www.ijiarec.com

ISSN:2348-2079

Volume-6 Issue-2

International Journal of Intellectual Advancements and Research in Engineering Computations

Enhanced Diffie-Hellman Algorithm and its Applications

[1]Mr. R. NavinKumar, M.C.A., M.Phil., Assistant Professor/MCA
[2] Mr. R. Saravanan, III MCA
Department of MCA, Nandha Engineering College (Autonomous) Erode – 52.
Email ID: navinsoccer@gmail.com, kuttysaravana65@gmail.com

Abstract: **This thesis presents the overview of the Diffie-Hellman Key Exchange algorithm and reviews several common cryptographic techniques in use on the Internet today that incorporate Diffie-Hellman. DH is a method for securely exchanging a shared secret between two parties, in real-time, over an untrusted network. A shared secret is important between two parties who may not have ever communicated previously, so that they can encrypt their communications. As such, it is used by several protocols like Secure Sockets Layer (SSL).**

The goal of this thesis is for achieving communication between two users by agrees upon a shared secret that an eavesdropper will not be able to determine. This shared secret is used by two users to independently generate keys for symmetric encryption algorithms that will be used to encrypt the data stream between them. The "key" aspect is that neither the shared secret nor the encryption key does not ever travel over the network.

In this thesis, along with these terms, study is also made such that different encryption concepts are applied based on the system's hardware/ software capability. In this aspect, some of the data is split into two segments and first one is encrypted by 3DES and second one is encrypted by AES encryption mechanism. This reduces the processing and communication overhead based on the system's capability.

With the rapid development of Internet, more web based services are into our daily life, and thus security protection of those services, especially data privacy protection, becomes more important. However to perform privacy protection causes huge overhead. Thus it is a critical issue to perform the most suitable protection to decline performance consumption while provide privacy protection.
Finally, the thesis shows that it not only fulfills the user-demand privacy but also maintains the system performance in network/ Internet environments. The application is designed using Microsoft Visual Studio .Net 2005 as front end. The coding language used is Visual C# .Net. MS-SQL Server 2000 is used as back end database.

Index Terms **– Diffie–Hellman Key, Computer Network, National Security Agency, Cable TV, NIST, KGC.**

I. INTRODUCTION

A. Networking

A computer network consists of a collection of computers, printers and other equipment that is connected together so that they can communicate with each other. Fig 1 gives an example of a network in a school comprising of a local area network or LAN connecting computers with each other, the internet, and various servers.

B. Peer-To-Peer Networks

Peer-to-peer networks are more commonly implemented where less then ten computers are involved and where strict security is not necessary. All computers have the same status, hence the term 'peer', and they communicate with each other on an equal footing. Files, such as word processing or spreadsheet documents, can be shared across the network and all the computers on the network can share devices, such as printers or scanners, which are connected to any one computer.

C. Diffie–Hellman Key Exchange

Diffie–Hellman key exchange (D–H) is a cryptographic that allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher.

Synonyms of Diffie–Hellman key exchange include [8]:

- Diffie–Hellman key agreement
- Diffie–Hellman key establishment
- Diffie–Hellman key negotiation
- Exponential key exchange
- Diffie–Hellman protocol

The Diffie-Hellman key agreement was invented in 1976 during a collaboration between Whitfield Diffie and Martin Hellman and was the first practical method for establishing a shared secret over an unprotected communications channel. Ralph Merkle's work on public key distribution was an influence. The method was followed shortly afterwards by RSA another implementation of public key cryptography using assymetric algorithms. The idea of public key cryptography was born as a result of two major challenges. The first of these was the problem of key distribution: if two people who have never met before are to communicate using digital systems as a medium, using conventional cryptography would mean that they must somehow agree on a common key that will be known to themselves and no one else. The other problem was the issue of signatures: this is a method of providing the recipient of a purely digital electronic message with a way of demonstrating to other people that it had come from a particular person, serving as a signature comparable to a written one on a letter.

II. LITERATURE SURVEY

Limits of agreement provide a straightforward and intuitive approach to agreement between different methods for measuring the same quantity. When pairs of observations using the two methods are independent, i.e., on different subjects, the calculations are very simple and straightforward. Some authors collect repeated data, either as repeated pairs of measurements on the same subject, whose true value of the measured quantity may be changing, or more than one measurement by one or both methods of an unchanging underlying quantity. In this paper we describe methods for analyzing such clustered observations, both when the underlying quantity is assumed to be changing and when it is not.

The rapid progress in wireless mobile communication technology and personal communication systems has prompted new security questions. Since open air is used as the communication channel, the content of the communication may be exposed to an eavesdropper, or system services can be used fraudulently. In order to have reliable proper security over the wireless communication channel, certain security measures, e.g., condentiality, authenticity, and untraceability need to be provided [1].

In a wireless mobile communication system, users and network servers need to authenticate one another and agree on a session key to be used for encryption purposes in their conversation [2]. Several authentication protocols have been proposed for wireless mobile communication systems. Among them, the protocol of Beller, Chang, and Yacobi [3] provides mutual authentication and key agreementbetween users and servers with low computational burden on the user side. This is important since the users usually communicate using a small, portable handset with limited power and processing capability. Furthermore, Aziz and Die [4] also proposed a similar authentication protocol, which does not require pre-computation. These two protocols along with several others[5] use public key cryptography techniques and online cortication procedures.

To meet today's needs for wireless digital communication, the developed protocols need to be highly secure, requiring low computational overhead and thus low power. Here, we introduce a new authentication and key agreement protocol for wireless mobile communication systems. The protocol is based on elliptic curve cryptography. In the following, we rst provide the necessary background for security requirements in a wireless communication system in x2, and then give a brief introduction to elliptic curve cryptographyin x3. We give the details of the proposed protocol in x4, a discussion of its implementation in x5, and its comparison to the Beller-Chang-Yacobi and Aziz-Die protocols in x6. The conclusions are found in x7.

Encryption is the transformation of data into a form that is as close to impossible as possible to read without the appropriate knowledge (a key; see below). Its purpose is to ensure privacy by keeping information hidden from anyone for whom it is not intended, even those who have access to the encrypted data. Decryption is the reverse of encryption; it is the transformation of encrypted data back into an intelligible form.

Encryption and decryption generally require the use of some secret information, referred to as a key. For some encryption mechanisms, the same key is used for both encryption and decryption; for other mechanisms, the keys used for encryption and decryption are different.

Today's cryptography is more than encryption and decryption. Authentication is as fundamentally a part of our lives as privacy. They are use authentication throughout our everyday lives -- when we sign our name to some document for instance -- and, as we move to a world where our decisions and agreements are communicated electronically, we need to have electronic techniques for providing authentication.

While modern cryptography is growing increasingly diverse, cryptography is fundamentally based on problems that are difficult to solve. A problem may be difficult because its solution requires some secret knowledge, such as decrypting an encrypted message or signing some digital

document. The problem may also be hard because it is intrinsically difficult to complete, such as finding a message that produces a given hash value physicaly. Some textbook treatments are provided by Stinson [6] and Stallings [Sta95], while Simmons provides an in-depth coverage of the technical aspects of cryptography [7]. A comprehensive review of modern cryptography can also be found in Applied Cryptography [9]; Ford [8] provides detailed coverage of issues such as cryptography standards and secure communication.

Cryptography standards are needed to create interoperability in the information security world. Essentially they are conditions and protocols set forth to allow uniformity within communication, transactions and virtually all computer activity. The continual evolution of information technology motivates the development of more standards, which in turn helps guide this evolution. The main motivation behind standards is to allow technology from different manufacturers to ``speak the same language", that is, to interact effectively. Perhaps this is best seen in the familiar standard VHS for video cassette recorders (VCRs).

A few years ago there were two competing standards in the VCR industry, VHS and BETA. A VHS tape could not be played in a BETA machine and viceversa; they were incompatible formats. Imagine the chaos if all VCR manufacturers had different formats. People could only rent movies that were available on the format compatible with their VCR. Standards are necessary to insure that products from different companies are compatible.

In cryptography, standardization serves an additional purpose; it can serve as a proving ground for cryptographic techniques because complex protocols are prone to design flaws. By establishing a well-examined standard, the industry can produce a more trustworthy product. Even a safe protocol is more trusted by customers after it becomes a standard, because of the ratification process involved. The government, private industry, and other organizations contribute to the vast collection of standards on cryptography. A few of these are ISO, ANSI, IEEE, NIST, and IETF .

There are many types of standards, some used within the banking industry, some internationally and others within the government. Standardization helps developers design new products. Instead of spending time developing a new standard, they can follow a pre-existing standard throughout the development process. With this process in place consumers have the chance to choose among competing products or services.

The Internet, comprised of millions of interconnected computers, allows nearly instantaneous communication and transfer of information, around the world. People use e-mail to correspond with one another. The World Wide Web is used for online business, data distribution, marketing, research, learning, and a myriad of other activities.

Cryptography makes secure web sites and electronic safe transmissions possible. For a web site to be secure all of the data transmitted between the computers where the data is kept and where it is received must be encrypted. This allows people to do online banking, online trading, and make online purchases with their credit cards, without worrying that any of their account information is being compromised. Cryptography is very important to the continued growth of the Internet and electronic commerce.

E-commerce is increasing at a very rapid rate. By the turn of the century, commercial transactions on the Internet are expected to total hundreds of billions of dollars a year. This level of activity could not be supported without cryptographic security. It has been said that one is safer using a credit card over the Internet than within a store or restaurant. It requires much more work to seize credit card numbers over computer networks than it does to simply walk by a table in a restaurant and lay hold of a credit card receipt. These levels of security, though not yet widely used, give the means to strengthen the foundation with which e-commerce can grow.

People use e-mail to conduct personal and business matters on a daily basis. E-mail has no physical form and may exist electronically in more than one place at a time. This poses a potential problem as it increases the opportunity for an eavesdropper to get a hold of the transmission. Encryption protects e-mail by rendering it very difficult to read by any unintended party. Digital signatures can also be used to authenticate the origin and the content of an e-mail message. Cryptography is also used to regulate access to satellite and cable TV. Cable TV is set up so people can watch only the channels they pay for. Since there is a direct line from the cable company to each individual subscriber's home, the Cable Company will only send those channels that are paid for. Many companies offer pay-per-view channels to their subscribers. Pay-per-view cable allows cable subscribers to ``rent" a movie directly through the cable box. What the cable box does is decode the incoming movie, but not until the movie has been ``rented." If a person wants to watch a pay-per-view movie, he/she calls the cable company and requests it. In return, the Cable Company sends out a signal to the subscriber's cable box, which unscrambles (decrypts) the requested movie. Satellite TV works slightly differently since the satellite TV companies do not have a direct connection to each individual subscriber's home. This means that anyone with a satellite dish can pick up the signals.

To alleviate the problem of people getting free TV, they use cryptography. The trick is to allow only those who have paid for their service to unscramble the transmission; this is done withreceivers (``unscramblers"). Each subscriber is

given a receiver; the satellite transmits signals that can only be unscrambled by such a receiver (ideally). Pay-per-view works in essentially the same way as it does for regular cable TV. As seen, cryptography is widely used. Not only is it used over the Internet, but also it is used in phones, televisions, and a variety of other common household items. Without cryptography, hackers could get into our e-mail, listen in on our phone conversations, tap into our cable companies and acquire free cable service, or break into our bank/brokerage accounts.

These are lecture notes from two courses on cryptography that I teach at Santa Clara University. The course has few technical prerequisites. At one moment I use implicit dierentiation from quarter calculus. Beyond that, I develop everything else from high-school level mathematics. I have given history short-shrift in my attempt to get to modern cryptography as quickly as possible. Yin as well as the publications listed in the bibliography. I am very grateful to each person listed above. Any mistakes in this document are mine. Please notify me of any that you and at the above e-mail address.

If Alice wants to send a message to Bob and she does not want the eavesdropping Carol to understand, then Alice can encrypt it and send it to Bob. Bob receives the message and decrypts it. We study these two actions in the cryptography course. If Carol intercepts the message, then she can try to break the code and read the message. We study this action in the cryptanalysis course. In this course we will cover vocabulary, history, number theory, simple cryptosystems, simple cryptanalysis, running time anaylsis and modern cryptosystems.

Cryptography is used to hide information. It is not only use by spies but for phone, fax and e-mail communication, bank transactions, bank account security, pin numbers and passwords. It is also used for electronic signatures which are used to prove who sent a message. NIST issues standards for cryptographic algorithms that U.S. government agencies are required to use. A large percentage of the private sector often adopts them as well.

In NIST declared DES the official U.S. encryption standard and published it as FIPS 46; DES soon became a de facto standard throughout the United States. NIST is currently taking nominations for the Advanced Encryption Standard (AES), which is to replace DES. There is no definite deadline for the completion of the AES. Several years ago, NIST was asked to choose a set of cryptographic standards for the U.S., this has become known as the Capstone project. After a few years of rather secretive deliberations, NIST, in cooperation with the NSA, issued proposals for various standards in cryptography. The combination of these proposals, including digital signatures and data encryption, formed the Capstone project. NIST has been criticized for allowing the NSA too much power in setting cryptographic standards, since the interests of the NSA sometimes conflict with that of the Commerce Department and NIST. Yet, the NSA has much more experience with cryptography, and many more qualified cryptographers and cryptanalysts than does NIST so it is perhaps unrealistic to expect NIST to forego such readily available assistance.

NSA is the National Security Agency, a highly secretive agency of the U.S. government created by Harry S. Truman. The NSA's very existence was kept secret for many years. For a history of the NSA, see Bamford [10]. The NSA has a mandate to listen to and decode all foreign communications of interest to the security of the United States. It has also used its power in various ways to slow the spread of publicly available cryptography in order to prevent national enemies from employing encryption methods that are presumably too strong for the NSA to break. As the premier cryptographic government agency, the NSA has huge financial and computer resources and employs a host of cryptographers. Developments in cryptography achieved at the NSA are not made public; this secrecy has led to many rumors about the NSA's ability to break popular cryptosystems like DES as well as rumors that the NSA has secretly placed weaknesses, called ``trapdoors," in government-endorsed cryptosystems. These rumors have never been proved or disproved. Also the criteria used by the NSA in selecting cryptography standards have never been made public. Recent advances in the computer and telecommunications industries have placed NSA actions under unprecedented scrutiny, and the agency has become the target of heavy criticism for hindering U.S. industries that wish to use or sell strong cryptographic tools. The two main reasons for this increased criticism are the collapse of the Soviet Union and the development and spread of commercially available public-key cryptographic tools. Under pressure, the NSA may be forced to change its policies.

The NSA's charter limits its activities to foreign intelligence. However, the NSA is concerned with the development of commercial cryptography, since the availability of strong encryption tools through commercial channels could impede the NSA's mission of decoding international communications. In other words, the NSA is worried that strong commercial cryptography may fall into the wrong hands.

III. METHODOLOGY

A. EXISTING METHODOLOGY

In the existing system, the goal is for Alice and Bob to be able to agree upon a shared secret that an eavesdropper will not be able to determine. This shared secret is used by Alice and Bob to independently generate keys for symmetric encryption algorithms that will be used to encrypt the data stream between them. The “key” aspect is that neither the

shared secret nor the encryption key does not ever travel over the network.

In addition, two parties can exchange encrypted data by getting the other's public key and certificate from the Public Key Infrastructure (PKI). Of key importance in this process is that the public key is not altered in any way in transit, so an encrypted hash of the certificate is made, where the encryption key is derived from a DH exchange.

The existing system has following disadvantages.

- Only single user-to-user communication process is considered.
- Since individual encryption and decryption mechanism for one user to other user is required, it increases the processing and storage overhead.
- Not suitable for session based communication with group of users involved.

B. PROPOSED METHODOLOGY

In the proposed system, the existing approaches are carried out. In addition, the proposed system includes a group key transfer protocol using secret sharing scheme. In this system, each user needs to register at Key Generation Center (KGC) to subscribe the group key transfer service and to establish a secret with KGC. Thus, a secure channel is needed initially to share this secret with each user. Later, KGC can transport the group key and interact with all group members in a broadcast channel. The confidentiality of group key distribution is information theoretically secure; that is, the security of this transfer of group key to each group member does not depend on any computational assumption. The authentication of the group key is achieved by broadcasting a single authentication message to all group members. The proposed system has following advantages,

- Every user needs to register at a trusted KGC initially and preshare a secret with KGC.
- KGC broadcasts group key information to all group members at once.
- Multiple user-to-user communication process is considered.
- The confidentiality of our group key distribution is information theoretically secure.
- We provide group key authentication.
- Security analysis for possible attacks is included.
- Suitable for session based communication with group of users involved.

IV. ALGORITHM

a. Protocol in action

Diffie-Hellman is not an encryption mechanism as we normally think of them in that we do not typically use it to encrypt data. Instead, it is a protocol to securely exchangethe keys that encrypt data. Diffie-Hellman accomplishes this secure exchange by creating a "shared secret" (sometimes called a "Key Encryption Key" or KEK) between two devices. The shared secret then encrypts the symmetric key for secure transmittal. The symmetric key is sometimes called a Traffic Encryption Key (TEK) or Data Encryption Key (DEK). Therefore, the KEK provides for secure delivery of the TEK, while the TEK provides for secure delivery of the data itself. [14] The protocol has two system parameters *p* and *g*. They are both public and may be used by all the users in a system. Parameter *p* is a prime number and parameter *g* (usually called a generator) is an integer less than *p*, with the following property: for every number *n* between 1 and *p*-1 inclusive, there is a power *k* of *g* such that $n = g^k \bmod p$. [15]

To make a more simple description we shall imagine two people – Alice and Bob[16] who want to securely exchange data.Suppose Alice and Bob want to agree on a shared secret key using the Diffie-Hellman key agreement protocol. They proceed as follows: Alice and Bob agree on a finite cyclic group *G* and a generating element *g* in *G*. (This is usually done long before the rest of the protocol; *g* is assumed to be known by all attackers). First, Alice generates a random private value *a* and Bob generates a random private value *b*. Both *a* and *b* are drawn from the set of integers . Then they derive their public values using parameters *p* and *g* and their private values. Alice's public value is $g^a \bmod p$ and Bob's public value is $g^b \bmod p$. They then exchange their public values. Finally, Alice computes $g^{ab} = (g^b)^a \bmod p$, and Bob computes $g^{ba} = (g^a)^b \bmod p$. Since $g^{ab} = g^{ba} = k$, Alice and Bob now have a shared secret key *k*. [15] The important point is that the two values
generated are identical. They are the "Shared Secret" that can encrypt information between systems [illustration 2].

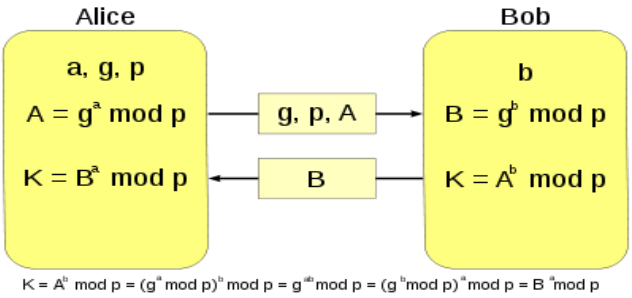

Illustration 2: Diffie-Hellman key exchange

Fig.1

Here is an example of the protocol, with non-secret values in green, and secret values in boldface red:

1. Alice and Bob agree to use a prime number p=23

and base g=5.

2. Alice chooses a secret integer a=6, then sends Bob $A = g^a \bmod p$
   - $A = 5^6 \bmod 23 = 8$.
3. Bob chooses a secret integer b=15, then sends Alice $B = g^b \bmod p$
   - $B = 5^{15} \bmod 23 = 19$.
4. Alice computes $s = B^a \bmod p$
   - $19^6 \bmod 23 = 2$.
5. Bob computes $s = A^b \bmod p$
   - $8^{15} \bmod 23 = 2$. [8]

At this point, the Diffie-Hellman operation could be considered complete. The shared secret is a cryptographic key that could encrypt traffic. That is very rare however because the shared secret is an asymmetric key. As with all asymmetric key systems, it is inherently slow. If the two sides are passing very little traffic, the shared secret may encrypt actual data. Any attempt at bulk traffic encryption requires a symmetric key system such as DES, Triple DES, or Advanced Encryption Standard (AES), etc. In most real applications of the Diffie-Hellman protocol (SSL, TLS, SSH, and IPSec in particular), the shared secret encrypts a symmetric key for one of the symmetric algorithms, transmits it securely, and the distant end decrypts it with the shared secret. Because the symmetric key is a relatively short value (256 bits for example) as compared to bulk data, the shared secret can encrypt and decrypt it very quickly.

Which side of the communication actually generates and transmits the symmetric key varies. However, it is most common for the initiator of the communication to be the one that transmits the key. [14]. Once secure exchange of the symmetric key is complete, data encryption and secure communication can occur. Changing the symmetric key for increased security is simple at this point. The longer a symmetric key is in use, the easier it is to perform a successful cryptanalytic attack against it. Therefore, changing keys frequently is important. Both sides of the communication still have the shared secret and it can be used to encrypt future keys at any time and any frequency desired. In some IPSec implementations for example, it is not uncommon for a new symmetric Data Encryption Key to be generated and shared every 60 seconds. [14] . The protocol depends on the discrete logarithm problem for its security. It assumes that it is computationally infeasible to calculate the shared secret key $k = g^{ab} \bmod p$ given the two public values $g^a \bmod p$ and $g^b \bmod p$ when the prime p is sufficiently large. It is stated that breaking the Diffie-Hellman protocol is equivalent to computing discrete logarithms under certain assumptions.

b. EXPERIMENTAL SETUP

1. Symmetric encryption/decryption
2. Asymmetric encryption/decryption
3. Diffie Hellman – SSL
4. Session based group key protocol

1. Symmetric Encryption/Decryption

a. Message Selection
b. Encryption
c. Decryption

a.) Message Selection

In this module, the message content is keyed in text box control or selected from file. The message is saved into the 'RawMessages' table.

b.) Encryption

In this module, three types of encryption word are carried out.

Speed: The requirement of this level presents that no sensitive information in the data. Users want to use the weak encryption composition to obtain more performance for using cloud services.

Hybrid: The requirement of this level presents that data include some sensitive information. The data requires weak encryption for partial data (such as address, mail id of corporates') and strong encryption for remaining data (such as account balances and other secure information).

Security: In this privacy level, the data contains most important information. In order to protect the data security, more privileged users view most of the data and less privileged users view limited data.

c.) Decryption

In this module, three types (speed, hybrid and security) of decryption work is carried out.

2. Asymmetric Encryption/Decryption

a. Setup TTP (Param)
b. Setup TTP (Param)
c. PSig(m, SKUi, APK):
d. PVer (m, □p, PKUi, APK) (outputs {0,1})
e. Sig (m, SKU, □p, APK)
f. Ver (m, □, PKUi, APK)

a.) Setup TTP (Param):

In this module, the Setup TTP Algorithm is worked out. On input the parameter Param, the arbitrator executes this algorithm to obtain a public–private key pair .(APK, ASK).

b.) Setup User (Param):

In this module, the Setup User Algorithm is worked out. On input the parameter Param ,the signer Ui executes this algorithm to obtain a public–private key pair (PKUi, SKUi).

c.) PSig(m, SKUi, APK):

In this module, the Setup Partial Signature Algorithm is worked out. On input a message m, Ui's private key SKUi and the arbitrator's public key APK, the partial signing algorithm PSig outputs a partial signature □p.

Here, 'm' should include the description of the items to be exchanged between Ui and Uj. This will help the arbitrator process the resolution request. Notice that the partial signature □p is not Ui's agreement of sending its item to Uj, even though in some concrete constructions □p is a classical digital signature.

Instead, only a full signature (will be defined shortly) can be viewed as Ui's agreement of fulfilling its obligation (as described in m).

The partial signature □p only shows Ui's willingness to send its item to if fulfills its obligation.

d.) PVer (m, □p, PKUi, APK) (outputs {0,1}):

In this module, the Partial Verification Algorithm is worked out. On input a pair (m, □p) and two public keys (PKUi, APK), this algorithm outputs '1' (if □p is valid) or "0" (otherwise). (m, □p) is said to be a valid message–partial-signature pair under (PKU, APK) if PVer (m, □p, PKUi, APK) = 1.

e.) Sig (m, SKU, □p, APK):

In this module, the Signature Algorithm is worked out. On input a message m, Ui's private key SKUi, (optionally) Ui's valid partial signature □p on m and (optionally) the arbitrator's public key APK, this algorithm outputs a full signature □.

To distinguish from the partial signature □p generated by PSig, □ is called the full signature.

f.) Ver (m, □, PKUi, APK):

In this module, the Verification Algorithm is worked out. On input a pair (m, □□□and two public keys (PKUi, APK), this algorithm outputs "1" (if □□is valid) or "0" (otherwise). (m, □□□is said to be a valid message–signature pair under (PKUi, APK) if Ver (m, □, PKUi, APK) = 1.

3. DIFFIE HELLMAN – SSL

a. Collect the client's original request details in client proxy stand alone application.
b. Convert the client request to cryptographic and steganographic data in client proxy application.
c. Send data from client proxy to server proxy application.
d. Convert the request data in server proxy to original client request and pass the request to web server.
e. Process the request and prepare the response and send response content to server proxy.
f. Convert the original response data in server proxy to cryptographic and steganographic data and pass the response to client proxy.
g. Convert the data to original client response in client proxy application and display the response content to client.

a.) Collect the client's original request details in client proxy stand alone application

The client proxy application collects the HTTP request data. This is being done by getting html web page URL along with query string.

b.) Convert the client request to cryptographic and steganographic data in client proxy application

The web site's domain name, page url and query string data information is collected and cryptographed using TripleDES (Data Encryption Standard). The key information used is common to both client and server proxy application.

c.) Send data from client proxy to server proxy application

The cryptographed data is sent to server proxy application which may be a web service which is running in destination.

d.) Convert the request data in server proxy to original client request and pass the request to web server.

The web site's domain name, page url and query string data information is collected from cryptographed data and pass the request to server using this module.

e.) Process the request and prepare the response and send response content to server proxy

The web server processes the request and prepares the HTML content and sends response to server proxy application using this module.

f.) Convert the original response data in server proxy to cryptographic and steganographic data and pass the response to client proxy.

The response information is collected and cryptographed using TripleDES (Data Encryption Standard). The key information used is common to both client and server proxy

application. The response is sent to client proxy using this module.

g.) Convert the data to original client response in client proxy application and display the response content to client.

The response information is collected from cryptographed response and content is displayed using this module.

4. SESSION BASED GROUP KEY PROTOCOL

a. Initialization of kgc (key generation center)
b. User registration
c. Group key generation and distribution.

a. Initialization Of Kgc (Key Generation Center)

The KGC randomly chooses two safe primes p and q (i.e., primes such that p' = (p-1)/2 and q' = (q-1)/2 are also primes) and compute n =pq. n is made publicly known.

b. User Registration

Each user is required to register at KGC for subscribing the key distribution service. The KGC keeps tracking all registered users and removing any unsubscribed users. During registration, KGC shares a secret, (x_i, y_i), with each user U_i, where x_i, y_i $\in \mathbb{Z}_n^*$

c. Group Key Generation And Distribution

Upon receiving a group key generation request from any user, KGC needs to randomly selects a group key and access all shared secrets with group members. KGC needs to distribute this group key to all group members in a secure and authenticated manner. All communication between KGC and group members are in a broadcast channel. For example, we assume that a group consists of t members, {U1, U2, . . . , Ut), and shared secrets are (x_i, y_i), for i = 1, . . . , t. The key generation and distribution process contains five steps.

Step 1.The initiator sends a key generation request to KGC with a list of group members as {U1, U2, . . . , Utg}.

Step 2. KGC broadcasts the list of all participating members, {U1, U2, . . . ,Ut}, as a response.

Step 3. Each participating group member needs to send a random challenge, Ri belongs to $\mathbb{Z}_n^*$, to KGC.

Step 4. KGC randomly selects a group key, k, and generates an interpolated polynomial f(x) with degree t to pass through (t + 1) Ri), for i =1, . . . , t. KGC also computes t points, (0, k) and (xi, yi additional points, Pi, for i = 1, . . . , t, on f(x) and Auth = h(k,U1, . . . , Ut, R2, . . .,Rt, P1, . . . , Pt), where h is a one-way hash function. All computations on f(x) are over $\mathbb{Z}_n^*$. KGC broadcasts (Auth, Pi), for i = 1, . . . , t, to all group members. All computations are performed in $\mathbb{Z}_n^*$

Step 5. For each group member, Ui, knowing the shared Ri), and t additional public points, Pi, for i = 1, . . . , secret, (xi, yi t, on f(x), is able to compute the polynomial f(x) and recover the group key k = f(0). Then, Ui computes h(k, U1, . . . , Ut;R1, . . ., Rt, P1, . . . , Pt) and checks whether this hash value is identical to Auth. If these two values are identical, Ui authenticates the group key is sent from KGC.

V. CONCLUSION

Through this thesis, the problem of secure communication is eliminated. In addition, the application required less working experience in systems to run the software. The application is tested well so that the end users use this software for their whole operations. The proposed system includes a group key transfer protocol using secret sharing scheme. In this system, each user needs to register at Key Generation Center (KGC) to subscribe the group key transfer service and to establish a secret with KGC.

Thus, a secure channel is needed initially to share this secret with each user. Later, KGC can transport the group key and interact with all group members in a broadcast channel. It is believed that almost all the system objectives that have been planned at the commencement of the software development have been net with and the implementation process of the project is completed. A trial run of the system has been made and is giving good results the procedures for processing is simple and regular order. The process of preparing plans been missed out which might be considered for further modification of the application. The project effectively stores and retrieves the records from the cloud space database server. The records are encrypted and decrypted whenever necessary so that they are secure.

SCOPE FOR FUTURE DEVELOPMENT

The following enhancements are should be in future.

- ✓ The application if developed as web services, then many applications can make use of the records.
- ✓ The data integrity in cloud environment is not considered. The error situation can be recovered if there is any mismatch.

- ✓ The web site and database can be hosted in real cloud place during the implementation.

REFERENCES

[1] M. J. Beller, L.-F. Chang, and J. Yacobi.Privacy and authentication on a portable communications systems. IEEE JournalonSelectedAreasin Communications, 11(6):821{829, August 1993.

[2] W. Die, P.C. VanOorschot, and M.J. Wiener. Authentication and authenticated key exchanges. Designs, CodesandCryptography, 2:107{125,1992.

[3] A.Azizand W.Die. A secure communications protocol to prevent unauthorized access :Privacy and authentication for wireless local area networks.IEEE Personal Communications, pages 25{31,First Quarter 1994.

[4] M. J. Beller and J. Yacobi. Fully-edged two-way public key authentication and key agreementfor low-cost terminals.ElectronicsLetters, 29(11):999{1001, May 27th 199.

[5] J.Dj.Goli c. CrytanalysisofallegedA5stream cipher. In W.Fumy, editor, Advancesin Cryptology |EUROCRYPT97, Lecture Notesin Computer Science, No.1233, pages239 {255. New York, NY: Springer-Verlag, 1997.

[6] D.R. Stinson, Cryptography -- Theory and Practice, CRC Press, Boca Raton, 1995.

[7] G.J. Simmons, editor, Contemporary Cryptology -- The Science of Information Integrity, IEEE Press,1992.

[8] W. Ford, Computer Communications Security Principles, Standard Protocols and Techniques, PrenticeHall, New Jersey (1994).

[9] B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd Edition, Wiley, 1995.

[10] J. Bamford, The Puzzle Palace, Houghton Mifflin, Boston, 1982.

[11] Carts, D.A. (2001) A Review of the Diffie-Hellman Algorithm and its Use in Secure Internet Protocols. [Online]

[12] Available at http:// www.sans. org/reading_ room/ whitepapers/vpns/review-diffie-hellman-algorithm-secure internetprotocols_751 [Accessed 25 October 2012].

[13] Diffie Hellman Encryption Algorithm, (2009). Diffie Hellman Encryption Overview. [http://www. diffiehellman.com/overview.html]

[14] Keith Palmgren, CISSP (2006, August).Diffie-Hellman Key Exchange A Non-Mathematicians Explanation [http://www.netip.com/articles/keith/diffie- elman.htm]

[15] RSA Laboratories, (2009). What is Diffie-Hellman? [http://www.rsa.com/rsalabs/node.asp?id=2248#]

[16] Wikipedia, (2009, November 15). Diffie-Hellman key exchange[http://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange]

[17] Wikipedia, (2009, November 12). Alice and Bob [http://en. wikipedia. org/wiki/Alice_and_Bob]

[18] Raymond, J.F. and Stiglic, A. (2000) Security Issues in the Diffie-Hellman Key Agreement Protocol. [Online]

[19] Oracle (2008-12) "Diffie-Hellman", Oracle Think Quest. [Online] Available at http://library.thinkquest. org/C0126342/dh.htm [Accessed 12 March 2012].

[20] Carts, D.A. (2001) A Review of the Diffie-Hellman Algorithm and its Use in Secure Internet Protocols. [Online]

[21] riedl, M., Provos, N., and Simpson, W. (2006) Diffie-Hellman Group Exchange for the Secure Shell (SSH) Transport Layer Protocol. [Online] Available at http://www.ietf.org/rfc/rfc4419.txt [Accessed 14 March 2012].

[22] Florian Tegeler, (2008) Security Analysis, Prototype Implementation and Performance Evaluation of a New Ip Sec Session Resumption Method. [Online] Available at: http:// www. net.informatik. unigoettingen.de /publications /1512/ FT egeler_MSc Thesis.pdf [Accessed 29 October 2012].

[23] Williams, S. (2011) Analysis of the SSH Key Exchange Protocol. [Online] Available at http://eprint.iacr.org/2011/276.pdf [Accessed 15 March 2012].

[24] Kahate, A. (2008) Cryptography and Network Security. McGraw-Hill. Available at http://www.sans.org/reading_room/whitepapers/vpns/review-diffie-hellman-algorithm -secure-internetprotocols_751 [Accessed 25 October 2012]