



Asymmetric Key Based Encryption and Authentication System for File Sharing

Kuntal Shah¹, Prof. Chandresh Parekh², Asst.Prof. Bhadresh Gohil³

¹M.Tech Cyber Security, Raksha Shakti University, Ahmedabad

²Department of Telecommunication, Raksha Shakti University, Ahmedabad

³GTU P.G.School, Gandhinagar

Abstract — Most of the files sharing systems are using symmetric key based authentication system where password is provided by the separate communication channel but this is cumbersome and some time lead to guessing the password and opening the file by unintended person. There is a need of technique in which the authentication and the file encryption can be possible effectively. In this paper we have given a solution based on asymmetric encryption based file sharing and authentication system for the users where keys can be generated through the system software and distributed to users by the administrator. The solution is provided with the use of various open source operating systems and file sharing systems.

Keywords- Asymmetric cryptography; file sharing; open Source; Software based key generation;

I. INTRODUCTION

With the help of the internet it is easy to share the files but as every technology has its own pros and cons the file sharing with internet has also its vulnerability. For any security system the main need is Privacy, Authenticity, Integrity and Non-repudiation (P.A.I.N).[1]

Privacy relates that data or file being transferred cannot be accessed by the unauthorized parties. Authenticity relates that the authenticated person only should be able to access the data. Integrity relates to the data is not modified by the unauthorized parties when data is in transit. Non-repudiation means the sender should not able to deny that message is sent by them.[1][2]

Traditionally files were sent without encryption but due to security concern the encryption of file came in picture. To secure the file sharing system there were many cryptographic algorithms had been developed.

There are mainly two types of cryptographic systems are used based on their key distribution system. The mainly used encryption system is **symmetric key** based encryption system where the sender and receiver are using the same key to encrypt and decrypt the file and communication. The main problem is that those who have the key can decrypt the message.[2] Main problem of this type of cryptographic algorithm is sharing of key. So for secure transmission of the key one extra communication channel is used to share the key. But the main problem is it is easy to crack this password with brute force attack and man in the middle attack. [3] The main advantage of the symmetric key cryptography is it gives better privacy but it is not capable to solve the issues like authenticity, integrity and non-repudiation. [4] So there is a need of new type of authentication system that can provide authentication that the client or user is authorized, integrity of file is maintained and the non-repudiation. The perfect solution for P.A.I.N is Asymmetric key based encryption system with digital certificate and digital signature. [5] The concept of the digital certificate is widely used by server to authenticate client. The mainly used this type of scheme is known as SSL or TLS services. The main problem in this type of system is the generation and the distribution of the key.[6] Many organizations have provided hardware based solution of it but it is not scalable. In this paper we have given the solution for the file transmitting service which is based on asymmetric key cryptography and is scalable for the large user and completely software based key generation. This system will transfer the file in encrypted form so that the intruder cannot be able access it.[7][8]

II. PROPOSED ALGORITHM FOR AUTHENTICATION

For the sharing of file in a network with security we have proposed a system with more secured system and is developed by open source tools. It is easy to implement.

My research is based on four steps:

- Establish a Browser based File Sharing System
- Create an Authentication system based on Asymmetric key
- Redirect all traffic to https from http
- Restrict attacks on file sharing system (Make secured file sharing system)

In this project we have taken “Owncloud” which is an open source framework for the access & sharing of files, calendar, contacts, mail etc. Now we have installed it in cent os 7 which is a open source operating system from redhat. [9][10]

In owncloud there is a database of USERS, Groups and password which can be managed by network administrator. As shown in the figure when client want to access owncloud server than he/she will make a request to the server with browser, if the request is HTTP than it will be redirected to HTTPS. For that we have to create a ssl certificate and give the path of that certificate to the configuration file of the web server.

Clients have to enter username and password. System will match that with username and hash of password stored in server. If both match than User will grant access but if the attempt is failed than system will count continuous failed attempts. If the total failed attempt is 4 (Here I have taken 4 but you can set according to your need) the authentication system will block the ip for particular time (Period can be set according to your need). [11]

The complete procedure is shown in the flow chart.

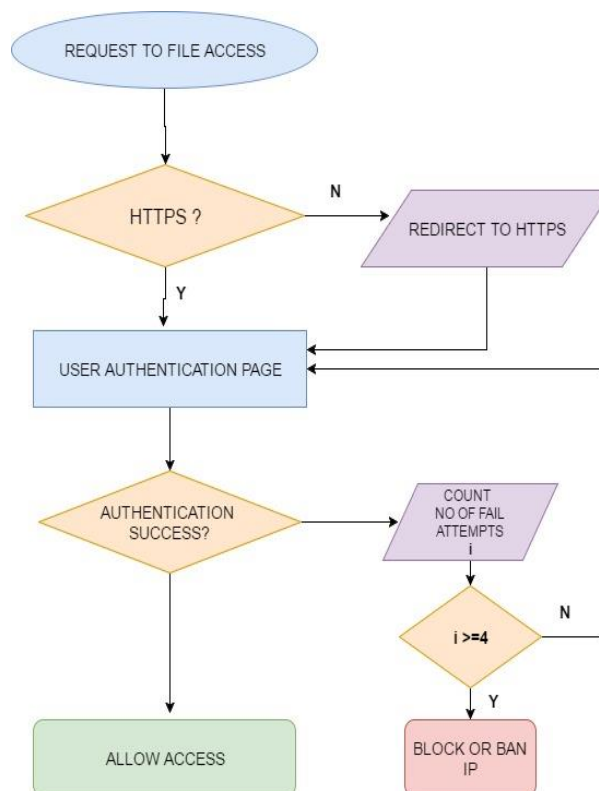
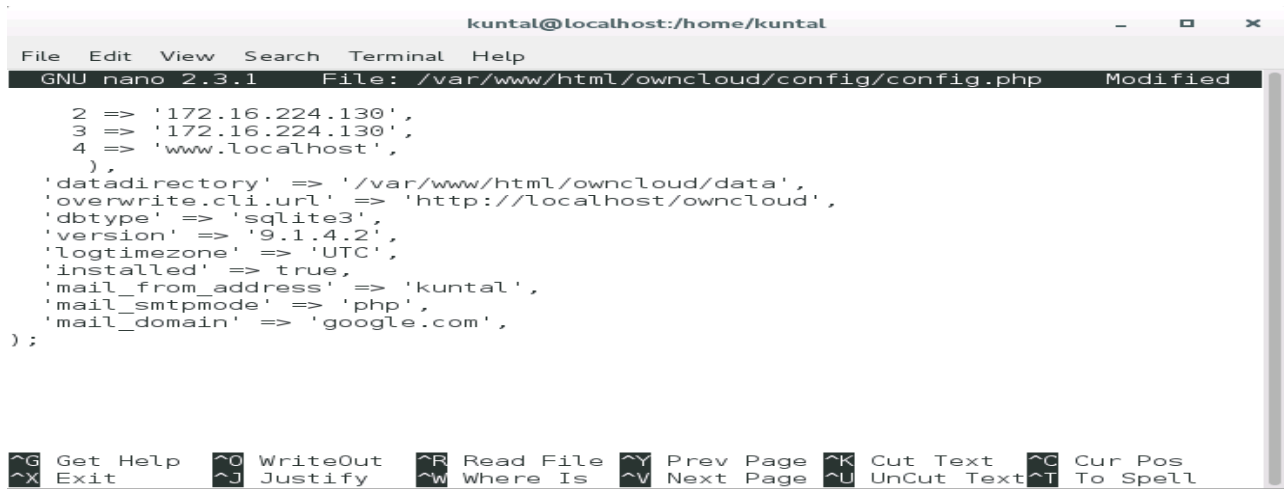


Fig.1 Flow Chart

For the security of the owncloud we have created a jail in fail2ban which will analyze the logs and ban the client if it will perform malicious activity.[12] For log generation modify the source code of the owncloud as given in figure:

```
kuntal@localhost:/home/kuntal
GNU nano 2.3.1 File: /var/www/html/owncloud/config/config.php Modified
<?php
$CONFIG = array (
  'updatechecker' => false,
  'instanceid' => 'oc36y4ng4ibf',
  'passwordsalt' => 'Pw5EPAz9AiLM6RgW5MjkSTPSfAHNfL',
  'secret' => 'PDJQTWb3mXj+howS9Ljl fKq64rtZ5D4yLAY1be7A7JwjaVv4',
  'trusted_domains' =>
    array (
      0 => 'localhost',
      1 => '172.16.224.130',
      2 => '172.16.224.130',
      3 => '172.16.224.130',
      4 => 'www.localhost',
    ),
  'datadirectory' => '/var/www/html/owncloud/data',
  'overwrite.cli.url' => 'http://localhost/owncloud',
  'dbtype' => 'sqlite3',
  'version' => '9.1.4.2',
  'logtimezone' => 'UTC',
);
```

Fig.2 Source code modification of owncloud



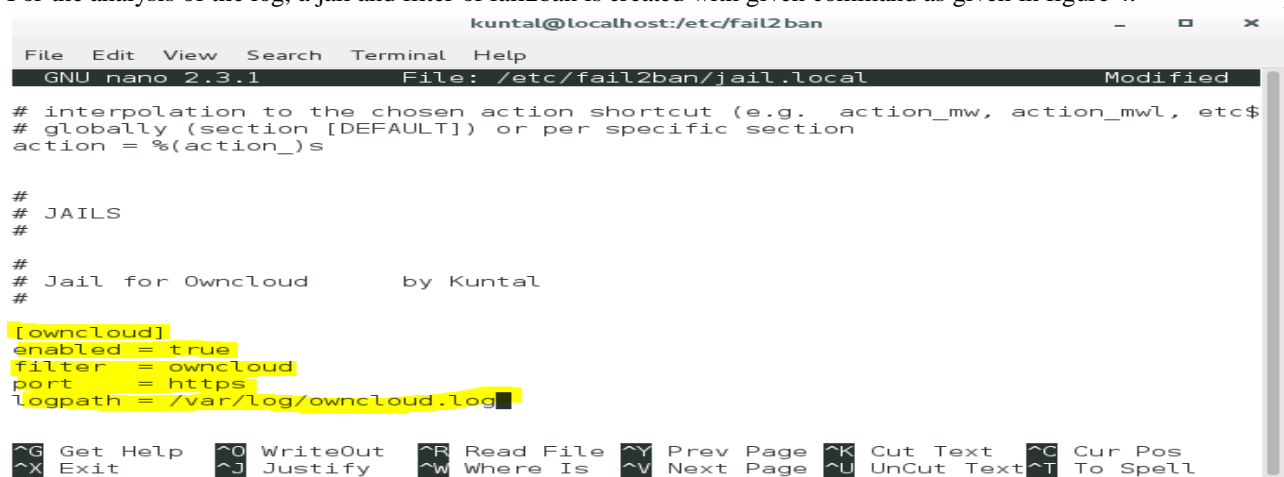
```
kuntal@localhost:/home/kuntal
File Edit View Search Terminal Help
GNU nano 2.3.1 File: /var/www/html/owncloud/config/config.php Modified

2 => '172.16.224.130',
3 => '172.16.224.130',
4 => 'www.localhost',
),
'datadirectory' => '/var/www/html/owncloud/data',
'overwrite.cli.url' => 'http://localhost/owncloud',
'dbtype' => 'sqlite3',
'version' => '9.1.4.2',
'logtimezone' => 'UTC',
'installed' => true,
'mail_from_address' => 'kuntal',
'mail_smtpmode' => 'php',
'mail_domain' => 'google.com',
);

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Fig 3: Source code modification of owncloud

For the analysis of the log, a jail and filter of fail2ban is created with given command as given in figure 4.



```
kuntal@localhost:/etc/fail2ban
File Edit View Search Terminal Help
GNU nano 2.3.1 File: /etc/fail2ban/jail.local Modified

# interpolation to the chosen action shortcut (e.g. action_mw, action_mwl, etc$
# globally (section [DEFAULT]) or per specific section
action = %(action_)s

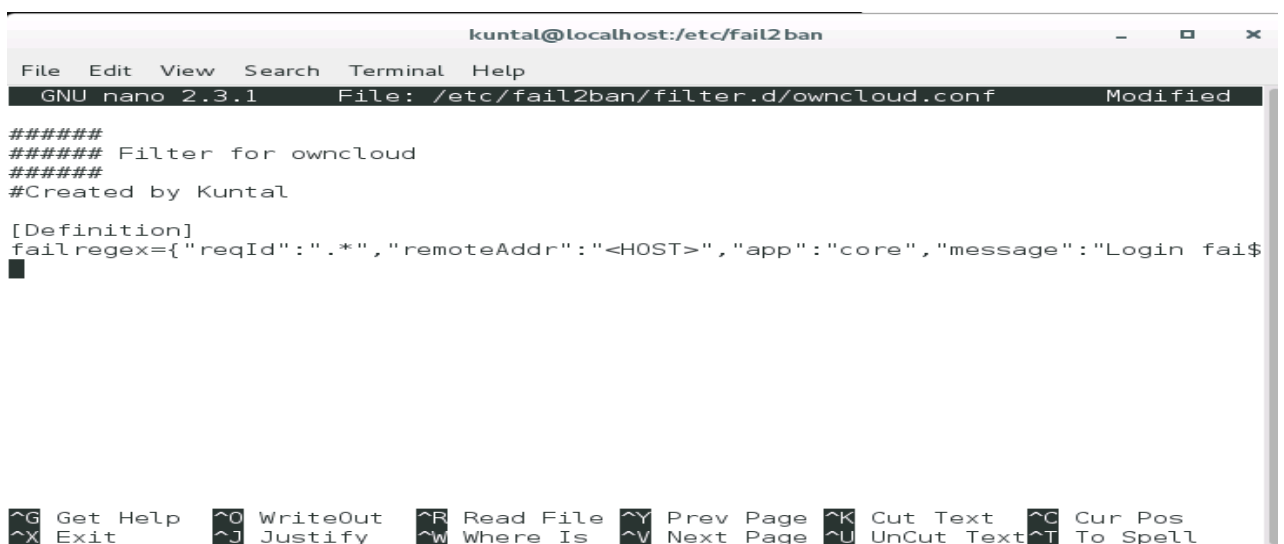
#
# JAILS
#
#
# Jail for Owncloud by Kuntal
#
[owncloud]
enabled = true
filter = owncloud
port = https
logpath = /var/log/owncloud.log

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Fig.4 Jail modification

For the generation of the filter add the following file in the filter directory of the fail2ban:

```
[Definition]
failregex={"reqId": ".*", "remoteAddr": "<HOST>", "app": "core", "message": "Login failed: .*", "level": 2, "time": ".*"}
```



```
kuntal@localhost:/etc/fail2ban
File Edit View Search Terminal Help
GNU nano 2.3.1 File: /etc/fail2ban/filter.d/owncloud.conf Modified

#####
##### Filter for owncloud
#####
#Created by Kuntal

[Definition]
failregex={"reqId": ".*", "remoteAddr": "<HOST>", "app": "core", "message": "Login fai$

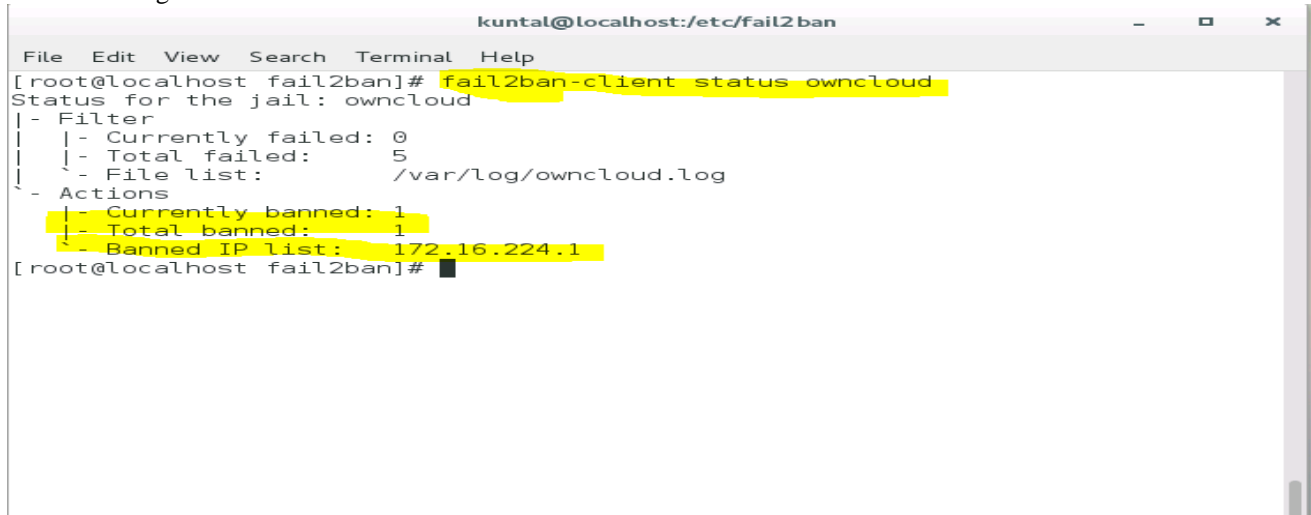
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Fig.5 Generation of filter in fail2ban

Depending on the number of failed attempts the user will be banned for predefined time.

IV. EVALUATION AND ANALYSIS

By modifying source code of the web server all http traffic can be redirected to the https. It creates encrypted channel which will secure the data in transmission. Asymmetric key based authentication system will provide the services to the authenticated user only. If any attacker tries to guessing the password he will be blocked for some predefined time period as shown in fig 6:



```
kuntal@localhost:/etc/fail2ban
File Edit View Search Terminal Help
[root@localhost fail2ban]# fail2ban-client status owncloud
Status for the jail: owncloud
|- Filter
| |- Currently failed: 0
| |- Total failed: 5
| |- File list: /var/log/owncloud.log
|- Actions
| |- Currently banned: 1
| |- Total banned: 1
| |- Banned IP list: 172.16.224.1
[root@localhost fail2ban]#
```

Fig.6: Report of blocked IP

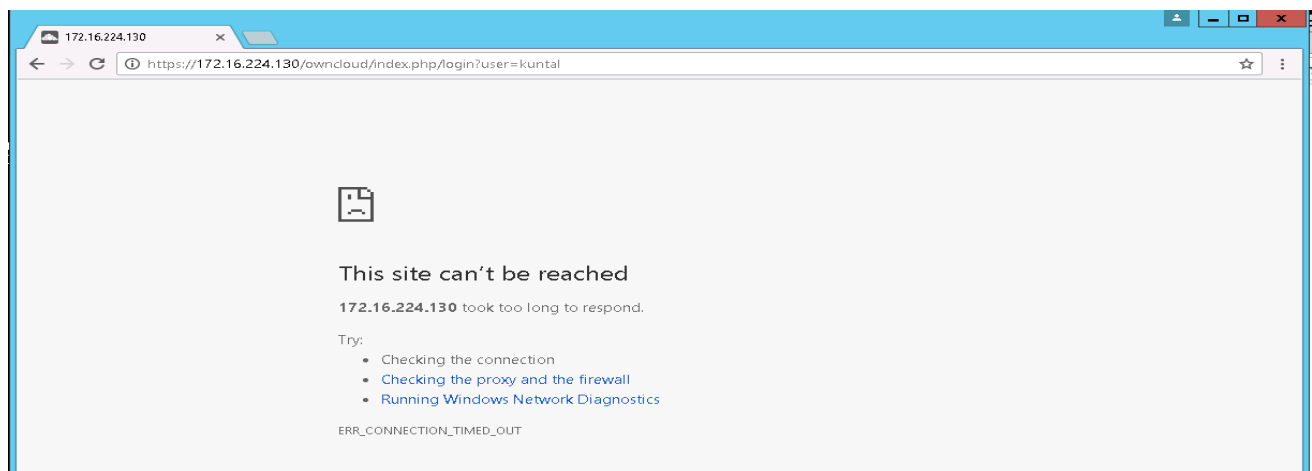


Fig.7 Error shown in the attacker's machine

There some predefined modules in the fail2ban which can prevent the web server from different type of attacks like Denial of Service, Bad bots, no script and other.

V. CONCLUSION AND FUTURE WORK

Files sharing systems that are using symmetric key based authentication system where password is provided by the separate communication channel but this good for the privacy but not provides authenticity, integrity and nonrepudiation. This paper provides authentication system which is based on asymmetric key cryptography and software based certificate generation technique which is easy to implement and more secured. Encrypting the channel will provide protection from the unauthorized discloser of the file. With the use of fail2ban the log analysis is easy and the attacks like brute-force and dos can be prevented.

REFERENCES

[1] Singhal, Nidhi and Raina, J P S. "Comparative Analysis of AES and RC4 Algorithms for Better Utilization", International Journal of Computer Trends and Technology, ISSN: 2231-280, July to Aug Issue 2011, pp. 177-181.

- [2] Singh, S Preet and Maini, Raman. "Comparison of Data Encryption Algorithms", International Journal of Computer Science and Communication, vol. 2, No. 1, January-June 2011, pp. 125-127.
- [3] Elminaam, D S Abd; Kader H M Abdual and Hadhoud, M Mohamed. "Evaluating the Performance of Symmetric Encryption Algorithms", International Journal of Network Security, Vol. 10, No. 3, pp. 216-222, May 2010.
- [4] N. Gura, A. Patel, A. Wander, H. Eberle, S. Chang Shantz, "Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs", CHES, August 2004.
- [5] A. Khalique, K. Singh, S. Sood, "A Password-Authenticated Key Agreement Scheme Based on ECC Using Smart Cards", *International Journal of Computer Applications*, vol. 2, no. 3, pp. 26-30, 2010.
- [6] R. L. Rivest, A. Shamir, L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", *Communications of the ACM Magazine*, vol. 21, no. 2, pp. 120-126, 1978.
- [7] W. Stallings, "Cryptography and Network Security Principle and Practice" in , Prentice Hall, 2011.
- [8] M. Strebe, "Network Security Foundations" in SYBEX Inc., San Francisco, London:1151 Marina Village Parkway, Alameda, CA 94501, 2004.
- [9] <https://www.centos.org/>
- [10] <https://owncloud.org/>
- [11] Fail2ban, <http://www.fail2ban.org>
- [12] Justin M. Beaver, Christopher T. Symons, Robert E. Gillen, "A Learning System for Discriminating Variants of Malicious Network Traffic", *8th Annual Cyber Security and Information Intelligence Research Workshop*, October 30-November 2, 2012.
- [13] J. Owens and J. Matthews, A Study of Passwords and Methods Used in Brute-Force SSH Attacks, *USENIX Workshop on Large Scale Exploits and Emergent Threats (LEET)*, 2008.
- [14] DenyHosts, <http://denyhosts.sourceforge.net/>