Journal of Software Engineering and Applications, **2011, 4, 311-315**
doi:10.4236/jsea.2011.45034 Published Online May 2011 (http://www.SciRP.org/journal/jsea)

Secure File Multi Transfer Protocol Design

Murali Krishna, Pradeep Jamwal, K. S. R. Chaitanya, B. Vinod Kumar

Gayatri Vidya Parishad College of Engineering, Visakhapatnam, India.
Email: krishna.murali564@gmail.com, pradeep_jamwal@yahoo.co.in, chaitanya_p_v_k@yoo.co.in, vinukumar5b9@gmail.com

Received March 28th, 2011; revised April 22nd, 2011; accepted April 30th, 2011.

ABSTRACT

As the internet grows in popularity and therefore also in size more and more transmission takes place mainly because the technology is more readily available and applications have become more user friendly allowing entry to less sophisticated user over a broad spectrum. Most data transfer are mainly text based not secure and vulnerable to various forms of security risks. So the model that uses SSH for securing channel like intranet/internet which provides client authentication encryption and decryption with high degree of security by transferring the data in an encrypted format, up on this model enhances the efficiency of data transmission by encrypting or decrypting the data with AES in Counter Mode. AES is a symmetric key encryption standard. Moreover the permutation controlled by data can be performed at high speed in generic cpu. This scheme also expands the key space without costing more to run. And also finally through the combination of secure shell (*ssh*) *and AES* (*Counter Mode*) *not only enhances the security of communication channel. It also provides various applications like remote user creation, remote user deletion, remote command execution, remote system shutdown, remote file transfer applications in a highly secure manner.*

Keywords: *Remote SSH, AES, Remote Administration*

1. Introduction

This paper addresses the problem of providing a secure means of client to client or server to server or client to server over an insecure channel like internet. The paper aims to use the SSH and AES in Counter Mode which is the enhanced algorithm for securing the transmission channel between any two remote computers.

1.1. Secure Shell

SSH™ (or *S*ecure Shell) is a protocol which facilitates secure communication between two systems using a client/server architecture and allows users to log into server host systems remotely. Unlike other remote communication protocols, such as FTP or Telnet, SSH encrypts the login session, making it impossible for intruders to collect unencrypted passwords.

SSH is designed to replace older, less secure terminal applications used to log into remote hosts, such as telnet or rsh. A related program called scp replaces older programs designed to copy files between hosts, such as rcp. Because these older applications do not encrypt passwords transmitted between the client and the server, avoid them whenever possible. Using secure methods to log into remote systems decreases the risks for both the client system and the remote host. This increasing the remote file transfer solutions and it also increases the popularity has been fueled by the broader availability of commercially developed and supported client and server applications for windows, Unix and other platforms and by the effort of the OPENSSH [1] project to develop an open source implementation.

1.2. Features of SSH:

The SSH protocol provides the following safeguards:

- After an initial connection, the client can verify that it is connecting to the same server it had connected to previously.
- The client transmits its authentication information to the server using strong, 128-bit encryption.
- All data sent and received during a session is transferred using 128-bit encryption, making intercepted transmissions extremely difficult to decrypt and read.

The client can forward X11 [2] applications from the server. This technique, called *X*11 *forwarding*, provides a secure means to use graphical applications over a network. Because the SSH protocol encrypts everything sends and receives, it can be used to secure otherwise insecure protocols. Using a technique called *port forwarding*, an SSH server can become a conduit to securing otherwise insecure protocols, like POP, and increaseing overall system and data insecurity.

According to Shannon [3] claims that SSH has three main capabilities. Secure command shell: such as those available to Linux, UNIX, Windows or the familiar DOS prompt, provide the ability to execute programs and other commands, usually with character input and output. Port forwarding: allows TCP/IP applications data to be securely transmitted over insecure channels.

Secure file transfer: SFTP is an interactive file transfer protocol which performs all operations over the SSH transport layer and is replacement for the original SCP protocol existed in SSH. It is highly recommended that SFTP is used to perform the file transfer in preference to the legacy FTP protocol. As in the latter, authentication details are transmitted in plain text format and such may be compromised through “password sniffing” attacks. The former also uses the same port as the SSH server, eliminating the need to open another port on the firewall of the router.

According to the van Shannon [3] the SSH protocol provides four basic security benefits. Which are user authentication, data encryption, and data integrity?

Authentication: public based and host based authentication .of these, public key authentication is one of the most secure methods to authenticate using SSH. Public key authentication uses a public/private key pair, generated typically by using key generation utility.

Data Encryption: when a client establishes a connection with SSH server or independent servers they must agree with cipher they will use to encrypt and decrypt data. The server generally supports the list of ciphers it supports. And the client then selects the first cipher in its list that matches one on the server’s list. Session keys are the “shared keys” described above and are randomly generated. Both the client and server use the same key for both encryption and decryption.

Data Integrity: even with SSH encryption, the data being sent over the network could still be vulnerable to someone inserting unwanted data into the data. SSH uses HMAC algorithms to greatly improve up on SSH’s simple 32-bit CRC data integrity checking method.

SSH enabled applications are gaining popularity because of the security they supply for the task carried out over the network. Some of the popular one are putty, SSH client for windows and VNC over SSH [4]. As a security protocols’ has not been as popular as SSL [5].

1.3. The Realization of AES Algorithm

Figure 1 describes AES is a symmetric block cipher having variable key and fixed data length. The key lengths can be independently chosen as 128, 192 or 256 bits, which result in 10, 12 and 14 rounds of operation respectively. The data length is however fixed to 128 bits. The input as well as intermediate data can be considered as a matrix with four rows and four columns called state. Each element of the matrix is composed of eight bits, therefore enabling efficient implementation of AES on 8 bit platforms also. AES is mainly used to ensure secrecy in important communications, such as those of government covert operations, military leaders, and diplomats

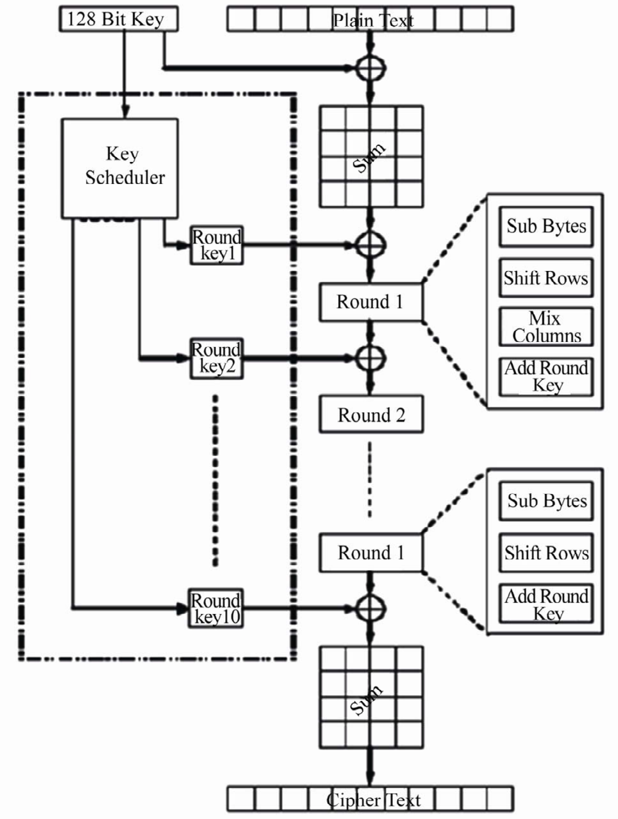


Figure 1. Rounds in AES algorithm.

SubBytes-a non-linear substitution step where each byte is replaced with another according to a lookup table.

ShiftRows-a transposition step where each row of the state is shifted cyclically a certain number of steps.

Mix Columns-a mixing operation which operates on the columns of the state, combining the four bytes in each column.

AddRoundKey-each byte of the state is combined with the round key using bitwise XOR.

AES in Counter Mode: CTR mode (CM) is also known as integer counter mode (ICM) and segmented integer counter (SIC) mode. Counter mode turns a block cipher into a stream cipher as shown in **Figure 2**. It generates the next key stream block by encrypting successive values of a “counter”. The counter can be any function which produces a sequence which is guaranteed not to repeat for a long time, although an actual counter is

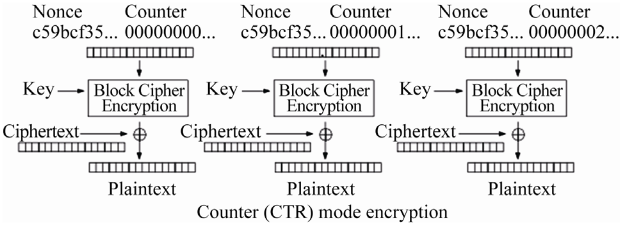


Figure 2. Counter mode encryption.

the simplest and most popular. The usage of a simple deterministic input function used to be controversial; critics argued that "deliberately exposing a cryptosystem to a known systematic input represents an unnecessary risk." By now, CTR mode is widely accepted, and problems resulting from the input function are recognized as a weakness of the underlying block cipher instead of the CTR mode. Nevertheless, there are specialized attacks like a Hardware Fault Attack that is based on the usage of a simple counter function as input. CTR mode has similar characteristics to OFB, but also allows a random access property during decryption. CTR mode is well suited to operation on a multi-proc- essor machine where blocks can be encrypted in parallel.

2. Contribution

The running time that DES took as much as irrational DES. *i.e.* the confidentiality of the key is enhanced without spending more time, secondly based on the same plain text and key, the cipher text of DES after several simulation is the same, but it is random about the DES with irrational numbers. *i.e.* the key space is expanded through increasing the randomness of sub- keys and it is combined with secure shell protocol the proposed model provides maximum amount of security to an insecure communication link between remote clients and servers. This model enhances the security of communication channel. In which some of the following aspects are unique. The system offers supreme security due to double encryption. *i.e.* once with irrational DES and once with RSA in builtness of SSH protocol. SSH is normally used to secure applications like Telnet, and FTP but in this model it is used very similar to SSL. Like SSL, it runs over TCP/IP and secures the data sent between TCP/IP or client/server applications, retaining all the security benefits of SSH. Not only that SSH also provides various administration applications like remote command execution, remote user creation or deletion or remote shutdown or reboot or file transfer application in an highly secure manner.

3. Desighn and Implementation

Figure 1 and **Figure 6** showed the arrangement of high level secure communication of clients to server or clients to clients.

3.1. SSH Server Application Implementation

The GUI is where the user interact s with the system. It receives data input from user and displays received information in both encrypted format and sent data also.

3.2. SSH Client Application Implementation

The process of setting up an SSH secured communication channel is as follows:- configure the encryption algorithms for use from client to server and from server to client—configure the hash algorithm used in both directions—use this properties (containing the configurations) object and use this as parameter for establishing a SSH connection and subsequently returning a handle on the connection by means of an object—use the toolkit's password authentication method s-instantiate the channel class, and use the channel object as a parameter for the open channel method (of SSH connection object). Through SSH connection and with irrational DES the multiple messages also can be passed to the various client to servers.

Automatic login remote host

The person under consideration is the system administrator. He has different responsibilities while working in a network. He may have a requirement to login to client systems and do the necessary modifications. For this, if he is coerced to key in the password then there is a high probability for the password to be sniffed and intruders attacking the system? To overcome these hurdles we have provided the feature, automatic login to remote host which shown in **Figure 3**.

Creating users/Deleting users

The system administrator has the responsibility of supervising the network. In this effort, he might be required users in the system. So, this feature allows him to create users which are shown in **Figure 4**.

Steps:

1) Connect to the remote host through secure shell.

2) Provide proper password through secure shell.

3) Crate the users through command "user add" by sending command in encrypted format using secure shell.

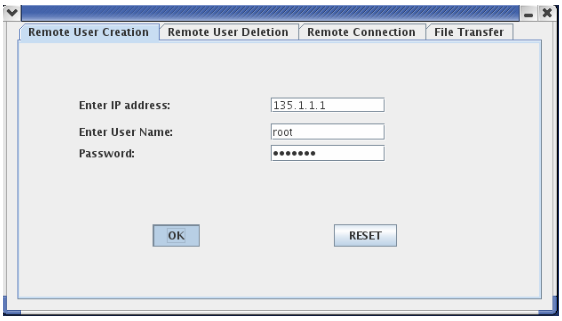
Execution of command(eg. ls, cat, finger, find)

Unix supports commands like cat, finger, find etc. each command is a file. The commands are executed to accomplish different functionalities. For example, "ls" command is used to list all the files. So, this feature allows administrator to execute the different commands which are shown in **Figure 5**. And the steps follows for remote command execution are:

Steps:

1) Connect to the remote host through secure shell.

2) Provide proper password through secure shell.

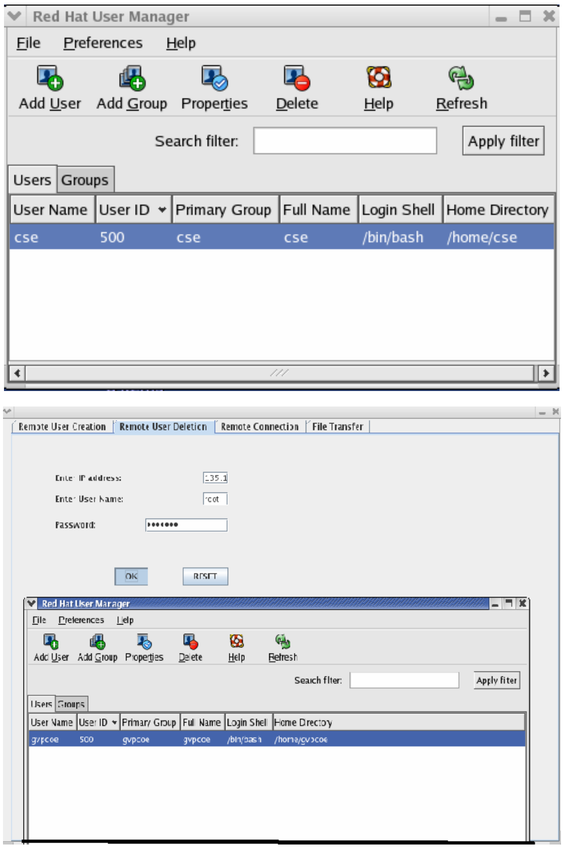


Enter IP address: 135.1.1.1

Enter User Name: root

Password: •••••••

Figure 3. SSH remote accesing.



User Name	User ID	Primary Group	Full Name	Login Shell	Home Directory
cse	500	cse	cse	/bin/bash	/home/cse

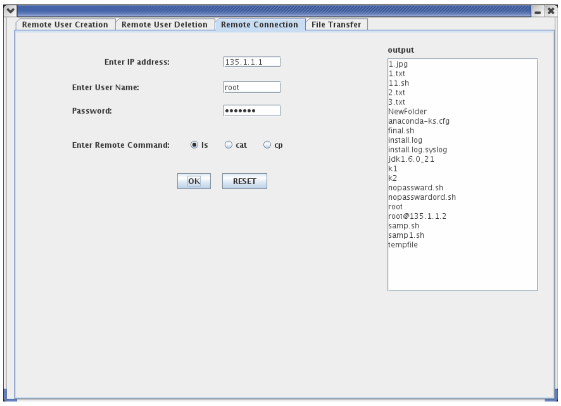
Enter IP address: 135.1

Enter User Name: root

Password: •••••••

User Name	User ID	Primary Group	Full Name	Login Shell	Home Directory
gvpcoe	500	gvpcoe	gvpcoe	/bin/bash	/home/gvpcoe

Figure 4. SSH remote user creation or deletion.



Enter IP address: 135.1.1.1

Enter User Name: root

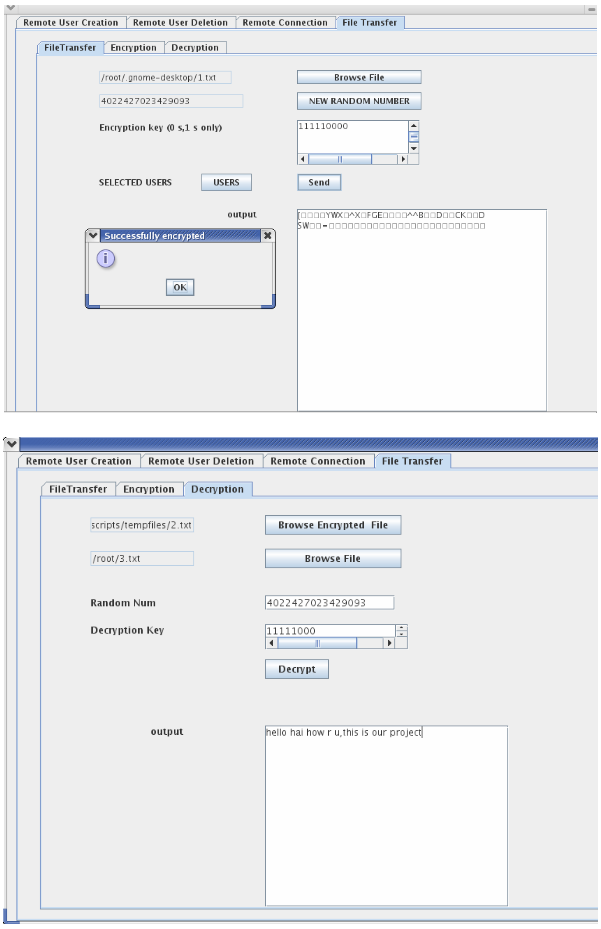
Password: •••••••

Enter Remote Command: ● ls ○ cat ○ cp

output

```
1.jpg
1.txt
11.sh
2.txt
3.txt
NewFolder
anaconda-ks.cfg
final.sh
install.log
install.log.syslog
jdk1.6.0_21
k1
k2
nopassward.sh
nopasswardord.sh
root
root@135.1.1.2
samp.sh
samp1.sh
tempfile
```

Figure 5. SSH remote command execution.



/root/.gnome-desktop/1.txt

4022427023429093

Encryption key (0 s,1 s only): 111110000

output

```
[□□□□YWX□^X□FGE□□□□^^B□□D□□CK□□D
SW□□=□□□□□□□□□□□□□□□□□□□□□□□□□□
```

Successfully encrypted

scripts/tempfiles/2.txt

/root/3.txt

Random Num: 4022427023429093

Decryption Key: 11111000

output: hello hai how r u,this is our project

Figure 6. Secure file transfer.

3) Execution of any command will be done by passing corresponding executable command.

Rebooting/shut down of remote system

In some cases, after installing the software the system may be required to reboot. In this regard, the system administrator can reboot the remote system after the software has been installed.

Steps:

1) Connect to the remote host through secure shell.

2) Provide proper password thorough secure shell.

3) Reboot or shut down of the remote system will be done through the "power off" command.

4. Tests & Results

This section provides some of the tests carried out and presents their results.

4.1. Channel Confidentiality

Verifying whether the data can be transmitted between client and server is in fact encrypted both with AES with counter mode and SSH protocol which protecting the system against passive attacks which is shown in:

Data: Hai how ru This is our project

SSH with AES with Counter Mode:

Encrypted data:

P└ PDX‼ ABF¶ F[F├ Z┤ ↑ V¶ FC¶ AXXV−⊥ > l

Encrypted data SSH with AES in Counter Mode.

4.2. SSH with AES in Counter Mode Time Comparisons

Table 1 describes the comparison of encryption or decryption time periods with existing DES and SSH with irrational DES.

EAESt—encryption time period for existing AES.

EAESCTRt—encryption time period for SSH with AES in Counter Mode.

DAESt—decryption time period for existing AES.

DAESCTRt—decryption time period for SSH with AES in Counter Mode.

4.3. SSH with AES in Counter Mode Cipher Text Comparisons

Table 2 describes about the cipher text generation for existing AES and SSH with AES with Counter Mode in which the cipher text for the same plain text is always different with existing DES whereas differ in SSH with AES in Counter Mode which causes a high security of protection against the attacks for the message.

Table 1. Time comparisons for AES vs AES with counter mode.

Data Size	Eaest	Eaesctrt	Daest	Daesctrt
1 KB	411 msec	413 msec	407 ms	409 ms
512 B	**298 msec**	**299 msec**	**276 ms**	**277 ms**
256 B	159.4 ms	160 msec	147.4 ms	148 ms
128 B	**84.7 ms**	**85 ms**	**85.7 ms**	**86 ms**

Table 2. Cipher text comparisons for AES SSH AESCTR.

Paes	Caes	Paesctr	Caesctr
Hai how ru This is our project	Cbcc883cd0d 3be-aafo66259c76 328078b	Hai how r u This is our project	P└ PDX‼ AB F¶ F[F├ Z┤ ↑ V¶ FC¶ AXX V−⊥ >l

CAES—cipher text for existing AES.

CAESCTR—cipher text for SSH with AES in Counter Mode.

PAES—plain text for existing AES.

PAESCTR—plain text for SSH with AES in Counter Mode.

5. Conclusions

The results presented in Section 4 are extremely good. They clearly showed that: the data being transmitted between client and server is in fact encrypted, protecting the system against passive attacks. And not only that the running time that existing AES took as much as that of SSH with Counter Mode in AES. And the confidentiality of the key is enhanced without spending more time. And the based on the same plain text and the key the cipher text of AES after several simulation is not same but it is random about the SSH with Counter Mode in AES.

REFERENCES

[1] Open SSH, “The Open SSH Project [Online],” 2002. http://www.openssh.org

[2] A. J. Menezes, “Elliptic Curve Public Key Cryptosystems,” 1st Edition, Kluwer Academic Publishers, Dordrecht, 1993.

[3] C. E. Shannon, “Communication Theory of Secrecy Systems,” *Bell Systems Technical Journal*, Vol. 28, No. 4, 1949, pp. 656-715.

[4] SSH Tools, “Open Source SSH Toolkit for Java [Online],” 2003. http://www.sshtools.com.

[5] J. Daemen and V. Rijmen, “AES—the Advanced Encryption Standard,” Springer, Berlin, 2002.