquitous Fashionable Computer with an i-Throw Device on a Location-Based Service Environment," Proc. 21st IEEE Int'l Conf. Advanced Information Networking and Applications Workshops, vol. 2, pp. 59-65, 2007.

[11] N. Modadugu, D.Bonech, and M.Kim, "Generating RSA keys on a Handheld Using an Untrusted Server", proceedings of the First International Conference in Cryptology in India, Lecture Notes in Computer Science, Vol. 11977, Springer-Verlag, Calcutta, India, 2000, pp.271-282.

[12] N. Asokan, G. Tsudik, and M.waidner, " Server Supported Signatures", Proceedings of the Fourth Symposium on Research in Computer Security (ESORICS), Lecture Notes in Computer Science, Vol. 1146, Springer-Verlag, Berlin, Germany, September 1996, pp.131-143.

[13] Mohsen Toorani, Ali Asghar Beheshti Shirazi, "A Lightweight Public Key Infrastructure for the Mobile Environ.", IEEE International Conference on Communication Systems (IEEEICCS'08), pp.162-166, Nov. 2008

[14] J.-H. Han, Y.-J. Kim, S.-I. Jun, K.-I. Chung and C.-H. Seo, "Implementation of ECC/ECDSA cryptography algorithms based on Java card," Proceedings of 22nd IEEE International Conference on Distributed Computing Systems, pp.272-276, 2002.

[15] D. Hankerson, A. Menezes, and Scott Vanstone, "Guide to Elliptic Curve Cryptography," Springer-Verlag, New York, 2004.

[16] H. Krawczyk, "HMQV: A high-performance secure Diffie-Hellman protocol (Extended Abstract)," Advances in Cryptology – CRYPTO'05, LNCS 3621, pp.546-566, Springer-Verlag, 2005.

[17] S. Tuecke, V. Welch, D. Engert, L. Pearlman, and M. Thompson, RFC 3820: Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile. IETF Network Working Group, 2004.

[18] C. A. Meadows. Formal verification of cryptographic protocols: A survey. In Advances in Cryptology — ASIACRYPT '94, volume 917 of Lecture Notes in Computer Science, pages 133–149. Springer-Verlag, 1995.

[19] J. Clark and J. Jacob. A survey of authentication protocol literature: Version 1.0, 1997. http://www.cs.york.ac.uk/ ~jac/papers/drareviewps.ps. Accessed May 2003.

[20] Borisov, N., I. Goldberg and D. Wagner, Intercepting mobile communications: the insecurity of 802.11, in: Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, 2001, pp. 180–189.

[21] C. Boyd and P. Kearney. Exploring fair exchange protocols using specification animation. In Proceedings of the Information Security Workshop (ISW 2000), volume
1975 of Lecture Notes in Computer Science, pages 209–223. Springer-Verlag, 2000.

[22] Mohamed Salah Bouassida and others, Automatic Verification of a Key Management Architecture for Hierarchical Group Protocols, LORIA, Campus
scientifique, B.P. 239, 54506 Vandoeuvre-les-Nancy Cedex – France, pages 1-15, 2006.

[23] Benjamin W. Long, Cross-Layer Verification of Type Flaw Attacks on Security Protocols, 30th ACSC-2007, Vol-62, Australia, pages 1-10, 2006.

[24] Borisov, N., I. Goldberg and D. Wagner, Intercepting mobile communications: the insecurity of 802.11, in: Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, 2001, pp. 180–189.

[25] M. Fahrmair, W. Sitou, and B. Spanfelner, "Security and Privacy Rights Management for Mobile and Ubiquitous Computing," Proc. Seventh Int'l Conf. Ubiquitous Computing, 2005.

[26] S. Tillich, and J. Großschädl, “A Survey of Public-Key Cryptography on J2ME-Enabled Mobile Devices,” Proceedings of ISCIS'04, LNCS 3280, pp.935-944, 2004.