International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

SSO-Key Distribution Center Based Implementation Using Symmetric Encryption Technique for Distributed Network (Securing SSO in Distributed Network)

M Durga Prasanna[1], Roopa S R[2]

[1]Department of CSE, Mangalore Institute of Technology and Engineering, Moodabidri,-574225, India

[2]Assistant Professor, Department of CSE, Mangalore Institute of Technology and Engineering, Moodabidri-574225, India

Abstract: ***Single sign-on (SSO) is a session/user authentication process that permits a user to enter one name and password in order to access multiple applications. The process authenticates the user for all the applications they have been given rights to and eliminates further prompts when they switch applications during a particular session. Unforgeability, credential privacy, and soundness are the basic requirements of any SSO scheme. Chang and Lee proposed new SSO scheme and claimed its security by providing well organized security arguments. But their scheme is actually insecure as it fails to meet credential privacy and soundness of authentication. Specifically, their scheme suffers from two severe attacks. The first attack allows a malicious service provider, who has successfully communicated with a legal user twice, to recover the user's credential and then to impersonate the user to access resources and services offered by other service providers. In another attack, i.e. an outsider attack, by an unauthorized or illegitimate user of the system may be able to enjoy network services freely by impersonating any legal user or a nonexistent user. So to overcome these drawbacks, we propose an improvement to Chang and Lee SSO scheme by making use of efficient symmetric key encryption technique of SERPENT key signatures.***

Keywords: Authentication, authorization, distributed computer networks, information security, single sign-on (SSO).

1. Introduction

1.1 Aim of the Paper

This project provides research into user authentication and focuses particularly on single sign-on scheme. This work is based on java application. This mechanism will reduce the fatigue of the password again and again. This can be very secure to implement the SSO scheme using symmetric encryption.
The aims of this paper are as follows:

1. The several single sign-on schemes have been proposed. However, most of them have security flaws, and even worse, their improvements are also insecure against possible attacks. Thus, this paper aims to give an approach into the most recent SSO schemes identifying their flaws, issues and challenges.

2. The second aim of this paper is to formalize the single sign-on and its security model to formally resolve the issues identified. Also, an efficient and provably secure single sign-on authentication scheme without the identified drawbacks will be provided according to the formal model.

1.2 Background

Entity authentication is becoming more and more important. With the widespread use of distributed computer networks, for example, cellular networks, virtual reality communities, World Wide Web, peer-to-peer networks and multiplayer online games, there is a need to be more alert about the security and privacy of users. One way to address the security and privacy concerns is remote user authentication and this is widely used in distributed systems for identifying users and servers. Remote user authentication is a means of identifying a user and verifying whether this user has permission to access the network services and resources. However, an attacker may impersonate a server to communicate with a user and then, the attacker is able to steal the user's information. Thereafter, the attacker can pass authentication with the real server by using the stolen information of the user. Therefore, mutual authentication is needed in order to prevent bogus server attacks. Other requirements of user authentication include ensuring the confidentiality of further exchanging messages, protecting user privacy, providing user anonymity and achieving un-link ability. In the complex environments of computer networks, however, it is a challenge to design efficient and secure mutual authentication protocols under such security requirements.

This paper provides efficient and secure identification services with further security requirements for users in distributed systems and networks. In general, the identification services may require three factors, i.e., password, symmetric key signature characteristics. The authentication which based on password is called password-based authentication. Password-based authentication together with another factor, symmetric key , is called two-factor authentication. In which, a successful user authentication can be achieved if the user has a correct password together with a corresponding signature The two-factor authentication consists all of these three factors, i.e., password, symmetric key signature characteristics. There is another concept which belongs to two-factor authentication, called single sign-on

(SSO). It enables a user to use a unitary secure credential (or token) to access multiple computers and systems where he/she has access permissions.

1.3 Challenges

The need for authentication of individual identity is a fundamental requirement in our society. In this computer age, single sign-on is a highly desirable solution for user authentication, suiting most common users since it reduces requirements for multiple logins and for remembering multiple IDs/passwords. This also alleviates forgotten password problems. Unfortunately, there are some shortcomings in the existing schemes such as (a) the inability to preserve user anonymity properly, (b) vulnerability to possible attacks, e.g. impersonation attacks, (c) a seeming absence of formal study and proof on soundness of the single sign-on, (d) the requirement for additional time-synchronized mechanisms, (e) lower efficiency and higher cost. Thus, it is a challenge to design an efficient and provably secure single sign-on scheme in distributed networks.

1.4 Objective

By make use of symmetric encryption technique called efficient verifiable encryption of RSA signatures for Detect and mitigate the *"credential recovering attack*" and *"impersonation attack without credentials" in* user authentication phases.
The main objectives of this project are:

- Implementing serpent encryption technique.
- Analyzing Distributed computer security.
- Decreasing the burden of users from authenticating themselves again and again.
- Reducing password fatigue from different user name and password combinations.
- Analyzing the RSA-VES encryption algorithm with modified serpent encryption technique.

2. Related Work

A. Guidelines for Technical Preparation

In paper, **"Improving the Security of SSO in Distributed Computer Network using Digital Certificate and one Time Password (OTP)"**, we study about the different SSO scheme and their differences between the following SSO schemes.

Authentication information and Replication

The simple SSO scheme (architecture) is authentication database replication. Clients can authenticate to a centralized authentication system (server) and the server stores information for current time logged in clients (online clients)in a session database. This database is send all the data to all the server. In essence, the authentication server acts as the "master" server (holder) of the authentication database and all the other servers are the "slaves" server (holder). When a client contacts another server to request for a service, the server authenticates the client based on its copy of the authentication database (which store in master server (holder)) and allows the client to connect if the client is found in the replicated database.

Token Based

In token based SSO scheme a client can authenticate to authorization system (server) and server send a token to client. It is one type of cryptographic token it is like a secrete key. The client uses this token to prove the identity to each application serve it wants to access. The server does some mathematical calculation (cryptographic processing), some calculation on the token to verify the identity of the client and validate of the token which is created by server. Tokens are rely on shared secret keys. Token describe the trust between two parties (application server and the authentication server). The basic example of a token-based SSO scheme is the Kerberos authentication protocol it describe additional tokens called tickets in Kerberos and in addition client server messages for single sign on.

PKI-Based

Public key infrastructure (PKI)-based Security technique is secure method in network security. In this scheme SSO require that user register to a certification authority (CA).In this process registration it is validate(identify) with credential generation of private key, public key and the creator of user certificate by CA. Digital certificate contains their unique public key and serial number. Here the CA is the main trusted party (authority) which gives a digital certificate. With the use PKI, generate the digital certificate. Digital certificate is a small file in there is a public key and serial number. Digital certificate establish a connection between a user and a public key, so digital certificate must contain user name and user's public key.

Proxy-Based

In a proxy-based SSO scheme, the user authenticates to the centralized authentication system(server), and the authentication server itself supplies the user personal data (e.g., username and password) to the similar (appropriate) server whenever the user can send requests for use an application services on another server. Proxy-based SSO is used often when different servers have different authentication mechanisms to use the services and the client (user) has multiple sets of credentials (e.g., username and password) to authenticate their identity.

Fundamental of Cryptography

Confidential communication has long been a common practice in the social life. However, as information can be communicated electronically, it is exposed in public domain and unavoidably resulted in interceptions. A scientific approach to respond the demands on achieving the sense of security is cryptography. The term cryptosystem, also called cipher, is often used in cryptography. Intuitively, its meaning is clear enough which refers to an encryption system.

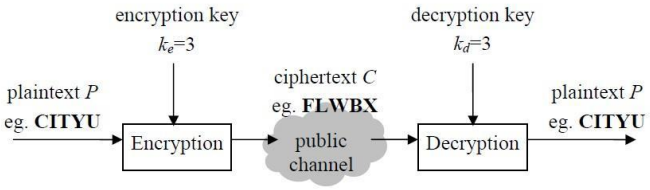


Figure 1: The encryption process performed by Caesar cipher

The central idea of encryption is to transform the message in which its original information can only be reconstructed by a designated recipient. By definition, a message in its original form is known as plaintext *P* and the information concealed in an unintelligible form is known as cipher text *C*. The encryption process consists of an algorithm and a key. It is generally described as $C = E(P, ke)$, where *ke* is the encryption key and $E(\ )$ is the encryption algorithm. Therefore, the cipher text *C* can be transmitted over public channels without exposing the information it
represents . Similarly, a corresponding decryption process is the reverse of encryption which is based on the cipher text *C* with decryption key *kd* for the reconstruction of the original plaintext:

$P = D(C, kd)$, where $D(\ ) = E^{-1}(\ )$.
As of today's information security, both two branches of cryptosystems have a significant importance and one cannot substitute another. A more detailed discussion on these two cryptosystems will be given in the following sections.

Private Key Cryptography

In brief, the principle of private-key cryptography, as shown in Figure 2, is based on the fact that the sender and receiver agree on a common secret key *k* before they can communicate securely. Similar to the generic encryption model described in Figure 1, the cipher text *C* is unintelligible without the aid of the secret key *k*. Such an unintelligible piece of information can finally be transformed back into the original plaintext *P* by the receiver possessing the same key. However, it should be stressed that a secure channel between the parties for key agreement is critical but practically inconvenient to follow. This refers to the key distribution problem. As a remedy, key distribution center (KDC) together with some associated protocols is suggested for the secret key establishment. (Key distribution problem)
Ex: DES, AES

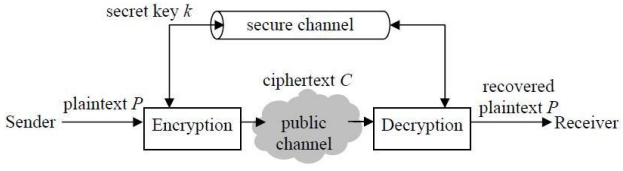


Figure 2: Private-key cryptography scenario.

Advanced Encryption Standard

Since the security deficits are found in DES, the need for a stronger alternative has been officially declared by NIST. After calls for proposal, a Belgian cipher, Rijndael has eventually been adopted as the Advanced Encryption Standard(AES), a successor of DES in 2001. It is also an iterated block cipher with a scalable key length which can be 128, 192 or 256 bits. In the core of AES algorithm, there is no Feistel cipher-like structure. However, the entire block of input data can be processed in parallel and intertwined with operations such as substitutions, row shifting, and column mixing and round key additions. In this regard, the new AES with an expanded key length has many potential advantages over other block ciphers by offering a more secure and faster implementation. Many recent security applications have been migrated to meet this new standard.

Serpent Encryption Technique

Serpent [7] was designed by Ross Anderson, Eli Biham and Lars Knudsen as a candidate for the Advanced Encryption Standard. It has been selected as one of the five finalists in the AES competition. Serpent is faster than DES and more secure than Triple DES. It provides users with a very high level of assurance that no shortcut attack will be found. To achieve this, the algorithm's designers limited themselves to well understood cryptography mechanisms, so that they could rely on the wide experience and proven techniques of block cipher cryptanalysis. The algorithm uses twice as many rounds as are necessary to block all currently known shortcut attacks. This means that Serpent should be safe against as yet unknown attacks that may be capable of breaking the standard 16 rounds used in many types of encryption today. However, the fact that Serpent uses so many rounds means that it is the slowest of the five AES finalists. But this shouldn't be an issue because it still outperforms Triple DES.

Serpent is a 128-bit block cipher, meaning that data is encrypted and decrypted in 128-bit chunks. The key length can vary, but for the purposes of the AES it is defined to be either 128, 192, or 256 bits. This block size and variable key length is standard among all AES candidates and was one of the major design requirements specified by NIST. The Serpent algorithm uses 32 rounds, or iterations of the main algorithm.

B. Existing System

Chang and Lee proposed a new SSO scheme and claimed its security by providing well-organized security arguments. And it demonstrates that their scheme is actually insecure as it fails to meet credential privacy and soundness of authentication. Specifically, presents two impersonation attacks. The first attack allows a malicious service provider, who has successfully communicated with a legal user twice, to recover the user's credential and then to impersonate the user to access resources and services offered by other service providers. In another attack, an outsider without any credential may be able to enjoy network services freely by impersonating any legal user or a nonexistent user
The existing scheme resist following attacks.
1. Credential recovering attack.
2. Impersonation attack without credentials.

The existing SSO scheme also consists of three phases: *system initialization*, *registration*, and *user identification* phase are shown as follows.

Initialization Phase

SCPC selects two large safe primes p and q to set N= pq. Namely, there are two primes p' and q' such that p= 2p' + 1 and q= 2q' + 1. SCPC now sets its RSA public/private key pair (e, d) such that ed = 1 mod 2p'q', where e is a prime. Let QN be the subgroup of squares in Z*N whose order is G =p'q' unknown to the public but its bitlength lG= |N| - 2 is publicly known. SCPC randomly picks generator g of QN , selects an ElGamal decryption key u, and computes the corresponding public key y = gu mod N. In addition, for completing the Diffie-Hellman key exchange SCPC chooses generator g' Є Z*N, where n is another large prime number. SCPC also chooses a cryptographic hash function h (.): {0, 1}* -> {0, 1} k, where security parameter k satisfies 160 <= k <= |N|-1. Another security parameter e >1 is chosen to control the tightness of the ZK proof. Finally, SCPC publishes (e, N, h (.), g, y, g', n), and keeps (d, u) secret.

Registration Phase

In this phase, upon receiving a register request, SCPC gives Ui fixed-length unique identity IDi and issues credential Si = h (IDi) 2d mod N . Si calculated as SCPC's RSA signature on h (IDi)2 is an element of QN, which will be the main group we are calculating. Each service provider with identity IDi should maintain a pair of signing/verifying keys for a secure signature scheme (not necessarily RSA). σj(SKj, Msg) denotes the signature σj on message Msg signed by Pj using signing key SKj. Ver(PKj, Msg, σj) denotes verifying of signature σj with public key PKj , which outputs "1" or "0" to indicating if the signature is valid or invalid, respectively.

Authentication Phase

In this phase, RSA-VES is employed to authenticate a user, while a normal signature is used for service provider authentication. The details are illustrated in Fig. 2 and further explained as follows.

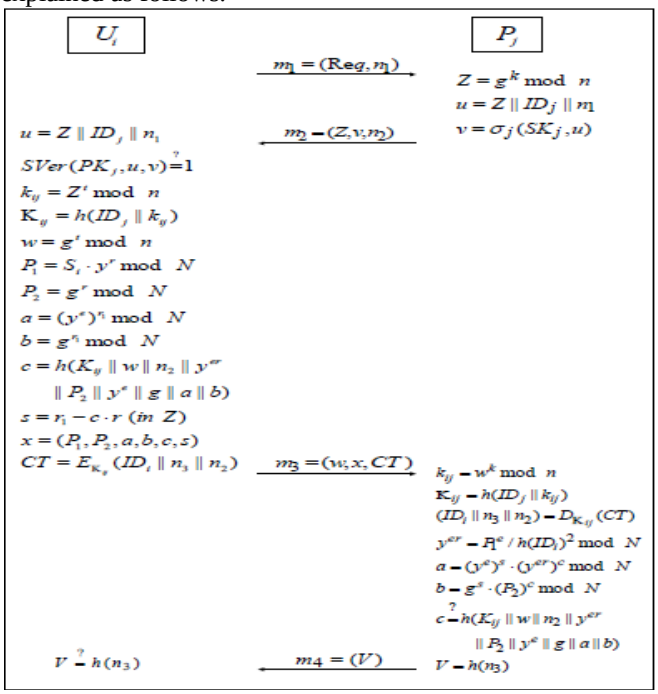


Figure 3: Existing SSO scheme

1) Ui sends a service request with nonce *n1* to service provider *Pj*.

2) Upon receiving (Req, n1), Pj calculates its session key material *Z = gk mod n* where *k Є Z*n* is a random number, sets *u = Z||IDj||n1*, issues a signature *v= σj(SKj, u)* and then sends *m2 = (Z,v, n2)* to the user, where *n2* is a nonce selected by *Pj* .

3) Upon receiving *m2= (Z, v, n2), Ui* sets *u= (Z|| Idj || n1).* Ui terminates the conversation if *Ver (PKj, u, v) = 0*. Otherwise, *Ui* accepts service provider *Pj* because the signature v is valid. In this case, Ui selects a random number t Є Z*u to compute *w =gt mod n, kij = Zt mod n*, and the session key *Kij = h (IDj || kij)* . For user authentication, Ui first encrypts his/her credential *Si as (P1 = Si. yr mod N, P2 = gr mod N)*, where r is is a random integer with binary length lG. Next, Ui computes two commitments *a= (ye) r1 mod N* and *b=gr1 mod N*, where r1 *Є ± {0,1} Є (lG+k)* is also a random number. After that, Ui computes the evidence showing that credential Si has been encrypted in *(P1, P2)* under public key y. For this purpose, *Ui* calculates *c= h (Kij|| w || n2 ||yer || P2 ||ye ||g ||a ||b) and s = r1 –c.r.* Then, *x = (P1, P2, a, b, c, s)* is the NIZK proof for user authentication. In fact, it is precisely, the processes of generating which is the proof part of RSA-VES.

Finally, encrypts his/her identity *IDi*, new nonce *n3*, and *Pj*'s nonce *n2* using session key *Kij* to get cipher text *CT = EKij (IDi || n3 ||n2)*, and thereafter sends *m3 = (w, x, CT)* to service provider *Pj*.

4) To verify Ui, Pi calculates kij = wk mod n , the session key *Kij = h (IDj || kij)* , and then uses *kij* to decrypt *CT* and recover *(IDi, n3, n2)*. Then, computes *yer =P1e/h (IDj) 2 mod N, a = (ye) s. (yer) c mod N, b= gs. Pc 2 mod N*, and checks if *(c, s) Є {0,1} k. ± {0, 1} Є (lG+k) +1 and c= h (Kij|| w || n2 ||yer || P2 ||ye ||g ||a ||b)*. If the output is negative, *Pj* aborts the conversation. Otherwise, *Pj* accepts *Ui* and believes that they have shared the same session key Kij by sending Ui, m4 = (V) where V= h (n3).

5) After Ui receives V, he checks if V= h (n3). If this is true, then Ui believes that they have shared the same session key Kij. Otherwise, Ui terminates the conversation.

3. Proposed System

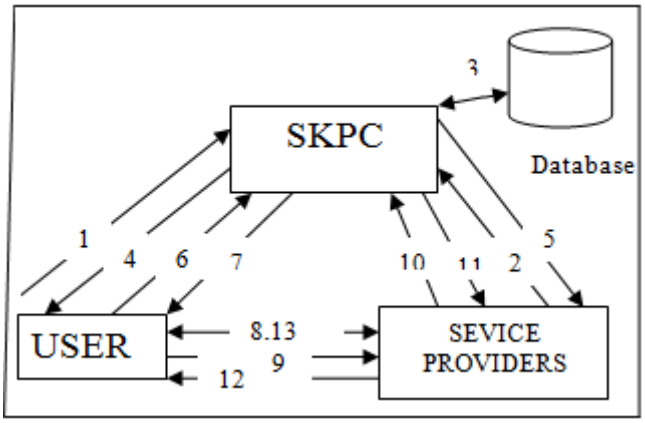
To overcome the flaws in the Chang-Lee scheme, propose an improvement by employing an symmetric encryption technique(SERPENT), which is an efficient primitive introduced for realizing fair exchange of symmetric key signatures. Serpent key Signature comprises three parties: a trusted party and two users say Alice and Bob. The basic idea of SERPENT KEY is that Alice who has a key pair of signature scheme signs a given message and encrypts the resulting signature under the trusted party's private key, and uses a no interactive zero-knowledge (NZK) proof to convince Bob that she has signed the message and the trusted party can recover the signature from the cipher text. After validating the proof, Bob can send his signature for the same message to Alice. For the purpose of fair exchange, Alice should send her signature in plaintext back to Bob after accepting Bob's signature. If she refuses to do so, however, Bob can get her signature from the trusted party by providing to Alice's signature and with that signature Bob can decrypt the cipher texte, so that the trusted party can recover Alice's

signature and sends it to Bob, meanwhile, forwards Bob's signature to Alice. Thus, fair exchange is achieved.

The proposed SSO scheme also consists of three modules:
1. User module.
2. SKPC module.
3. Service Provider module.

The figure 4 shows the proposed SSO scheme which is very secure than the existing SSO scheme. The proposed system uses symmetric encryption technique that can be secure to avoid the attacks presents in the existing system. The attacks are like impersonate attack, which consists two sub attacks 1.malicious service provider attack and 2. Outsider attack.
Description about proposed modules:

1) The first module called the **User**, used to request and accesses the services after the valid authentication between the service providers with symmetric serpent key signature.
2) The second module called **SKPC** (Symmetric Key Provider Center), used to generate the signature and sends to the user for the authentication process as shown in the figure 4.
3) The third module called **Service Provider**, used to provide the services after valid authentication between the user and service provider.



1) Registration of user to SKPC.
2) Registration of Service Provider to SKPC.
3) Generating key signature and store key signature and the user, Service Provider details into the database.
4) User Registration is successful.
5) Service Provider Registration is successful.
6) Login to the SKPC and request the Key Signature.
7) Login is successful and sends the requested Key Signature by using encryption technique.
8) Authenticating User to the Service Provider by using encrypted Key Signature.
9) View the services from Service Provider.
10) Send the User Key Signature to the SKPC for the verification.
11) Service Provider receives valid or invalid user Key Signature.
12) If User is valid, provide the services from the Service Provider.
13) Request the services to the Service Provider.
14) Send the requested services to the User.

The algorithm which is used in the login and authentication process of Encryption is serpent [7] to access the services and for valid authentication. The steps involved in the serpent algorithm are as shown below:

1. A 64 bit blocks of cipher with a variable key length.
2. There is a P-array and four 32-bit S-boxes. The P-array contains 18 of 32-bit subkeys, where each S-box contains 256 entries.
3. The algorithm consists of two parts: a key-expansion part and a data-encryption part.
4. Key expansion converts a key of at most 448 bits into several subkey arrays totaling 4168 bytes.
5. The data encryption occurs via a 16-round Feistel network. Each round consists of a key-dependent permutation, and a key and data-dependent substitution.
6. All operations are XORs and additions on 32-bit words.
7. The input is a 64 bit data element.

4. Conclusion

In this paper, we deal with two impersonation attacks on Chang and Lee's single sign-on (SSO) scheme [11]. The first attack shows that their scheme cannot protect the privacy of a user's credential, and thus, a malicious service provider can impersonate a legal user in order to enjoy the resources and services from other service providers. The second attack violates the soundness of authentication by giving an outside attacker without credential the chance to impersonate even a non-existent user and then freely access resources and services provided by service providers. We also discussed why their well-organized security arguments are not strong enough to guarantee the security of their SSO scheme. Furthermore, by employing an efficient symmetric key encryption of SERPENT signatures introduced by Ateniese [7], we proposed an improved Chang–Lee scheme to achieve soundness and credential privacy. As future work, it is interesting to formally define authentication soundness and construct efficient and provably secure single sign-on schemes. Based on the draft of this work [8], a preliminary formal model addressing the soundness of SSO has been proposed in [8]. Further research is necessary to investigate the maturity of this model and study how the security of the improved SSO scheme proposed in this paper can be formally proven.

References

[1] W. B. Lee and C. C. Chang, "User identification and key distribution maintaining anonymity for distributed computer networks," *Comput.Syst. Sci. Eng.*, vol. 15, no. 4, pp. 113–116, 2000.
[2] Y. Xu, R. Song, L. Korba, L.Wang, W. Shen, and S. Y. T. Lang, "Distributed device networks with security constraints," *IEEE Trans. Ind.Inf.*, vol. 1, no. 4, pp. 217–225, Nov. 2005.
[3] K. V. Mangipudi and R. S. Katti, "A secure identification and key agreement protocol with user anonymity (SIKA)," *Comput. Security*, vol. 25, no. 6, pp. 420–425, 2006.
[4] C.-C. Chang and C.-Y. Lee, "A secure single sign-on mechanism for distributed computer networks," *IEEE*

Trans. Ind. Electron., vol. 59, no. 1, pp. 629–637, Jan. 2012.

[5] Guilin Wang, Jiangshan Yu and, Qi Xie, "Security Analysis of a Single Sign-On Mechanism for Distributed Computer Networks," IEEE Trans.Ind.Info., VOL:9 NO:1 2013.

[6] NeetuSettia. "Cryptanalysis of modern Cryptography Algorithms". International Journal of Computer Science and Technology. December 2010.

[7] Ross Anderson, Eli Biham "Serpent: A Proposal for the Advanced Encryption Standard", IEEE Trans.ind.info,2010.

[8] Guilin Wang, Jiangshan Yu, and Qi Xie" Security Analysis of a Single Sign-On Mechanism for Distributed Computer Networks",IEEE Trans.ind.info,2013.

[9] Shashi Mehrotra Seth, Rajan Mishra "Comparative Analysis Of Encryption Algorithms For Data Communication" IJCST Vol. 2,2011

[10] DiaasalamaAbdElminaam, HatemMohamadAbdual Kader, Mohly Mohamed Hadhoud, "Evalution the Performance of Symmetric Encryption Algorithms", international journal of network security vol.10,No.3,pp,216-222,May 2010.

[11] Ragendu .T.B1, Dr. R. Manimegalai "Improving Security of Single Sign-On Mechanism for Distributed Service Enviornment", IJCST Vol. 2, 2014

Author Profile

M Durga Prasanna , pursuing M.Tech.in Computer Networks Engineering, Mangalore Institute of Technology & Engineering, Moodabidri (DK), under Visvesvaraya Technological University, Belgaum-590014, (Karnataka), India

Ms Roopa S R, working as Assistant Professor, Department of Computer Science and Engineering in Mangalore Institute of Technology & Engineering, Moodabidri (DK), under Visvesvaraya Technological University, Belgaum-590014, (Karnataka), India