Client Authorization and Secure Communication in Online Bank Transactions

Vyshali Rao K P [*], Adesh N D [**], A V Srikantan [***]

[*]M.Tech, CSE Department, Srinivas Institute of Technology, Valachil, Mangalore -corresponding author
[**]Asst. Professor, CSE Department, Srinivas Institute of Technology, Valachi, Mangalore
[***]Divisional Engineer(TM), RTTC, BSNL, Mysore.

Abstract- Network security is the condition of being protected against danger or loss. In the general sense, security is a concept similar to safety. This safety plays a vital role in bank transactions where disclosure of any data results in huge loss. In this paper, Various security threats are illustrated using a tree structure being root nodes as the threats and leaf nodes to achieve those threats and probable measures to overcome the same has been described. security of online bank transactions have been improved by increasing the number of bits used in establishing the SSL connection as well as in RSA asymmetric key encryption along with SHA1 used for digital signature to authenticate the client. Analysis and the results obtained will prove the improved security in proposed method.

Index Terms- Asymmetric encryption, Digital signature, Certificate, entropy

I. INTRODUCTION

A "Network" has been defined as any set of interlinking lines resembling a net, a network of roads parallel and interconnected system, a computer network is simply a system of interconnected computers. Security is often viewed as the need to protect one or more aspects of network's operation and permitted use (access, behavior, performance, privacy and confidentiality included). Security requirements may be Local or Global in their scope, depending upon the networks or internetworks purpose of design and deployment. Criteria for evaluating security solutions include ability to meet the specified needs/ requirements, effectiveness of approach across networks, computing resources needed vis-a-vis the value of the protection offered, quality and scalability, availability of monitoring mechanisms, adaptability, flexibility, practicability from sociological or political perspective economic considerations and sustainability.

Security Attacks compromises the information-system security. Active attacks involve active attempts on security leading to modification, redirection, blockage or destruction of data, devices or links. Passive attacks involve simply getting access to link of device and consequently data. Security Threats are those having potential for security violation. Security Mechanism is a mechanism that detects/ locates/ identifies/ prevents/ recovers from "security attacks" [1]. Security Service is a service that enhances security, makes use of the security mechanisms. The Internet is an integral part of our daily lives as told earlier, and the proportion of people who expect to be able to manage their bank accounts anywhere, anytime is constantly growing. As such, Internet banking has come of age as a crucial component of any financial institution's multichannel strategy. Information about financial institutions, their customers, and their transactions is, by necessity, extremely sensitive; thus, doing such business via a public network introduces new challenges for security and trustworthiness. Any Internet banking system must solve the issues of authentication, confidentiality, integrity, and non repudiation, which means it must ensure that only qualified people can access an Internet banking account, that the information viewed remains private and can't be modified by third parties, and that any transactions made are traceable and verifiable. For confidentiality and integrity, Secure Sockets Layer/Transport Layer Security (SSL/TLS)[2] is the de facto Internet banking standard, whereas for authentication and non repudiation, no single scheme has become predominant yet. Let us see the current authentication threats and the proposed solutions as well as how these solutions can be extended in the face of more complex future attacks.

In this paper, we describe the various security attacks [1] on banking transactions and the proposed solution which include improved SSL/TSL connection [2] between client and bank server and also improved RSA encryption [3]. The simulation of the banking transactions will be performed and the entropy of existing methodology and proposed solution will be compared.

II. ATTACKS ON AUTHENTICATION

Internet banking systems must authenticate users before granting them access to particular services. More precisely, the banking system must determine whether a user is, in fact, who he or she claims to be by asking for direct or indirect proof of knowledge about some sort of secret or credential. With the assumption that only an authentic user can provide such answers, successful authentication eventually enables users to access their private information.

The attack has been classified and categorized as a tree Fig. 1. An attack tree [1] has a root node and leaf nodes. The root node represents the target of the attack, while the leaf nodes represent the means for reaching the target, which are the events that comprise the attack. The attack tree has one root node, representing the final target of the attacker, which is the compromise of the user's bank account. An intruder may use one of the leaf nodes as a means for reaching the target. To categorize Internet banking attacks, each component of the process should be examined: the user terminal/user (UT/U), the

communication channel (CC) and the Internet banking server (IBS). The following types of attacks are identified:

UT/U attacks: These attacks target the user equipment, including the tokens that may be involved, such as smartcards or other password generators, as well as the actions of the user himself/herself. UT/U attacks include:

UT/U1Procedural attacks:

1. UT/U1a: User surveillance (piggybacking) Similar to the personal identification number (PIN) thefts facilitated by the installation of cameras in automatic teller machines (ATMs); the users actions may be monitored to capture credentials.
2. UT/U1b: Theft of token and handwritten note stealing Internet banking usernames are usually long and have to be written down. Users may also keep their passwords written, despite the security guidance provided by their banks. Notes may be stolen, providing access to the users credentials. Tokens may also be stolen, providing the attacker with one authentication factor that, when combined with other types of attacks (such as PIN calculators), can lead to identity theft.

UT/U2: Malicious software installation. The embedding of malicious content for compromising the users login information and password (e.g., keyboard loggers or screen capture in image or video files) may take place via a number of different methods, including:

1. UT/U2a: Hidden code. This is the use of hidden code within a web page that exploits a known vulnerability of the customer's web browser and installs malicious software in the user terminal. The exploit may target permissions on Java runtime support, ActiveX support, multimedia extensions, and running of software through the browser.

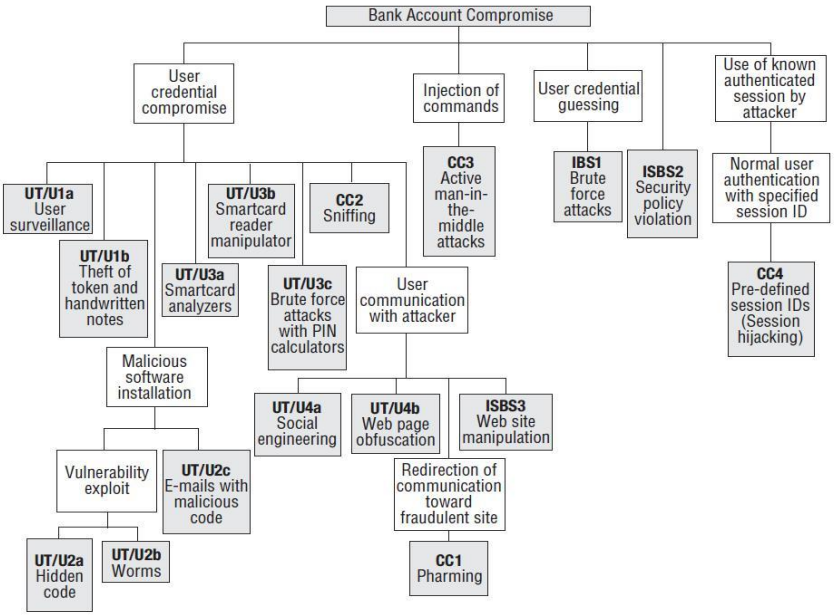

Figure 1: Hierarchy of attacks.

2. UT/U2b: Worms and bots Worms search vulnerabilities and exploit them automatically. This includes the exploit of instant messaging and chatting communication software (that allows the embedment of dynamic content), which may automatically depleted using bots.
3. UT/U2c: E-mails with malicious code This is the submission of e-mails with malicious content, such as executable files or HTML code with embedded applets.

UT/U3Token attack tools:

1. UT/U3a: Smartcard analyzers Attacks against smartcards, such as power consumption analysis or time

analysis, may expose the security of the smartcard by revealing cryptographic keys and passwords. Such attacks are sophisticated and not easy to implement, but are very effective, especially if the necessary countermeasures (noise generators, time-neutral code design) against these types of attacks are not implemented by the smartcard manufacturer.

2. UT/U3b: Smartcard reader manipulator, this is applicable to non certified smartcard readers with insecure interfaces, which may expose the contents of the smartcard by conducting unauthorized operations.
3. UT/U3c: Brute-force attacks with PIN calculators. These attacks focus on breaking the security of tokens that generate random PINs. The attack exploits the fact that a time window is necessary, for synchronization reasons. In some implementations, except from the present PIN, the subsequent and preceding codes are active for the same purpose. It is reported that it is possible to break such mechanisms with a minimum window of three PINs.

UT/U4 Phishing: These attacks use social engineering techniques masquerading as a trustworthy person or business in an electronic communication in an attempt to fraudulently acquire sensitive information, such as passwords and credit card details. The term was initially used in the mid-1990's by hackers who were stealing America Online (AOL) accounts by scamming passwords from unsuspecting AOL users. These attacks include:

1. UT/U4a: Social engineering, these attacks focus on the compromise of the user's credentials by nontechnical means, such as phone calls or the submission of e-mails masquerading as an official bank, asking the user for username and password.

2. UT/U4b: Web page obfuscation: These attacks are based on links that do not correspond to the destination they describe, or the use of Internet Protocol (IP) addresses instead of universal resource locators (URL) for confusing the user. Other techniques deploy hidden frames. These are used for covering the real activity of a web page by using several frames with malicious content, while the user sees only the URL of the master frame set. Other methods use graphics that spoof the interface of a web browser, such as the address bar.

Attack/Authentication Method	Static Password	Soft-token Certificate/ SSL-TLS	Hard-token Certificate/ SSL-TLS	One-time Password/ Time-based Code Generator	Challenge-response	Biometrics	Knowledge-based
UT/U1a: User surveillance	A	X	X	A	X	X	X
UT/U1b: Token/notes theft	A	X	A	A	X	X	X
UT/U2a: Hidden code	A	A	A	A	X	A	A
UT/U2b: Worms	A	A	A	A	X	A	A
UT/U2c: E-mails with malicious code	A	A	A	A	X	A	A
UT/U3a: Smartcard analyzers	X	X	A	A	X	X	X
UT/U3b: Smartcard reader manipulator	X	X	A	X	X	X	X
UT/U3c: Brute-force attacks with PIN calculators	X	X	A	A	X	X	X
UT/U4a: Social engineering	A	X	X	X	X	A	A
UT/U4b: Web page obfuscation	A	X	X	X	X	A	A
CC1: Pharming	A	X	X	A	A	A	A
CC2: Sniffing	A	X	X	A	A	A	A
CC3: Active man-in-the-middle attacks	A	X	X	A	A	A	A
CC4: Session hijacking	A	X	X	A	A	A	A
IBS1: Brute-force attacks	A	X	X	A	X	A	X
IBS2: Security policy violation	A	A	A	A	A	A	A
IBS3: Web site manipulation	A	X	X	A	X	A	A

Legend
A: Applicable
X: Not Applicable

Figure 2: Applicability of attacks in different authentication mechanisms.

CC attacks: This type of attack focuses on communication links. Examples include:

1. *CC1:* Pharming, These involve compromising domain name servers (DNSs), altering DNS tables and connecting the user to fraudulent sites, instead of the official bank's site, where information regarding the users account may be derived.

2. *CC2:* Sniffing, Active sniffing attacks masquerade the two communicating entities to each other (user client and the Internet banking server) to capture information, such as username and password. Passive sniffing captures information from the communication medium, without interception.

3. *CC3:* Active man-in-the-middle attacks, this type of attack regard a schema where the attacker receives and forwards information between the UT and the IBS. The attacker sends malformed user packets or injects new traffic, such as transfer commands, from one account to another.
4. *CC4:* Session hijackings, Attacks that force the user to connect to the IBS with a preset session ID. Once the user authenticates to the server, the attacker may utilize the known session ID to send packets to the IBS, spoofing the user's identity.

IBS attacks: These types of attacks are offline attacks against the servers that host the Internet banking application. Examples include:

1. *IBS1:* Brute-force attacks, Brute-force attacks in certain password-based mechanisms are reported to be feasible by sending random usernames and passwords. The attacked mechanisms implement a scheme based on guessable usernames and four-digit passwords. The attack mechanism is based on distributed zombie personal computers, hosting automated programs for username or password based calculation. This attack may be combined with username filtering methods for determining the identity of the user. These methods filter the different responses of the server, in the case of valid or invalid usernames.
2. *IBS2*: Bank security policy violation Violating the bank's security policy in combination with weak access control and logging mechanisms, an employee may cause an internal security incident and expose a customer's account.
3. *IBS3:* Web site manipulation, Exploiting the vulnerabilities of the Internet banking web server may permit the alteration of its contents, such as the links to the Internet banking login page. This may redirect the user to a fraudulent web site where his/her credentials may be captured. Applicability of attacks in different authentication mechanisms is shown in Fig. 2

III. PROPOSED METHOD

A. *SHA1 algorithm*

SHA-1 [4] is a cryptography hash function designed by the United States National Security Agency that produces a 160-bit (20-byte) hash value. A SHA-1 hash value typically forms a hexadecimal number, 40 digits long. The one-way hash function, or secure hash function, is important not only in message authentication but in digital signatures. The purpose of a hash function is to produce a" fingerprint" of a file, message, or other block of data. To be useful for message authentication, a hash function H must have the following properties:

1. H can be applied to a block of data of any size.
2. H produces a fixed-length output.
3. H(x) is easy to compute for any given x, making both hardware and software implementation practical.
4. For any given code h, it is computationally infeasible to find x such that H(x) = h. A hash function with this property is referred to as one-way or pre image resistant.
5. For any given code h, it is computationally infeasible to find x such that y! = x with H(y) = H(x). A hash function with this property is referred to as second pre image resistant, this is sometimes referred to weak collision resistant.
6. It is computationally infeasible to find any pair (x; y) such that H(x) = H(y).A hash function with this property is referred to as collision resistant. This is sometimes referred to as strong collision resistant.

B. *RSA algorithm*

RSA [5] is a block cipher in which the plain text and cipher text are integers between 0 and n for some n. Encryption and decryption are of the following form period for some plain text block M and cipher text block C: $C = M^e \bmod n$ $M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$. Both sender and receiver must know the values of n and e, and only the receiver knows the value of d. This is a public-key encryption algorithm with a public key of KU(e; n) and a private key of KR(d; n). For this algorithm to be satisfactory for public-key encryption, the following requirements must be met.

1. It is possible to find values of e,d, n such that $M^{ed} \bmod n = M$ for all $M < n$.
2. It is relatively easy to calculate M^e and C^d for all values of $M < n$.
3. It is in feasible to determine d given e and n.

The first two requirements are easily met. The third requirement can be met for large values of e and n.

RSA key generation:

- Choose two large prime numbers p, q.(e.g., 1024 bits each)(Many tests like The Solovay-Strassen Primarily Test [6],Rabin-Miller Primarily Test[7], Ferment Little Test will check the primarily of number
- Compute n = p q, z = (p 1)(q 1)
- Choose e(with e < n) that has no common factors with z. (e, z are "relatively prime").
- Choose d such that ed 1 is exactly divisible by z. (in other words: e dmodz = 1).
- Public key is (n; e). Private key is (n; d).

RSA Encryption:

Given (n; e) and (n; d) as computed above to encrypt [3] bit pattern, m,

- select random numbers,
- Shift the input text to (input value in ASCII+random number) value
- New value becomes the input text, then compute $c = m^e \bmod n$ (i.e., remainder when m^e is divided by n), for random number.

RSA Decryption: To decrypt [3] received bit pattern c, compute $m = c^d \bmod n$ (i.e., remainder when c^d is divided by n), for random number and then shift back to the (random number-ASCII value of text) value.

C. *Certificate*

With the advent of public key cryptography (PKI), it is now possible to communicate securely with untreated parties over the Internet without prior arrangement. One of the necessities arising

from such communication is the ability to accurately verify someone's identity (i.e. whether the person you are communicating with is indeed the person who he/she claims to be). In order to be able to perform identity check for a given entity, there should be a fool-proof method of binding the entity's public key to its unique domain name (DN). A X.509 digital certificate [8] issued by a well known certificate authority (CA) [9], like VeriSign, Entrust, Thawte, etc., provides a way of positively identifying the entity by placing trust on the CA to have performed the necessary verification. A X.509 certificate is a cryptographically sealed data object that contains the entity's unique DN, public key, serial number, validity period, and possibly other extensions. [Note: Refer to RFC 3280 for a complete list of attributes and X.509 v3 extensions.] Certificates are typically stored in PEM (Privacy Enhanced Mail) format.

D. Signature algorithm

The algorithm to generate digital signature [10] is as follows: (Fig 3)

1. Open a input document to be signed.
2. Select the hash function to be used (Here its SHA1)
3. Generate the 160 bit hash value
4. Generate the keys (Here RSA keys)
5. Encrypt the hash value
6. Attach certificate for authentication
7. Generate signature and store in the document.

Signature will now contain: Signature, Length of signature, Encryption algorithm used, Hash function used, Key, Original message

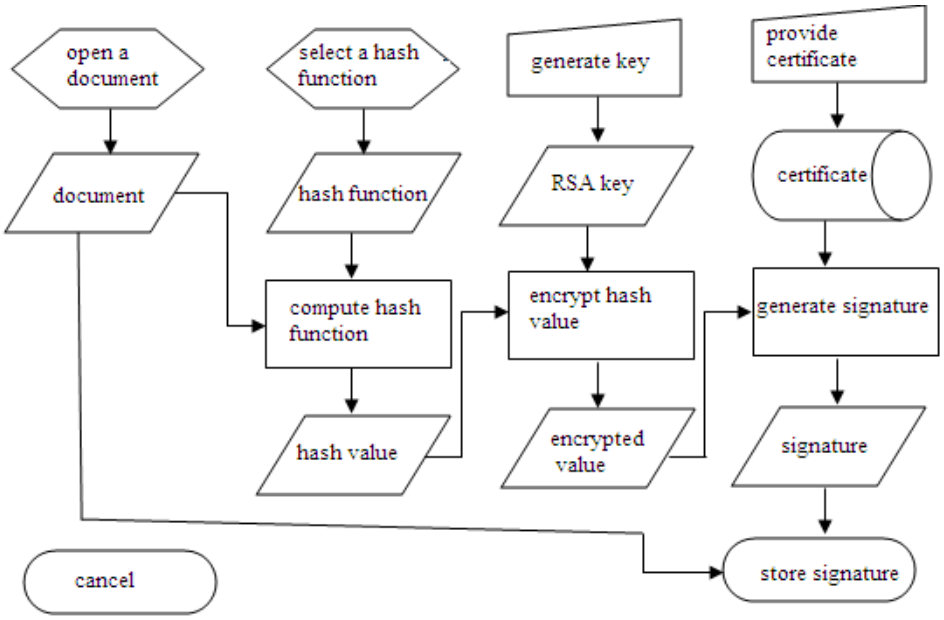

Figure 3: Signature Generation

IV. SIMULATION RESULTS

A. Simulation Environment

Strength of the key to resist the attack on cipher text is simulated with the simulation parameter called entropy; the entropy of a document is an index of its information content. The entropy is measured in bits per character.The information content of a message M[i] is defined by information content
(M[i]) = log (1=p[i]) = log (p[i])……..……. (1)

Where p[i] = Probability that message M[i] = Transmitted by the message source and log denotes logarithms to base2With the aid of the information content of the individual messages, the average amount of information which a source with a specified distribution delivers can be calculated. To calculate this mean, the individual messages are weighted with the probabilities of their occurrence.

Entropy (p[1]; p[2]; :::; p[r]) = [p[1] log(p[1]) + p[2] log(p[2]) + ::: + p[r] log(p[r])]……(2)

The entropy of a source thus indicates its characteristic distribution. It measures the average amount of information which one can obtain through observation of the source or, conversely, the indeterminacy which prevails over the generated messages when one cannot observe the source. List of entropy (Refer Table 1) values for same input text and different key length and there corresponding security is measured. Entropy for a input text where all the possible characters repeat is set as the threshold and related to that value conclusion is made that the entropy nearer to the threshold is better encryption method. Frequency of letters in input text and the histogram is plotted (Fig. 4), more the frequency of occurrence of character more the attack expected on that character. Values in Table 1 are plotted as Fig.6 and Fig. 7.

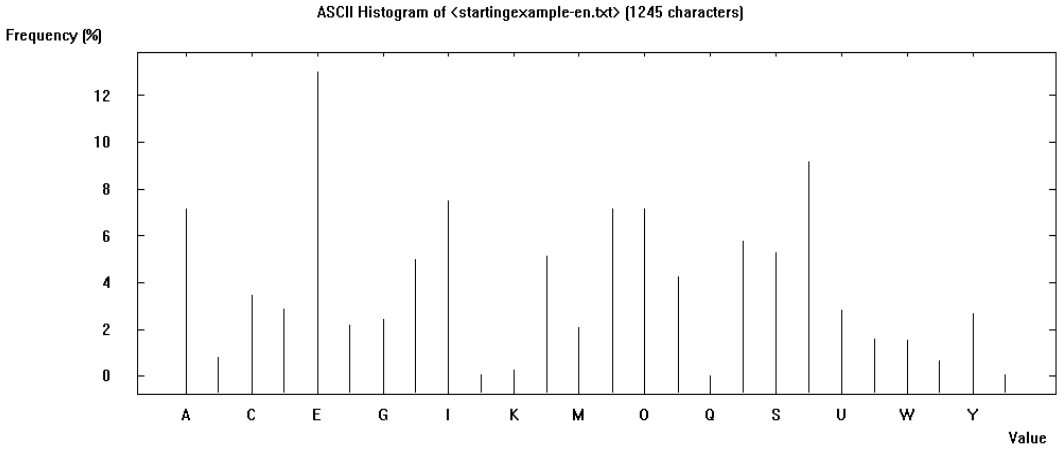

Figure 4: Histogram of character in plain text

Table 1: Strength of variable key bits in RSA

RSA Key length	Time to generate key (sec)	Entropy	Max. Entropy
521 bits	0.059	7.88	8.00
768 bits	0.146	7.80	8.00
1024 bits	0.369	7.87	8.00
2048 bits	3.428	7.95	8.00

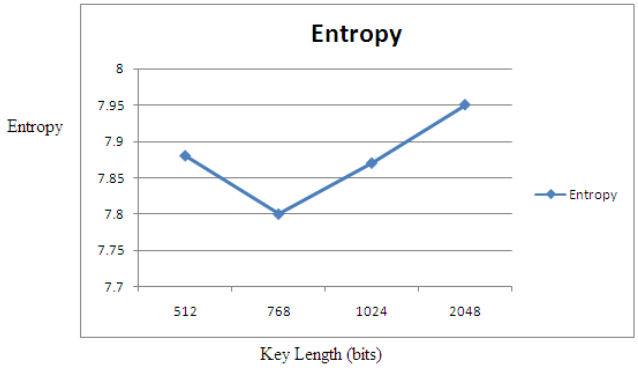

Figure 6: Graphical representation

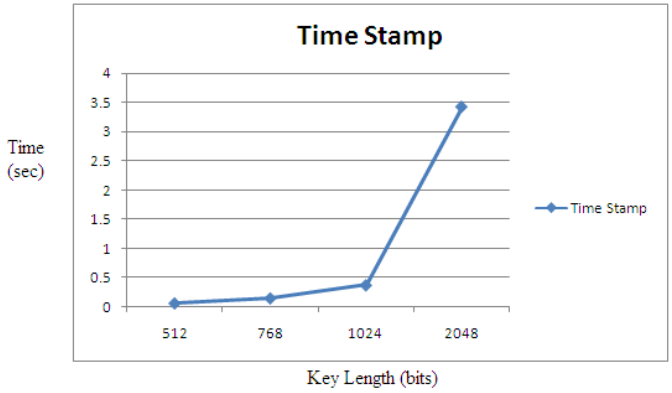

Figure 7: Time stamp required for key generation.

V. CONCLUSION

This paper analyses the various security threats for computer networking, various loop holes of present networking. These threats overcame by various methodologies for securing the network through cryptography and encryption. Effort was made to find out the security aspect of Networking and it was overcome by means of Cryptography and Encryption by using improved RSA algorithm and also increased number of bits in SSL connection.

Even though key generation time is more compared to that of present scenario, security can be assured which is more important than key generation time in the current scenario.

ACKNOWLEDGMENT

I heart fully thank my guide Mr. Adesh N D and my co-guide Mr. A V Srikantan for their constant support and guidance in completing my project and thesis.

REFERENCES

[1] Christos K. Dimitriadis, ”Analyzing the Security of Internet Banking Authentication Mechanisms”2007 ISACA

[2] Larry Seltzer, Security Analyst and Writer,” Securing Your Private Keys as Best Practice for Code Signing Certificates”

[3] Rajan.S.Jamgekar, Geeta Shantanu Joshi, ”File Encryption and Decryp-tion Using Secure RSA”. International Journal of Emerging Science and Engineering (IJESE) ISSN: 23196378, Volume-1, Issue-4, February 2013

[4] Nalini C. Iyer and Sagarika Mandal, Implementation of Secure Hash Algorithm-1 using FPGA, International Journal of Information and Com-putation Technology. ISSN 0974-2239 Volume 3, Number 8 (2013),

[5] P. Kitsos, N. Sklavos and O. Koufopavlou, An efficient implementation of Digital Signatire algorithm. VLSI Design Laboratory Electrical and Computer Engineering Department University of Patras. Patras, GREECE

[6] Yung-Chieh Lin; Yi-Ping Hung, Zen-chung Shih, The Solovay-Strassen Primality Test ,12 October, 1993 Burt Rosenberg Re-vised: 6 October, 2000

[7] Rudin,S. Osher. Rabin-Miller Primality Test.

[8] S Beucher, ”X.509 Certificate Generator User Manual”,

[9] ”Types of certification authorities”, Microsoft Certificate Authorities from Microsoft Technet.

[10] “S.R. Subramanya and byung K. YI ”Digital signatures”, IEEE March/April 2006.

AUTHORS

First Author – Vyshali Rao K P, M.Tech, CSE Department, Srinivas Institute of Technology, Valachil, Mangalore, Email:raovyshali@gmail.com

Second Author – Adesh N D, Asst. Professor, CSE Department, Srinivas Institute of Technology, Valachi, Mangalore

Third Author – A V Srikantan, Divisional Engineer(TM), RTTC, BSNL, Mysore.