A Modified Approach to KERBEROS Authentication Protocol using Improved BLOWFISH Scheme

[1]Hemraj Shobharam Lamkuche, [2]T. Santha
[1,2]Dept. of IT & Science, Dr. G. R. Damodaran College of Science, Coimbatore, TamilNadu, India

Abstract
An authentication mechanism is the most basic requirement for developing a highly secure environment. Authentication itself means confirming the truth of data. Since the information stored on the computer which is transmitted over internet need to ensure security and measuring safety to information, so without our awareness, the intruders can attempt to obtain and gain control over the system to access secure confidential information by using most common method like DoS Attack, Backdoor, Trojan horse, Packet Sniffing, etc. So there must not be any compromise in securing our resources. Hence Cryptography is arguably the most important tool a security expert has [1]; Cryptography only allows the solution for different security zones including Authentication Mechanism, Access Control, Data Confidentiality, Data Integrity, and Non-Repudiation. In this paper we have strongly focused to improve Kerberos authentication protocol to be more secure and unbeatable [2]; as we have also make use of the improved Blowfish encryption algorithm which is a block cipher with fixed 64bit block size and key size varies from 32 – 448 bits which will enhanced more security to the Kerberos network realm.

Keywords
Kerberos, Blowfish, Security, Session and Cryptography

I. Introduction

Secure Authentication is necessary to each and every organization system which requires more protection against today's sophisticated attacks. Since attackers continuously keep a track on your organization. Organization may consider deploying strong authentication mechanism to prevent unauthorized access to confidential information which is stored on the network resources. But moreover abnormal user demands stronger authentication mechanism and also it does interfere with the normal user experience. So, most of the framework environment today provides very strong authentication protocols, that may offer high security for them. Since the global arena is interconnected to each other over internetworking in which multiple users shares multiple resources anonymously. This results in communication between users over internet. Internet is used in many ways to conduct and to increase the business with other people, institute, organization etc over the world by mean of transmitting the information to and from the users. But when the question comes to security matter "How do we protect our most valuable assets over Internetwork?" when we talk about computer security, we mean that we are addressing three important aspects of any computer system: Confidentiality, Integrity and Availability [3]. It is sometime easy to consider vulnerabilities as they apply to all three broad categories of system resources (Hardware, Software and Data) Fig. 1 shows the type of vulnerabilities we might find as they apply to assets of hardware, software and data.

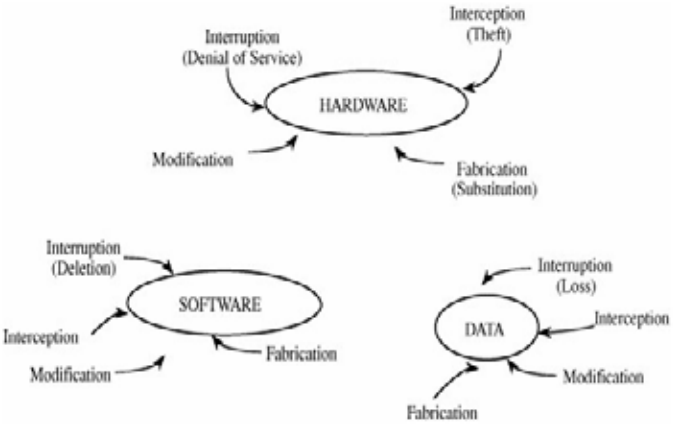

Fig. 1: Vulnerabilities of Computing Systems

A. Threats in Networks

Since the list of security attacks is long as threats aimed to compromise confidentiality, integrity, or availability, applied against data, software, and hardware by nature, accidents, non-malicious humans, and malicious attackers.

1. What makes a Network Vulnerable?

An isolated home user or a stand-alone office with a few employees is an unlikely target for many attacks. The network differs from a stand-alone environment in certain aspects like

- Anonymity.
- Many point of attacks.
- Sharing.
- Complexity of system.
- Unknown perimeter.
- Unknown path.

2. What are the Types of Network Vulnerabilities?

The number of ways your network can be attacked from inside and what you can do to ensure your business [3].

- Precursors to attack.
- Authentication failures.
- Programming flaws.
- Confidentiality.
- Integrity.
- Availability.
- Eavesdropping and Wiretapping.
- Website Vulnerabilities.

Encryption is probably the most important and versatile tool that a network security expert has. Encryption is powerful for providing privacy, authenticity, integrity, and limited access to data. Because networks involve greater risks, they are often secured with encryption, perhaps in combination with other controls. The main three reasons to use encryption to counter the network security threats are as under:

- Encryption is not a panacea or silver bullet.
- Encryption protects only what is encrypted.
- Encryption is no more secure than its key management.

B. Security Requirements

The heart of the security plan is its set of security requirements that includes functional, performance and demands placed on a system to ensure a desired level of security which are derived from organizational needs. Sometimes these needs include the need to conform to specific security requirements imposed from outside, such as by a government agency or a commercial standard.

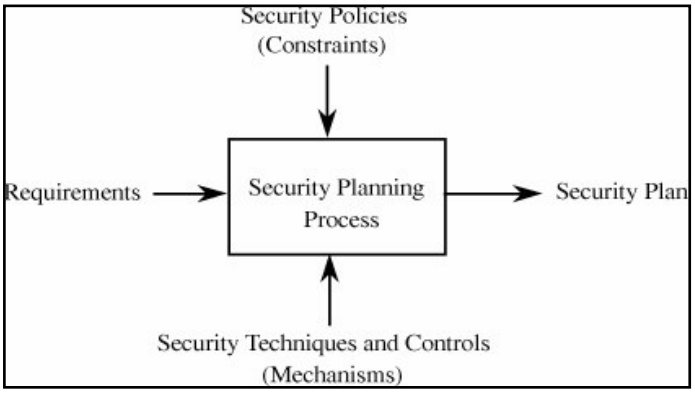

Fig. 2: Inputs to Security Plans

We must distinguish the requirements from constraints and controls. From fig. 2, a constraint is an aspect of the security policy that constrains, circumscribes, or directs the implementation of the requirements. And control is an action, devise, procedure, or technique that removes or reduces vulnerability. As listed by the U.S. Department of Defense's TCSEC, the six requirements are:

- Security policy.
- Identification.
- Marking.
- Accountability.
- Assurance.
- Continuous protection.

II. Network Security Controls

A. Security Controls

The network security based on encryption is for confidentiality, integrity, access control and overlapping controls for the network defense in depth. These strategies are also useful in protecting networks. Thus security threat analysis involves three important steps: First, we scrutinize all the parts of a system so that we know what each part does and how it interacts with other parts. Next, we consider possible damage to confidentiality, integrity, and availability. Finally, we hypothesize the kinds of attacks that could cause this damage.

B. Encryption and Authentication

Encryption is most commonly used for secrecy [1]; we usually encrypt something so that its contents or even its existence are unknown to all but only to a privileged audience. In some cases, however, integrity is a more important concern than secrecy. For example, in a document retrieval system containing legal records, it may be important to know that the copy retrieved is exactly what was stored. Likewise, in a secure communications system, the need for the correct transmission of messages may override secrecy concerns. In most files, the elements or components of the file are not bound together in any way. That is, each byte or bit or character is independent of every other one in the file. This lack of binding means that changing one value affects the integrity of the file, but that one change can easily go undetected. What we would like to do is somehow put a seal or shield around the file so that we can detect when the seal has been broken and thus know that something has been changed. This notion is similar to the use of wax seals on letters in medieval days; if the wax was broken, the recipient would know that someone had broken the seal and read the message inside. In the same way, cryptography can be used to seal a file, encasing it so that any change becomes apparent. One technique for providing the seal is to compute a cryptographic function, sometimes called a hash or checksum or message digest of the file. The most widely used cryptographic hash functions are MD4, MD5 (where MD stands for Message Digest), and SHA/SHS (Secure Hash Algorithm or Standard).

A digital signature is a protocol that produces the same effect as a real signature [3]. It is a mark that only the sender can make, but other people can easily recognize as belonging to the sender. Just like a real signature, a digital signature is used to confirm agreement to a message. A digital signature must meet two primary conditions as shown in fig. 3.

- It must be unforgeable. If person P signs message M with signature S (P, M), it is impossible for anyone else to produce the pair [M, S (P, M)].
- It must be authentic. If a person R receives the pair [M, S (P, M)] purportedly from P, R can check that the signature is really from P. Only P could have created this signature, and the signature is firmly attached to M.

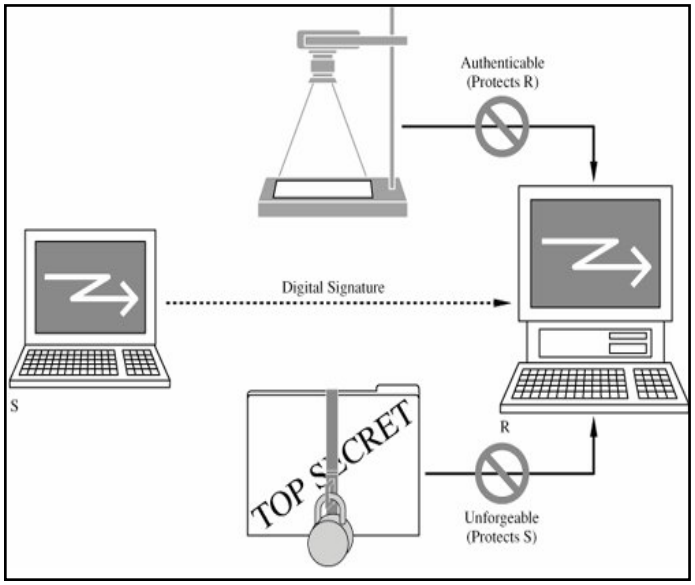

Fig. 3: Requirement of Digital Signature.

III. Proposed Methodology

A. BLOWFISH Algorithm

Blowfish is a new secret key block cipher; it has a feistel network, iterating a simple encryption function for 16 times [6-9]. The block size is fixed 64 bits, and the key size scalable from 32 – 448 bits. There is a complex initialization phase required before any encryption can take place; the actual encryption of data is very efficient on large microprocessors. It uses simple operations that are efficient on microprocessors: e.g. exclusive-OR, addition, table lookup, and modular-multiplication. Blowfish algorithm is suitable for applications where the key does not change often, like a communication link or an automatic file encryptor [4]. It is significantly faster than DES when implemented on 32-bit microprocessors with large data caches. Blowfish is a variable

–length key where each round in the feistel network consists of a key dependent permutation and a key-dependent & data-dependent substitution [5]. Blowfish uses a large number of subkeys [10]. These keys must be precomputed before any data encryption or decryption.

1. The P-array consists of 18 32-bit subkeys [9].

P1, P2, P3, P4, P5, …, P18.

2. There are four 32-bit S-boxes with 256 entries each [9].

S1,0, S1,1, …, S1,255;
S2,0, S2,1, …, S2,255;
S3,0, S3,1, …, S3,255;
S4,0, S4,1, …, S4,255;

A Feistel network is a general method of transforming any function (usually called an f function) into a permutation [9]. The working of a Feistel Network is given below in fig 4:

- Split each block into halves
- Right half becomes new left half
- New right half is the final result when the left half is XOR with the result obtained on applying f to the right half and the key.
- Note that the previous rounds can be derived even if the function f is not invertible.

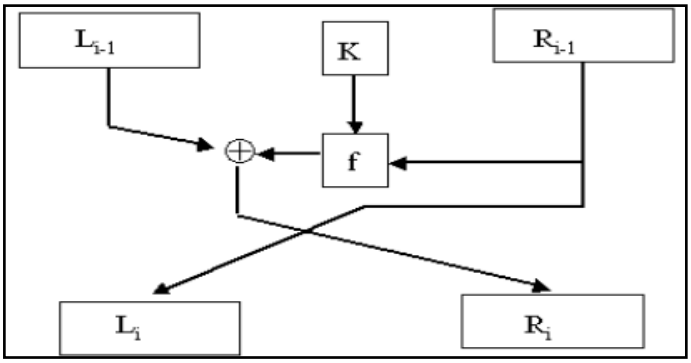

Fig. 4: Feistel Function

B. Modified f – Function

Blowfish f function is very important feistel function in this algorithm [7]; we have used the modified f function in this paper to give more security and reduce the execution time of an algorithm as compared to the existing f function shown in fig 5 which is defined as follows:

$f(x) = ((S1 + S2 \bmod 2^{32}) \text{ XOR } S3) + S4 \bmod 2^{32}$

Instead of using the existing f function we have used the modified f function which executes as follows:

$f(x) = ((S1 \text{ XOR } S2 \bmod 2^{32}) + (S3 \text{ XOR } S4 \bmod 2^{32}))$

So this modification leads to the simultaneous execution of two XOR operations. The original f function executes in sequential order and it requires 32 Addition operations along with 16 XOR operations. But in case of modified f function shown in fig 6, which requires the same 48 gate operations (32-XOR, 16-addition) but time taken to execute these 48 operations is computed to be less when compared to the existing f function, because of multithreading. We executed 32 XOR operations in parallel order using threads and hence the time taken to complete 16 gate operations will be equal to the time taken to complete 32 XOR operations since we are running it in parallel environment.

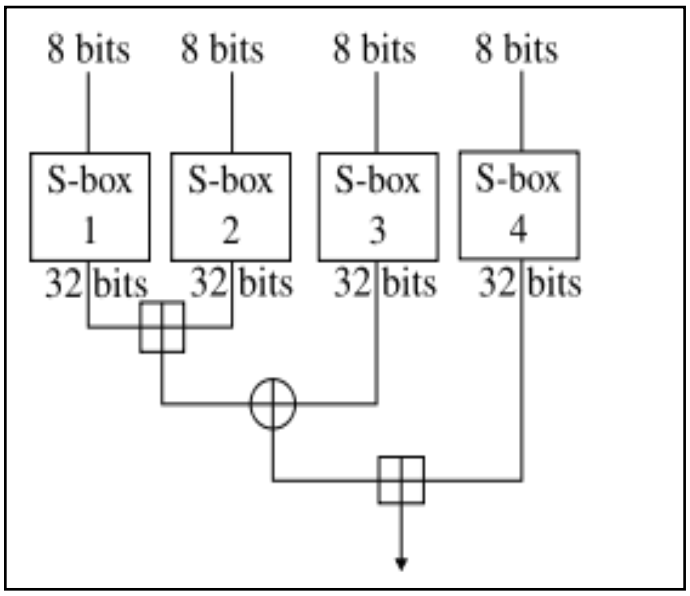

Fig. 5: Existing f Function

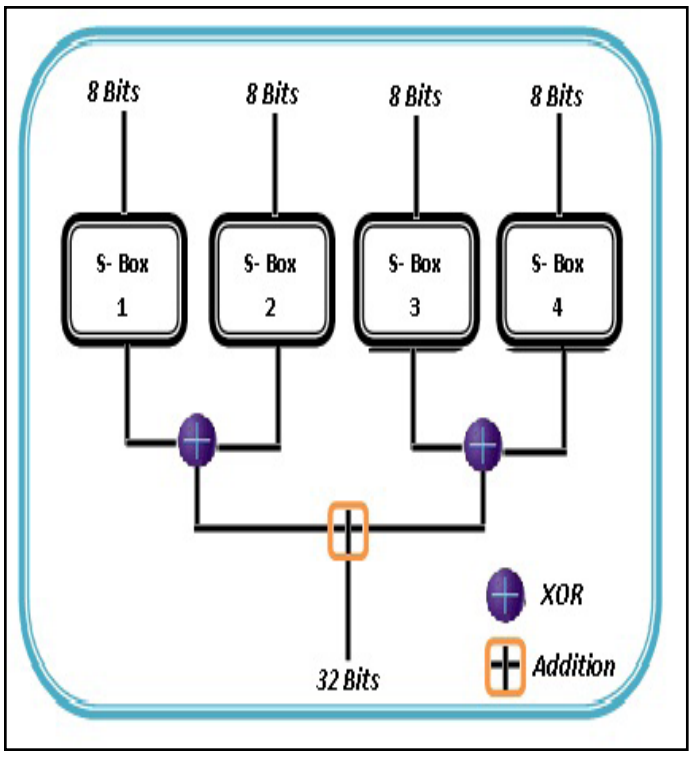

Fig. 6: Modified f Function

C. Kerberos

Kerberos is a system that supports authentication in distributed systems [2]; originally it is designed to work with secret key encryption. Kerberos in its latest version V5 uses public key technology to support key exchange. It was designed at Massachusetts Institute of Technology. Kerberos is used for authentication between intelligent processes, such as client-to-server tasks, or a user's workstation to other hosts. Kerberos is based on the idea that a central server provides authenticated tokens, called tickets to requesting applications [3]. A ticket is an unforgeable, nonreplayable, authenticated object. That is, it is an encrypted data structure naming a user and a service, which the user is allowed to obtain. The first step in using Kerberos is to establish a session with the Kerberos server [3]; as shown in Fig. 7. A user's workstation sends the user's identity to the Kerberos server when a user logs in. The Kerberos server verifies that the

user is authorized. The Kerberos server sends two messages shown in fig 7:

- To the user's workstation, a session key SG for use in communication with the ticket-granting server (G) and a ticket TG for the ticket-granting server; SG is encrypted under the user's password: E(SG + TG, pw)
- To the ticket-granting server, a copy of the session key SG and the identity of the user.

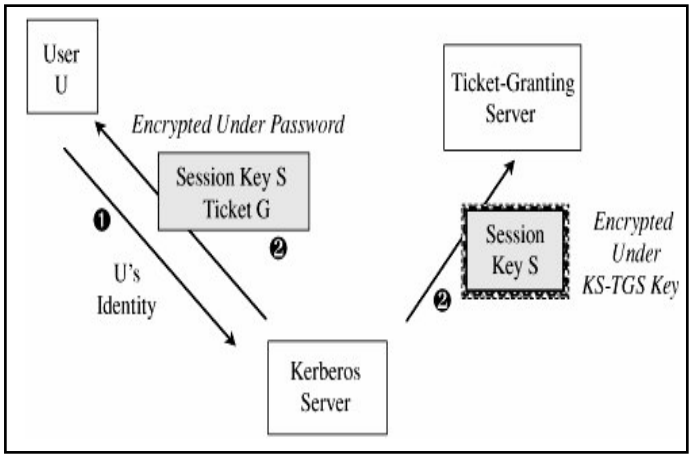

Fig. 7: Initiating KERBEROS Session

The user U requests a ticket to access file F from the ticket-granting server. After the ticket-granting server verifies U's access permission shown in fig 8, it returns a ticket and a session key. The ticket contains U's authenticated identity (in the ticket U obtained from the Kerberos server), an identification of F (the file to be accessed), the access rights (for example, to read), and a session key SF for the file server to use while communicating this file to U, with expiration date for the ticket.

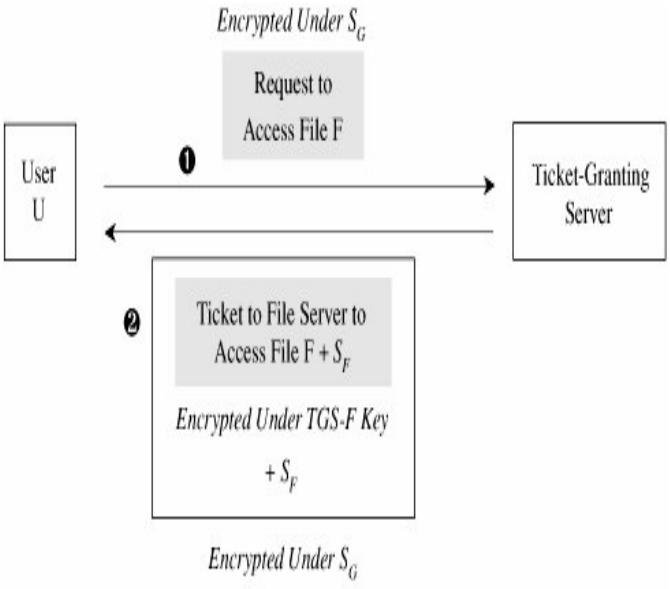

Fig. 8: Obtaining a Ticket to Access a File

D. Modified Kerberos Realm

We proposed to make use of RFC 5247 for Extensible Authentication Protocol (EAP) Key management framework in our scheme shown in fig 9; In addition to EAP methods which supports key derivation and mutual authentication known as peer-id, server-id, and session-id [8].

1. Peer-Id

An EAP method that generates keys authenticates one or more method-specific peer identities, those identities are exported by the method as the Peer-Id(s). It is possible for more than one Peer-Id to be exported by an EAP method.

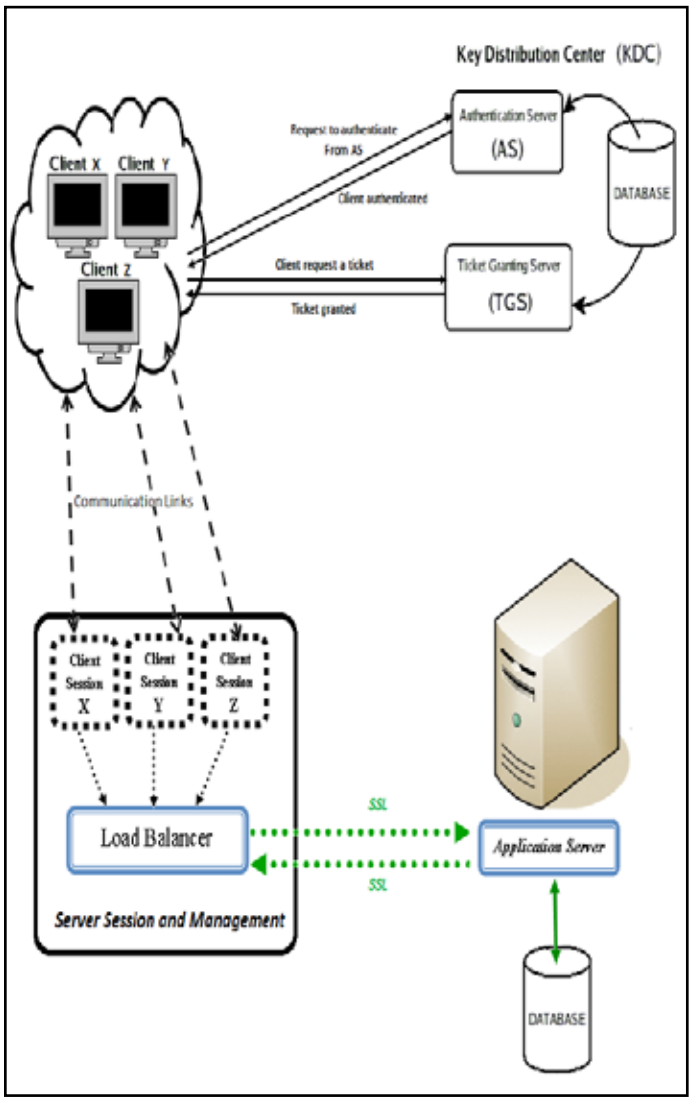

Fig. 9: Modified KERBEROS Authentication Realm

2. Server-Id

An EAP method that generates keys authenticates one or more method-specific server identities, the identities that are exported by the method as the Server-Id(s). It is possible for more than one Server-Id to be exported by an EAP method. If the EAP method does not generate keying material, the Server-Id is null string.

3. Session-Id

The Session-Id uniquely identifies an EAP session between an EAP peer (as identified by the Peer-Id) and server (as identified by the Server-Id). EAP Type Code and a temporally unique identifier are obtained from the method (known as the Method-Id).

4. Load Balancer

It is used to distribute the workload across multiple computers or a computer cluster network links, central processing units, disk drives, and other resources, to achieve optimal resource utilization, maximize throughput, minimize response time, and avoid overload shown in fig 9. Using multiple components with load balancing, instead of a single component, may increase

reliability through redundancy. Some load balancers provide a mechanism for doing something special in the event when all backend servers are unavailable. This might include forwarding to a backup load balancer, or displaying a message regarding the outage [11].

IV. Conclusion

Our effort in this paper will surely solve the problem in the open distributed environment which is unauthorized access to network resources & servers. The modified Kerberos authentication method implemented in this paper will certainly allow the centralized authentication server to perform authentication for client-server application. Kerberos entirely rely on "symmetric key" cryptography. We have also made used of the modified blowfish algorithm which is also a "symmetric key" encryption scheme. Blowfish has been extensively analysed and deemed reasonable secure by the cryptography community. It also satisfies the need of high-speed data transfer. The modified f function also enhances the performance by reducing the clock cycles and minimizing execution time to provide more security in Kerberos realm.

References

[1] W. Stallings,"Cryptography and Network Security", fourth edition, prentice hall, 2005.
[2] A. Melnikov, Ed., Isode, RFC 4752 – The Kerberos V5 ("GSSAPI") Simple Authentication and Security Layer (SASL) Mechanism, November 2006.
[3] Charles. P. Pfleeger,"Security in computing", fourth edition, prentice hall, October 2006.
[4] Bill Gatliff,"Encrypting data with the Blowfish algorithm", EETimes-India, August 2003.
[5] B. Schneier,"The Blowfish Encryption Algorithm Retrieved October 25, 2008, [Online] Available: http://www.schneier.com/blowfish.html
[6] B. Schneier,"Applied Cryptography: protocols, algorithms and source code in C", second edition, john wiley & sons, 1996.
[7] G. Manikandan, P. Rajendiran, K. Chakarapani, G. Krishnan, G. Sundarganesh,"A modified crypto scheme for enhancing data security, JATIT & LLS, 2012, Vol. 35, No. 2.
[8] B. Aboba, D. Simon,"RFC 5247 – Extensible Authentication Protocol (EAP) Key Management Framework", August 2008.
[9] B. Schneier,"Description of a new variable-length key, 64-bit block cipher (blowfish), Fast software encryption", Cambridge security workshop proceedings(December 1993), Springer-Verlag, 1994, pp. 191-204.
[10] "Blowfish Encryption Algorithm", pdf from pocket Brief, [Online] Available: http://www.pocketbrief.net/related/BlowfishEncryption.pdf
[11]. Chandra Kopparapu,"Load Balancing Servers, Firewalls & Caches", John Wiley & Sons, 25 January 2002, ISBN 0-471-41550-2.

Mr. Hemraj Shobharam Lamkuche received his B.Sc. degree in Computer Science from Dadasaheb Devidas Namdeo Bhole College, Bhusawal, Maharashtra, India in 2008, the M.Sc. degree in Computer Science from North Maharashtra University, Jalgaon, Maharashtra, India in 2010, and doing his M.Phil – Computer Science, with broad field of research in Information Security from Dr. G. R. Damodaran College of IT & Science, Coimbatore, Tamil Nadu, India. His core research interests include Cryptography, Key Management, IP Security, Network Security, Steganography, and Digital Signature. At present, He is engaged in Quantum Cryptography and enhancing the speed of symmetric key encryption algorithms.

Ms. T. Santha received her post graduation from P.S.G College of Arts & Science, Coimbatore in 1987 and M.Phil. from Bharathiar University in 1991. She is presently working as a Professor and Head, School of IT & Science, Dr. G. R. Damodaran College of Science, Coimbatore. She has more than two decades of teaching and research experience. Her research centers on Computer Applications and Optimization techniques, multimodal transportation and computer simulation, Discrete & Applied Mathematics, Information Security, Network security, Data Mining. At present, she is engaged in Transportation & Logistics Modalling.