A NOVEL APPROACH TO INFORMATION SECURITY USING SAFE EXCHANGE OF ENCRYPTED DATA (SEED)

Kavitha.V[1], Mohammed Shaffi.Y[2], Arun Kumar.R[3], Mani Muthiah.M[4]

[1]Associate Professor, [2,3,4] Student, Department of Computer Science and Engineering, Sri Sairam Engineering College, Tamilnadu, India ***mani.muthiah92@gmail.com, kavitha.cse@sairam.edu.in***

Abstract

In this modern era, with the vast improvement in the field of internet, security is a major issue at hand. A lot of crimes, or to say, hacking is prevalent. This system ***"Safe Exchange of Encrypted Data (SEED)"*** *handles sharing secret data between the sender and receiver in a cryptic manner by providing a new approach to symmetric encryption with ensured confidentiality, authenticity, integrity and availability of a secure communication, and protection against Man-in-the-Middle attacks even without a Public Key Infrastructure (PKI) or endpoint certificates, in the unprotected network space. This system makes use of an efficient concept called 'ephemeral shared session key', which being a combination of public and private keys can only be generated at both ends and negates the need of having to transmit a symmetric key between the users. The text data is encrypted using a new symmetric key algorithm known as "Xenacrypt" which is more secure than any other existing symmetric key algorithms. This system provides integrity through an efficient algorithm which we have implemented to indicate data thefts by any malicious attacks or threats. Application of this crypto-system will have a huge impact in the future of transmitting secure data especially in the field of business transaction and military operations.*

Keywords:-*encryption;signed diffie hellman;signature;VOIP Integrity,verification,decryption,authentication.*

----------***----------

1. INTRODUCTION

In day to day life, we see colossal number of people using a wide range of devices like personal computers, laptops, tablets, mobile devices, etc. These devices are broadly used for exchange of data with the help of some message transfer applications. Existing system provides variety of interfaces and add-ons for these data transfer but the degree to which security is provided remains a big question. PKI (Public Key Infrastructure) is a security architecture that has been introduced to provide an increased level of confidence for exchanging information over an increasingly insecure Internet. But our system provides the same level of confidence without using PKI which is complex to set up.

Highly secure algorithms are available for ensuring safety but still there are drawbacks like vulnerable attacks, incompatibility, low data rate, high memory usage, time latency and packet loss. To ensure secure data exchange we propose a system which uses an innovative transmission system with safe establishment of a secure connection and a novel encryption technique for text data. Our system is primarily developed for providing three parameters without compromising on the time for processing and memory usage. Firstly, we provide confidentiality which is a service used to hide the content of information from all but those authorized to have it. Next, we intend to provide data integrity which is a service that addresses the unauthorized alteration of data. Finally, we are providing authentication. It is a service which ensures that the transmitted data reaches the intended recipient. Providing all these three together for transferring of data is a hard task. But for the data transmission to be highly secure we need to satisfy all these parameters. Our system provides all these three services to a greater extent and ensures that the authorized user gets the original data without any loss or modification.

2. REVIEW OF EXISTING SYSTEM

The existing data exchange system uses base data annotated with "security metadata", which ensures confidentiality (by containing information about access control requirements and encryption algorithm details) and integrity (by containing evidence of legitimacy in the form of signatures) only in local networks. Since data is exchanged beyond domains of influence of data authors, we can't depend on secure systems to enforce confidentiality and integrity, but must rely on techniques of cryptography [1]. Yet, many compelling cryptographic primitives available are not very much secured and also adapting them to complicated data management is a major challenge. Another concern is that the system makes use of trusted third parties to generate keys, transmitted to both the users, in which there is a high chance of the key getting lost by any congestion in the network or third party attacks. Also, usage of public key encryption, poses complexity and delay in transmission even though it offers a high level of security to the data exchanged. PGP is one such system, involved in the field of providing

secure transmission of data in a complex network. Yet there is no way for the present system to provide integrity and availability to the data transmitted over a hostile network as such.

In cryptography, the one-time pad (OTP) is a type of encryption which has been proven to be impossible to crack if used correctly. Each bit or character from the plaintext is encrypted by a modular addition with a bit or character from a secret random key of the same length as the plaintext, resulting in a cipher text. If the key is truly random, as large as or greater than the plaintext, never reused in whole or part, and kept secret, the cipher text will be impossible to decrypt or break without knowing the key. But the same one-time pad data can not be shared securely among more than two people; for example, in cases where diverse messages will be sent between some recipients of the key data, which should not be readable to other recipients, using the same one-time pad amongst all of them subverts the security of the cipher. By contrast, with an asymmetric cipher you can grant the same public key to countless number of people, and they will all be able to use that same public key to encrypt messages for you without any apprehension the other people who have the public key will be able to read it — as long as the cipher is not wrecked [8] and the state of the art of computing technology does not advance enough to reasonably brute force decrypt the messages. This is because when something is encrypted with the public key, only the allied private key can be used to decrypt it.

Reusing a key potentially breaks the security of the one-time pad cipher because it suffers known-plaintext vulnerability. A one-time pad encryption key must be as long as the message it is used to encrypt and decrypt. Thus, if you want to encrypt or decrypt a three gigabyte file, you need three gigabytes of one-time pad key data. Because the one-time pad cipher is a symmetric cipher, both parties to an encrypted communication must have the exact same key data. For certain use cases for encryption, this makes a one-time pad completely useless, because to securely exchange the key data so both parties have it, one must have a secure means of sharing data that would work just as well for sharing the eventual messages themselves. Only in cases in which you do not know what messages you will need to send, and where you will not be able to use whatever secure means was used to exchange the key data (such as physically handing it to the person) at the later time, is the one-time pad cipher useful.

3. OVERVIEW OF SYSTEM DESIGN

3.1 Fundamental System

The system proposed generates a key pair consisting of a public key and its corresponding private key using pseudo random number generator (PRNG) algorithm. This protocol uses Signed Ephemeral-Diffie-Hellman exchange to agree on a session key. Hence it does not rely on a public key infrastructure or on certification authorities, in fact ephemeral Diffie-Hellman keys are generated on each session establishment: this allows bypassing the complexity of creating and maintaining a complex third trusted party [6]. These keys will contribute to the generation of the session key and it is transformed to a symmetric key using our key transformation algorithm. This gives protection against Man-in-the-Middle attacks. Symmetric key is preferred over asymmetric key because it is less complex and less time consuming [3]. Using this key the text message is encrypted by the highly secure cipher technique called "xenacrypt" and DES/AES is used for file and voice encryption. Signature is generated for maintaining authenticity and added along with encrypted message which is compressed to reduce the size and transmitted to the receiver. And for integrity, intrusion detection and prevention measures are provided to safeguard data. At the receiver end, the message is verified and checked for authentication, before decrypting the message.

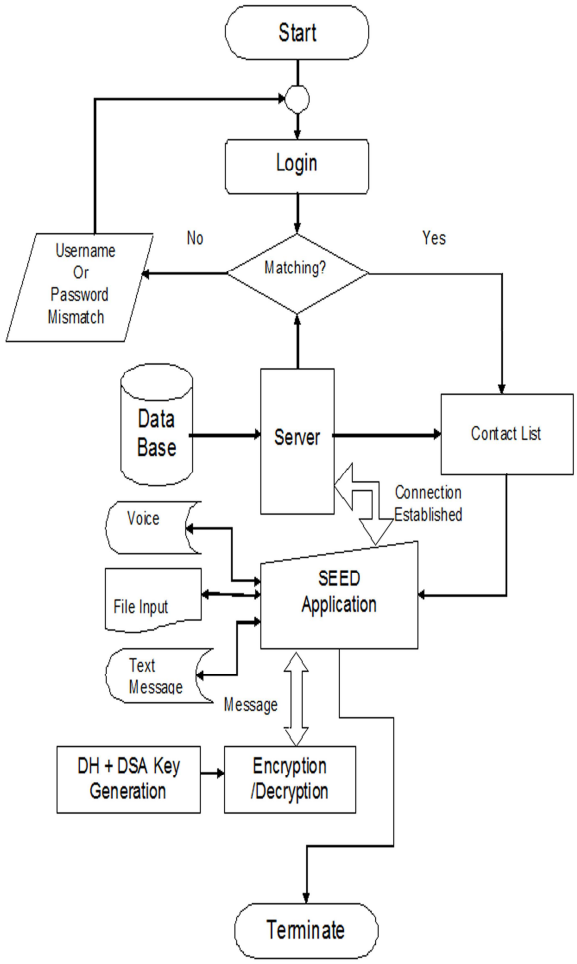

Fig 1 System Architecture

3.1.1 Signed Ephemeral Diffie-Hellman Key Exchange

The key agreement algorithm can be divided into 3 steps:

1. Handshake
2. DSA Signature Exchange
3. Signed Diffie Hellman Key exchange

3.1.1.1 Handshake

The messages exchanged during this phase are called HandShake messages (variable length), to which each party replies with a HandShakeACK message (variable length). The random value is generated, so an eavesdropper can't get to know the number of messages shared between the two parties.

For example, if Alice sends HandShake and Bob replies with HandShakeACK.

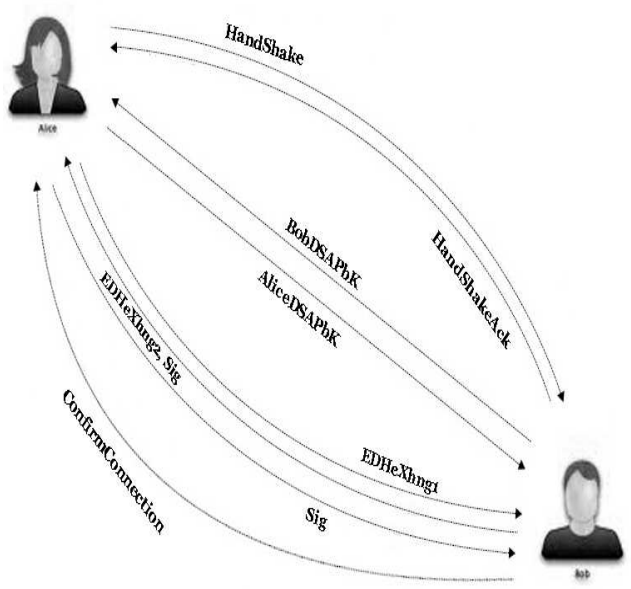

Fig 2 Key Exchange Algorithm

3.1.1.2 DSA Signature Exchange

After the handshake, Bob computes the Digital Signature Algorithm (DSA) keys. Then gets his DSA public key,

$$y_B = g^{\wedge}x_B \bmod q$$

Where g and q are DSA global parameters, x_B is the private parameter.

Then Bob calculates Hy_B as the hash of y_B concatenated with the information of Alice's Handshake message:

$$Hy_B = H(y \mid \text{Alice's HandShake Message})$$

Hence 'y_B' concatenated with 'Hy_B' is called BobDSAPbK messages. Bob sends this message to Alice. On receiving this message Alice validates the message compare received message and computed Hy_B on her side.

$$Hy_B = H(\text{received}(y) \mid \text{Alice's HandShake Message})$$

Once validation is successful, Alice computes her own DSA keys. Her DSA public key is

$$y_A = g^{\wedge}x_A \bmod q$$

Where g and q are DSA global parameters, x_A is the private parameter.

Then Alice computes Hy_A as the hash of y_A concatenated with the information of Bob's Handshake acknowledgement message:

$$Hy_A = H(y \mid \text{Bob's HandShakeACK Message})$$

Hence 'y_A' concatenated with 'Hy_A' is called AliceDSAPbK messages. This message is sent to Bob. On receiving it, Bob validates the message and gets Alice's DSA public key.

3.1.1.3 Signed Diffie Hellman Key Exchange

Three-round protocol of Diffie Hellman integrated with DSA [4] is modified and used here. This protocol supports interactive applications. Let us say user Alice wants to communicate with user Bob interactively. Here, K_{AB} and K_{BA} are the shared secret keys for directions Alice to Bob and Bob to Alice, respectively. Fig. 3 shows the algorithm [4] for this protocol.

Step	User A	User B
1	Select random integer v $m_A = g^v \bmod p$	
	m_A →	
2		Select random integer w $K_{BA} = (y_A)^w \bmod p$ $K_{AB} = (m_A)^{x_B} \bmod p$ $m_B = g^w \bmod p$ $r_B = m_B \bmod q$ $s_B = ((w)^{-1}(H(m_B \| K_{BA} \| K_{AB}) + x_B r_B)) \bmod q$
	← (m_B, s_B)	
3	$K_{AB} = (y_B)^v \bmod p$ $K_{BA} = (m_B)^{x_A} \bmod p$ $r_B = m_B \bmod q$ Verify DSA signature (r_B, s_B) of message m_B $r_A = m_A \bmod q$ $s_A = ((v)^{-1}(H(m_A \| K_{AB} \| K_{BA}) + x_A r_A)) \bmod q$	
	s_A →	
4		$r_A = m_A \bmod q$ Verify DSA signature (r_A, s_A) of message m_A

Fig 3 Integration of Diffie Hellman with DSA

In our modified protocol we generate new key-pairs with new parameter for every session. Hence security can be ensured even with a single session key that is derrived from Diffie Hellman protocol. It is described as below.

The Diffie Hellman public key of Alice is

$$m_A = g^v \bmod p$$

Where g and p are Diffie Hellman global parameters, v is the private parameter.

This key will be used for obtaining the ephemeral shared session key on completion of the protocol. Hence this message is called as EDHeXhng1. On receiving this message Bob computes K_{AB} using m_A. The Diffie Hellman public key of Bob is

$$m_B = g^w \bmod p$$

Where g and p are Diffie Hellman global parameters, w is the private parameter.

Hence the key m_B along with its signature s_B is sent as the message EDHeXhng2, Sig to Alice. On receiving this message Alice computes K_A with the similar technique [10]. Then Alice generates the signature s_A and it is sent as Sig message to Bob. On successful validation of the received signature Bob sends Confirm Connection message to Alice. Now both the parties will have their own shared session key. This protocol provides multiple secret keys, one for each direction. This arrangement conforms to most standard protocols, such as SSL and IPSec [2]. The shared secret key is included in the signature equation along with the message in this scheme. This arrangement prevents the known key attack and the key replay attack. The three-round protocol achieves key confirmation, which prevents the unknown key-share attack.

3.2 Xenacrypt

Xenacrypt is a cryptographic ciphering algorithm where plaintexts are encrypted into a cipher text using different keys and each plaintext can be decrypted from the cipher text using the corresponding key. It is done by using 9x9 boxes similar to a sudoku where a 27 character set including 26 English alphabets and a space is placed thrice(3*27=81) in the row or column wise manner. The length of the cipher text is twice the length of the plain text. There are several possibilities available to form this matrix which we use as keys. The key space is infinite and the key can be potentially any real number.

a	b	c	j	k	l	s	t	u
d	e	f	m	n	o	v	w	x
g	h	i	p	q	r	y	z	
s	t	u	a	b	c	j	k	l
v	w	x	d	e	f	m	n	o
y	z		g	h	i	p	q	r
j	k	l	s	t	u	a	b	c
m	n	o	v	w	x	d	e	f
p	q	r	y	z		g	h	i

Fig 4 Sample Xenacrypt Matrix

Our Xenacrypt cipher works on private-key cryptography which uses symmetric key algorithm. In a symmetric key algorithm, the sender and receiver must have a shared key set up in advance and kept secret from all other parties; the sender uses this key for encryption, and the receiver uses the same key for decryption. Here the shared session key generated by the ephemeral diffie hellman is transformed to symmetric key by using our own key transformation algorithm. In this encryption technique double encryption is done to improve the confidentiality of the data.

For every letter, the substitution of two letters are made corresponding to row and column of the matrix respectively. Now such a cipher is useful because, given a brute force attack on the cipher text, the attacker will encounter a number of messages as only one key will lead to the correct message from the number of available keys, all messages will deceivingly look like the intended message to transmit and the attacker may never know which message is the intended message to transmit. Thus such a cipher is resistant to brute force attacks and cryptanalysis.

3.3 Secure Voice Communication

The secure voice communication module involves the usage of RTP (Real-Time Transport Protocol – application layer protocol) format packets, which is encrypted using AES/DES cryptographic algorithms using the Shared Session key obtained by the Key Exchange algorithm that has been described earlier. This module, receives as input the audio from an input device, predominantly being the microphone. This input is transformed to byte stream, which is encrypted by either AES or DES (using Shared Session key) depending on the bandwidth and load of data, but preferably being the AES for its highly reliable encryption, which is finally compacted into a UDP packets/TCP stream and sent over the network. Hence Alice can have a securely encrypted voice communication with Bob even if someone eavesdrop the transmission line. On receiver side, Bob can receive the

encrypted data, get it decrypted using the corresponding Shared Session key and give it as input to the audio player. This approach of having to send multimedia content across an unreliable network is highly efficient and reliable of providing utmost security against unauthorized intruders or hackers. Usage of AES (strongest symmetric cipher technique), provides high resistance against all known cryptographic attacks, and the use of Signed Diffie-Helmann ensures a safe exchange of public keys from which the desired AES/DES keys are formed. This technique can also be used to share confidential files over the secure connection. Once the session is done the ephemeral keys are shattered in-order to ensure higher level of security.

3.4 Data Compression

In our proposed system, all the data to be transmitted is compressed before encryption in order to strengthen cryptographic security. Because the compressed message has less redundancy than the original data, cryptanalysis is more difficult. In addition to this data compression saves space both for transmission and for storage. Typical zipping algorithm is used for compression therefore it poses simplicity to the system and also high speed of compression is ensured.

3.5 Digital Signature Algorithm

This system uses the Digital Signature Algorithm (DSA) for ensuring both integrity and authenticity. This algorithm is used in the key exchange protocol for authenticating the legitimate sender and receiver. And once the secure connection is established every message that is exchanged between the two parties are accompanied with its corresponding digital signature [9]. In the receiver side this signature can be used for sender validation and also the message integrity verification.

4. IMPLEMENTATION

This system is be implemented entirely in Java and can run on Windows and Linux. Graphics and user input are handled by Java Swing Package and with extra interfaces from the Abstract Window Toolkit (AWT). The program is divided into two main components: the view and the model. The view is accountable for all user input and all graphics output. The user is able to login, handle contacts, maintain logs, create secure connection for text chat, file sharing and voice communication. The very simple user interface can be seen in Fig. 5.

Fig 5 Simple Login Interface

In Fig. 6 we see the exclusive contact management interface. User can add or remove any number of contacts easily. The validation option in the interface is used to invoke the key exchange procedure in order to establish a secure connection. On successful validation, user is permitted to make use of the communication features in the SEED client. In order to ensure key freshness and perfect forward secrecy, a new key is generated at the start of the session and it is destroyed at the end of the session. This provides high level of security to the system.

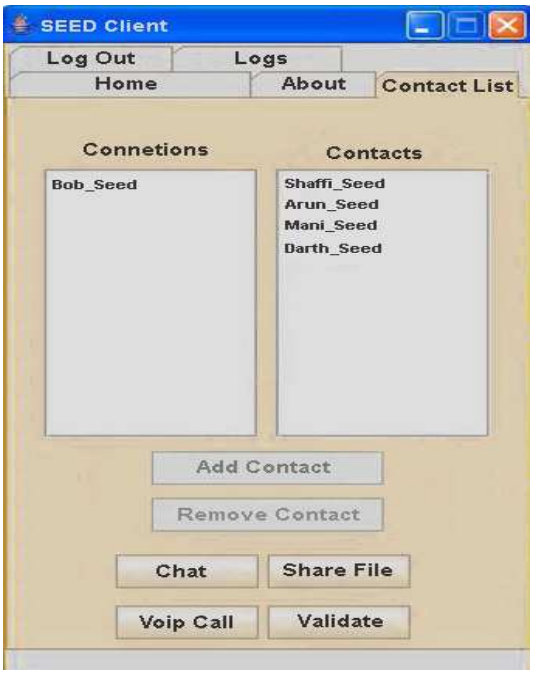

Fig 6 Contact Management Interface

Fig 7 Secure Voice Connection Interface

5. KEY FEATURES

The proposed technique has the following key features:

- Key Freshness
- Perfect Forward Secrecy.
- Less Bandwidth consumption.
- Provides precision control to convert entire message or ile.
- Cipher Text generated for same information is always different due to a new encryption technique.
- High Level of Secrecy in Transmission.
- Authentication of Identity.
- Preservation of data integrity.
- Low Level of Complexity.

6. SIMULATION AND RESULTS

Simulation of this system is done with the help of the AVISPA (Automated Validation of Internet Security Protocols and Applications) tool [5]. AVISPA is a cryptographic protocol verifier, developed by Artificial Intelligence Laboratory, DIST, University of Genova, Italy. The tool processes input files in IF format or in HLPSL format [12].

The latter is a higher level format (in fact it has to be translated into IF format before the protocol can actually be analyzed) [13].The AVISPA analysis relies on four different back-ends:

1. On-the-fly Model-Checker (OFMC);
2. CL-based Attack Searcher (CL-AtSe);
3. SAT-based Model-Checker (SATMC);
4. Tree Automata-based Protocol Analyser (TA4SP).

Only the first two back-ends have been used here, as the others do not support exponentiation.

OFMC is a tool performing protocol verification through the exploration of the transition system described in the protocol on a bounded number of sessions. [11]

CL-AtSe translates the protocol specification into constraints and runs it over a finite number of iterations, after reducing it by means of simplification heuristics and redundancy elimination techniques. [14]

This tool will allow industry and standardizations to automatically validate or detect errors in their products [15]. We tested our system with this AVISPA tool and obtained the result as “SAFE”. This simulation result shows that our system is highly secure and meets the protocol standards which are required by the industry norms.

Fig 8 On-the-fly Model-Checker (OFMC)

Fig 9 Attack searcher (ATSE)

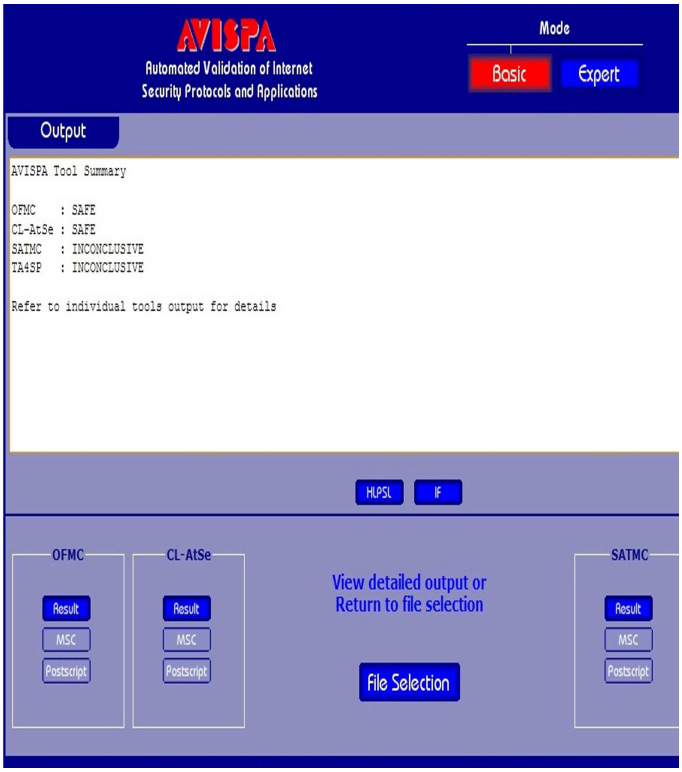

Fig 10 Avispa Results

CONCLUSIONS

Thus in this paper we have introduced a new system to provide a higher level of information security using our proposed idea Safe Exchange of Encrypted Data (SEED). This system uses a novel key exchange protocol for key distribution and agreement, without having to set up a complex Public Key Infrastructure (PKI) and use of endpoint certificates. Also a secure cipher technique has been introduced to guarantee safe transmission of data. We trust that this cipher technique will be a major accomplishment in the field of cryptography which will lead to lot of developments in the future. The identification of the key is very complex and even if the hacker is able to detect the key by some means he will not be able to retrieve the correct message which makes the decryption of message a near impossible by a unknown person or a hacker. This system will be a solution to a number of hacking activities like eavesdropping, masquerading, data modification, identity snooping, denial of service, man-in-the-middle attack and compromised-key attack. In the future this system can also be extended to provide secure multi-client VoIP conferencing, Video conferencing and multimedia file sharing. With the improvement in technology the video data streams can be encrypted and decrypted in a highly secure manner. Our system will hold the key for exploring new scenarios for a many-to-many secure data exchange application development over an unprotected public network. Hence SEED will be a revolution in the field of cryptography and information security.

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to our Prof. V. Kavitha, Department of Computer Science and Engineering, Sri Sairam Engineering College, India, under whose supervision this research was undertaken.

REFERENCES

[1] Gerome Miklau, Dan Suciu, “Enabling Secure Data Exchange,” 2004, [Online Document], Available: http://homepages.inf.ed.ac.uk/wenfei/qsx/reading/miklau-debul-enabling.pdf

[2] Abdel-karim Al Tamimi, Khalid AlHokail, “Secure Data Exchange System: Minimizing Security Attack Risks while Preserving Bandwidth,” paper submitted at Washington University in St. Louis, 2007.

[3] Govind Singh Tanwar, Ganesh Singh and Vishal Gaur, “Secured Encryption - Concept and Challenge,” International Journal of Computer Applications (0975 – 8887) Volume 2 – No.3, May 2010.

[4] Lein Harn, Manish Mehta, and Wen-Jung Hsin, “Integrating Diffie–Hellman Key Exchange into the Digital Signature Algorithm (DSA)” Proc. IEEE Communication Letters Volume 8 – No. 3, March 2004.

[5] AVISPA v1.1 User Manual , 2006 http://www.avispa-project.org/package/user-manual.pdf.

[6] A. C. Yao D. Dolev, "On the security of public key protocols", Information Theory, IEEE Transactions on, 1983.
[7] Ayushi, "A Symmetric Key Cryptographic Algorithm", 2010 International Journal of Computer Applications (0975 - 8887) Volume 1 – No. 15,2010.
[8] Galin Ivanov Zhelyazkov, "Protecting User Privacy in an Untrustworthy Environment", Bachelor of Software Engineering and Management Thesis ISSN: 1651-4769 Report No. 2009-063.
[9] Shafiqul Abidin and Dr. Kumar Balwant Singh, "Authentication of DSS and Secrecy", International Journal of Scientific & Engineering Research Volume 3, Issue 9, September-2012.
[10] Jie Liu and Jianhua Li, "A Better Improvement on the Integrated Diffe-Hellman-DSA Key Agreement Protocol", International Journal of Network Security, Vol.11, No.2, PP.66-69, Sep. 2010.
[11] David Basin, Sebastian Modersheim, Luca Vigano, OFMC: A symbolic model checker for security protocols, Springer-Verlag, 2004, http://www.avispa-project.org/papers/ofmc-jis05.pdf.
[12] HLPSL Tutorial, 2006, http://www.avispa-project.org/package/tutorial.pdf.
[13] The High Level Protocol Specification Language, 2003, http://www.avispa-project.org/delivs/2.1/d2-1.pdf.
[14] Mathieu Turuani, The CL-Atse Protocol Analyser, Springer,2006,http://hal.inria.fr/docs/00/10/35/73/PDF/RTA06_16_Turuani.pdf.
[15] The Intermediate Format, 2003, http://www.avispa-project.org/delivs/2.3/d2-3.pdf

BIOGRAPHIES:

Prof. V. Kavitha M.E, M.S, M Phil, Department of Computer Science and Engineering, Sri Sairam Engineering College, India Acted as co-coordinator for an AICTE sponsored Staff Development Programme on Telemedicine & e-health, Web Services & Applications Presented a paper titled "Secure voter verifiable audit trial", International conference organized by Dept of CSE, E.G.S.Pillay Engineering college, in association with Aichi Institute of Tech., Japan, March 2012.

Mohammed Shaffi .Y currently pursuing Bachelor of Engineering in the department of computer science from Sri Sairam Engineering College, India Deeply interested in the field of high-performance computing & programming, web and information security

Arun Kumar .R currently pursuing Bachelor of Engineering in the department of computer science from Sri Sairam Engineering College, India Deeply interested in the field of Cryptography and Data Structures

Mani Muthiah .M currently pursuing Bachelor of Engineering in the department of computer science from Sri Sairam Engineering College, India Deeply interested in the field of Artificial Intelligence and Network security