An efficient and secure user authentication and communication scheme based on blowfish and blake2b algorithm using Cloud computing

Yogendra Mohan and Thokchom Anandkumar Singh

December 31, 2019

An efficient and secure user authentication and communication scheme based on cloud computing

Yogendra Mohan[1] and Thokchom Anandkumar Singh[2]

[1] NERIST University, Nirjuli, Itanagar, Arunachal Pradesh, India
[2] NERIST University, Nirjuli, Itanagar, Arunachal Pradesh, India
[1]yogendra.mohan@gmail.com, [2]thokchomanandkumar@gmail.com

Abstract

Cloud computing is a current trend of computing that provides a lot of services to computer user, government agencies and business. Cloud computing is the delivery of on-demand computing services and enable on-demand access to shared pool of resources over the internet. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third party data centers. Security of data is one of the major issue in cloud computing. In Cloud computing data security can be improved by the Encryption & Decryption, Message authentication code, and Hash function. The proposed model included the combination of a symmetric algorithm and a hashing algorithm to achieve confidentiality and data integrity.

Keywords: Cloud Computing, Authentication, Authorization, Data integrity, Confidentiality.

1. INTRODUCTION

Cloud computing is an important topic in the field of Information technology which provides IT services to many users over the network. Cloud computing include services such as application, infrastructure, storage etc. The clients or organization stores and manage their file in a data centres provided by third party. Cloud model enables users to use the resources provided by the IT over the internet instead of using end user application or software and limited memory capacity in their personal computer network. In cloud computing security is the main issue when sending and storing the information at any stage. Securing the data for cloud computing includes access, management, authentication, data integrity, data confidentiality and key sharing.

1.1. Cloud Service model:

a) Software as a Service (SaaS): It allow the capability of consumer to use the services such as application available in the cloud. The application available in the cloud can be access by various users software or application that are installed in their devices .This services are not manage or control by the clients.

b) Platform as a Service (PaaS): It allows users the capability to create application using the development tools, application programming interfaces etc. The major service offered by cloud services provider includes data storage, operating system etc.

c) Infrastructure as a Service (IaaS): This provides users the capability to provision services such as
data storage, servers and networking components. It may also include software such as Operating system. This allows the client to avoid the difficulty and problem faces while setting up their own infrastructure such as server, data storage, and other networking components.

1.2. Four Type of Cloud Deployment Model:

a) Public Cloud: The cloud infrastructure such as application, server, storage are deploy for use by everyone. The cloud infrastructure may be managed and control by an individual, organization, university etc.

b) Private cloud: The cloud infrastructure such as application, server, storage database etc. are deploy for used by a single organization which may include many clients . This type of cloud deployment model is maintained and control by the Business Company, association, a third party services provider etc.

c) Community cloud: The services provided by the cloud infrastructure are deployed for a particular community which consist of many customers from different organization that share a common interest, goal or policies etc. This type of cloud deployment model may be manage or control by a third party, an organization or group of organization or coalition of organization and third party.

d) Hybrid cloud: It is form by the joining of two or more public and private cloud.

1.3. Why Cryptography in Cloud Computing:

In cloud computing all the users computing capabilities and resources are shifted to the cloud service provider. So one effective means for achieving security and preventing unauthorized access of the data in cloud computing system is to encrypted the user file and then upload it to the cloud storage

2. LITERATURE REVIEW

Prashant et al.[1] implement a model to provided data security using diginal signature, Diffie Hellman and AES Algorithm. The digital signature provided authentication for the client and Diffie Hellman for sharing key between client and server. The security of data stored in the cloud database is maintained by encrypting the data using Advance Encryption Standard (AES) Algorithm.

Priyanka et al.[2] provided a solution for maintaining data security and data integrity. Data security is maintained using RSA Partial homomorphic algorithm. The data integrity of the encrypted data is provided at the cloud server by calculating it hash value using MD5 hashing algorithm.

Harpreetet et al.[3] implemented a method for ensuring data security of the files uploaded by multiple users to the cloud storage. Blowfish encryption algorithm is used for data security and the message digest of the encrypted data is calculated using MD5 for data integrity. The method used in this paper show that the encrypted file size and time taken to encrypted and decrypted decreased as compared with the existing method.

Adviti et al.[4] aims to provide a parallel cryptographic algorithm using MD5 and Blowfish encryption algorithm..This scheme is compare with existing technique i.e. RSA-MD5 algorithm. The result shows that this scheme Blowfish and Md5 is more efficient as compared to the existing model RSA and MD5..

Divya et al.[5] create a cryptography model which consist of both symmetric encryption and asymmetric encryption algorithm . Blowfish provides data confidentiality whereas, RSA is used for authentication. This model consist SHA-2 for providing data integrity.The result of the model used in this paper has more security when transmitting data through communication network.

Rohini et al.[6] give more important on security problem in cloud computing. They use RSA encryption algorithm for providing data confidentiality. They used HMAC for providing data integrity over the cloud. The model used in this paper is compares with RSA Homomorphic technique . The result shows that it has more better performance as compare with the earlier model

3. PROBLEM STATEMENT AND METHODOLOGY

In paper [3], Author used Blowfish for encryption and MD5 for hashing .The output result shows input file size, encrypted sized of the file, time taken for encrypting and decrypting is decrease as compare with Diffie Hellman - AES technique. In paper [4], author Compared MD5-Blowfish with the existing technique RSA-MD5. The output result shows encrypted and decryption time is decrease as compare with RSA-MD5 technique.

3.1. Main issue of the above work:

a) No data integrity in transit
b) No authentication between client and server
c) Cloud storage is not efficiently utilizes.

3.2. Objective of our proposed work:

a) To provide data integrity in transit using hash function (Blake2b)
b) To provide user authentication by using password as key in hash function (Blake2b).
c) Efficient utilization of the memory space in cloud storage.

3.3. Proposed Model

This proposed model included Blowfish encryption algorithm for encrypting the data and BLAKE2b hashing algorithm for data integrity. Key can be used in BLAKE2b hashing algorithm making it functionally similar to a MAC. The authentication of the client is performed by using password as key in the BLAKE2b hashing algorithm. This authentication used in our proposed model is based on PAKE (Password Authentication Key Agreement) Protocol. . All the processes in our proposed model are implemented using Python Socket Programming in Spyder Software and SQLite as database for storing Users secure data and files.

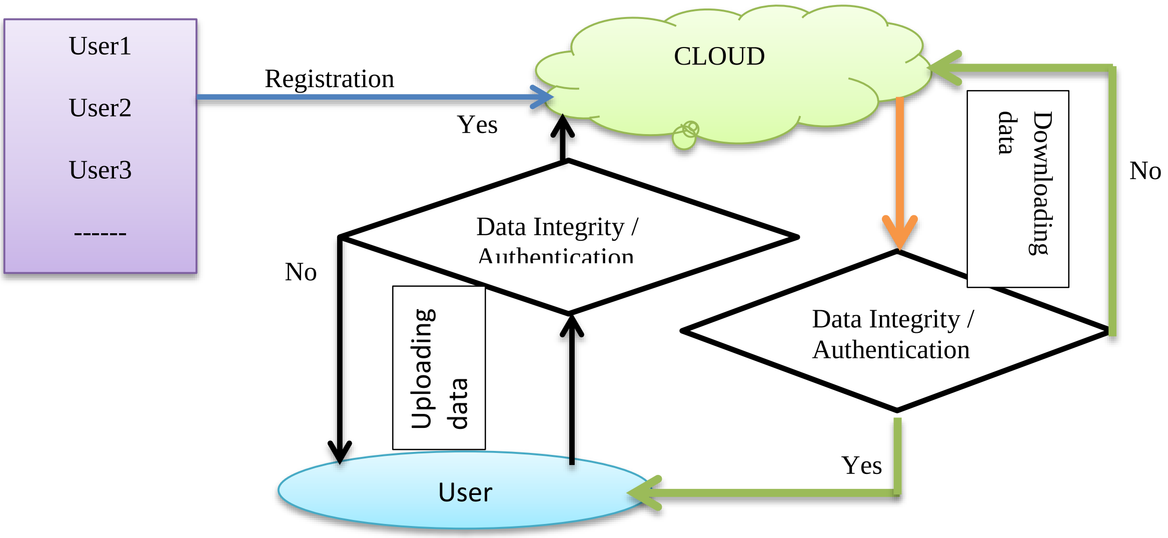


Fig 3.1 Simple flow diagram of the proposed model

TABLE 3.1: Notation used in our proposed model

x	a large prime number
N	N=2x+1
g	a generator modulo Q
H()	hash function(Blake2b)
S	a small salt
U	Username
P	User password
Vc	Password verifier calculated by the client after receiving salt(S) from server.
Vs	Password verifier of users in the cloud Registration database
|	Concatination

3.5. Registration

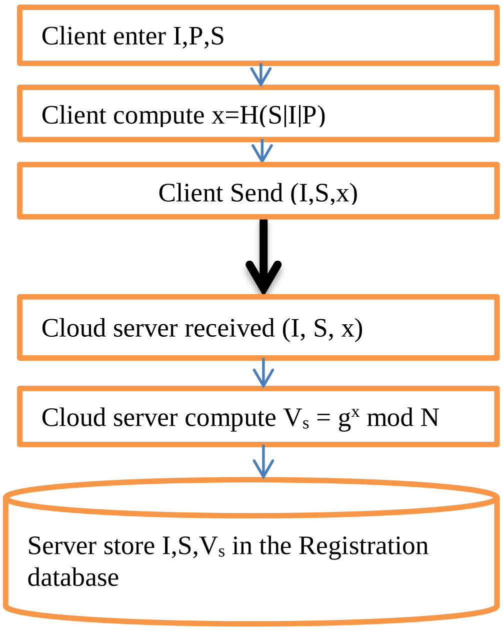

Client enter I,P,S
Client compute x=H(S|I|P)
Client Send (I,S,x)
Cloud server received (I, S, x)
Cloud server compute $V_s = g^x$ mod N
Server store I,S,V_s in the Registration database

Table 3.2

Steps	**Client Side**
1.	Enter I,P
2.	Generate a random S
3.	Compute x = H(S|I|P)
4.	Send I,S,x to the cloud

Table 3.3

Steps	**Server Side**
1.	Receiver I,S.x from client
2.	Compute V_s = gx mod N
3.	Store I,S,V_s in the Registration database.

Fig 3.2 Flow diagram of Registration process of the proposed model

3.6. Uploading and downloading data

Client performed the following task prior to uploading the data on the cloud:

1) Client sends I to the cloud server.
2) Server checks if I is in the registration database.
3) If I is found in registration database then

    S of I is return to the client.

    Client compute:

    a) x=H(S,I,P)
    b) Password verifier , Vc =gx mod N

   Else

    Return None.
4) V_C is later use by the client as key in the hashing function (blake2b).

3.6.1. Uploading data

Server checks data integrity and authentication before storing the data to the cloud database

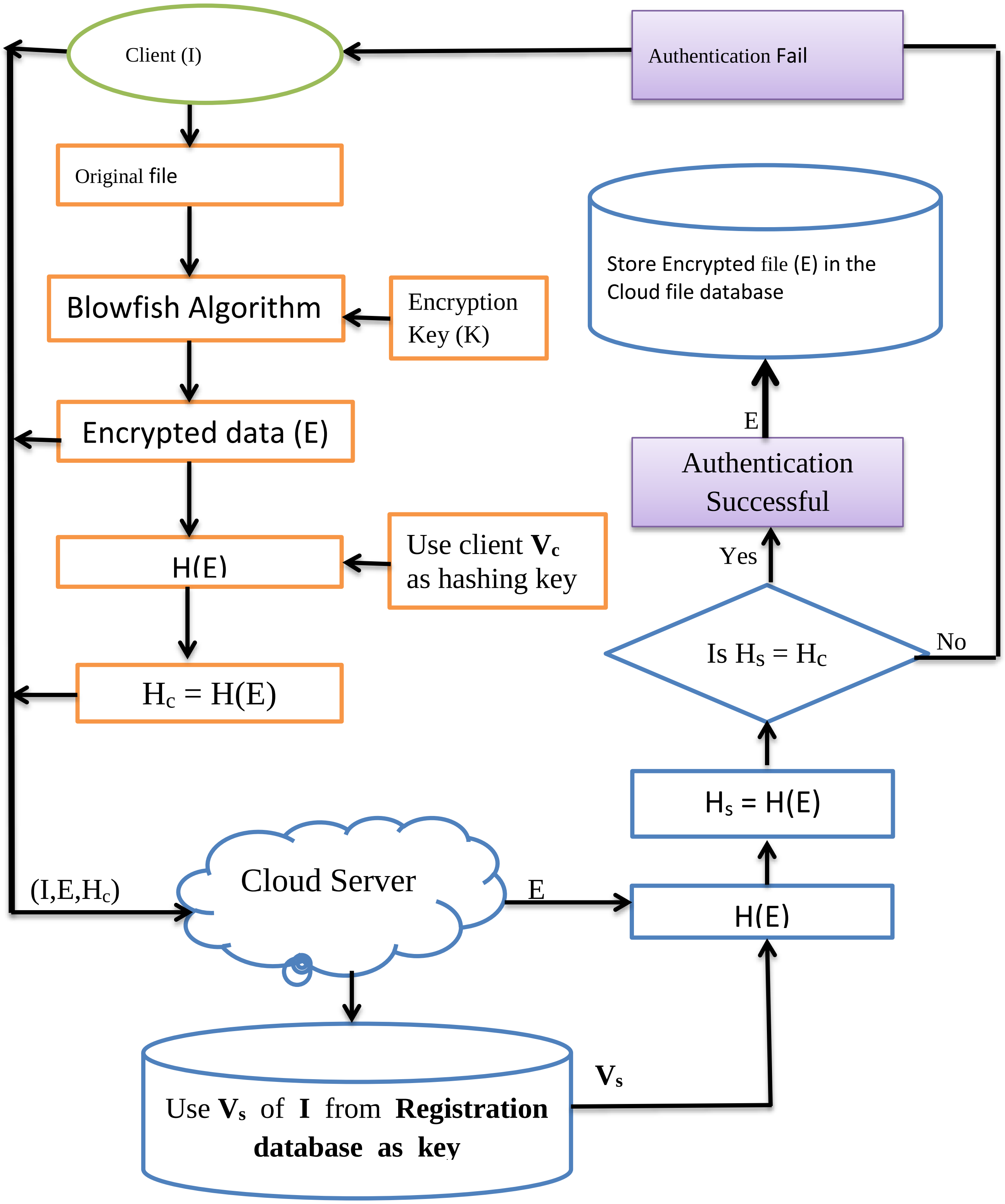


Fig 3.3 Flow diagram for uploading data on the cloud

3.6.1. Downloading data

Client checks data integrity and authentication before downloading the data from the cloud

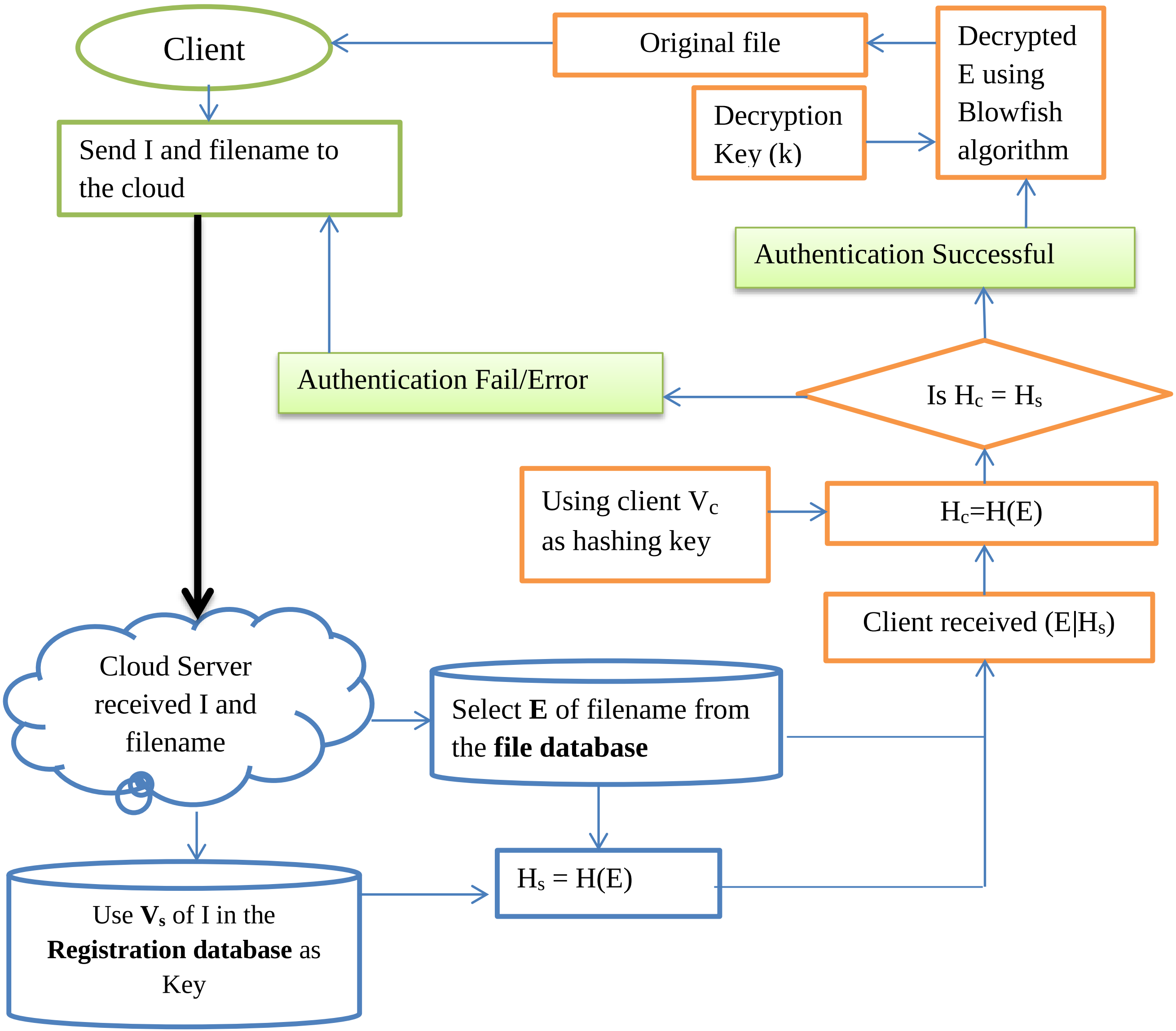
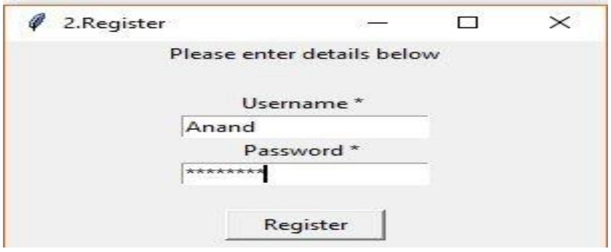

Client
Original file
Decrypted E using Blowfish algorithm
Decryption Key (k)
Send I and filename to the cloud
Authentication Successful
Authentication Fail/Error
Is $H_c = H_s$
Using client V_c as hashing key
$H_c=H(E)$
Client received $(E|H_s)$
Cloud Server received I and filename
Select **E** of filename from the **file database**
$H_s = H(E)$
Use $\mathbf{V_s}$ of I in the **Registration database** as Key

Fig 3.4 Flow diagram for downloading data from cloud

4. IMPLEMENTATION AND RESULT

The proposed model is implemented using Socket Programming in Python 3.6 and SQlite as database for storing Users information and secure data.

4.1. Registration Process


2.Register

Please enter details below

Username *

Anand

Password *

****

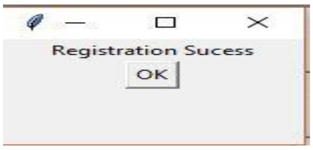

Registration Sucess

Fig 4.1 Client entering username and password

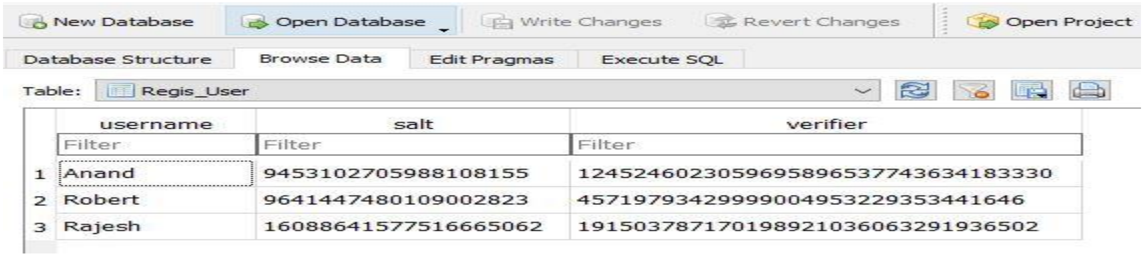


Table: Regis_User

	username	salt	verifier
	Filter	Filter	Filter
1	Anand	9453102705988108155	1245246023059695896537743634183330
2	Robert	9641447480109002823	4571979342999900495322935344l646
3	Rajesh	16088641577516665062	1915037871701989210360632919365O2

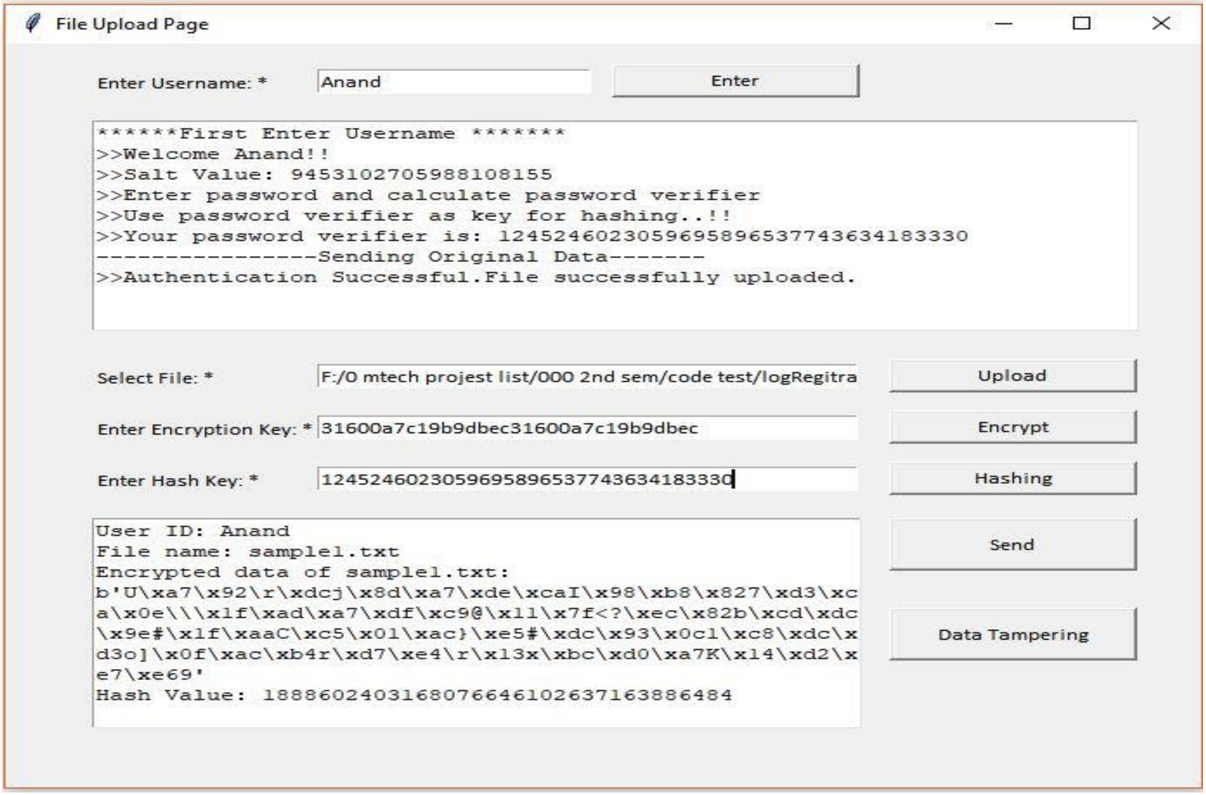
Fig 4.2 Registration Database containing Username, Salt and Password Verifier.

4.2. Uploading data on the cloud

4.2.1 Scenerio of Successful Authentication between Client and Server

The Send button in the Client side is used to send the Original encrypted data.

<u>Client Side</u>



Enter Username: * Anand

```
******First Enter Username *******
>>Welcome Anand!!
>>Salt Value: 9453102705988108155
>>Enter password and calculate password verifier
>>Use password verifier as key for hashing..!!
>>Your password verifier is: 1245246023059695896537743634183330
----------------Sending Original Data-------
>>Authentication Successful.File successfully uploaded.
```

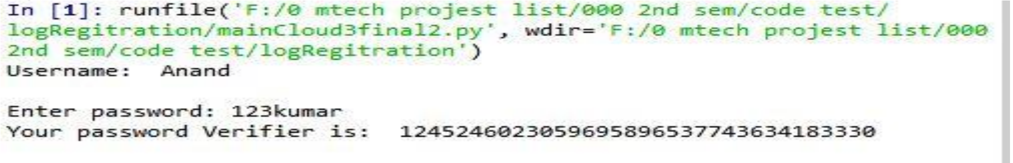
Select File: * F:/0 mtech projest list/000 2nd sem/code test/logRegitra

Enter Encryption Key: * 31600a7c19b9dbec31600a7c19b9dbec

Enter Hash Key: * 1245246023059695896537743634183330

```
User ID: Anand
File name: sample1.txt
Encrypted data of sample1.txt:
b'U\xa7\x92\r\xdcj\x8d\xa7\xde\xcaI\x98\xb8\x827\xd3\xc
a\x0e\\\x1f\xad\xa7\xdf\xc9@\x11\x7f<?\xec\x82b\xcd\xdc
\x9e#\x1f\xaaC\xc5\x01\xac}\xe5#\xdc\x93\x0c1\xc8\xdc\x
d3o]\x0f\xac\xb4r\xd7\xe4\r\x13x\xbc\xd0\xa7K\x14\xd2\x
e7\xe69'
Hash Value: 1888602403168076646102637163886484
```

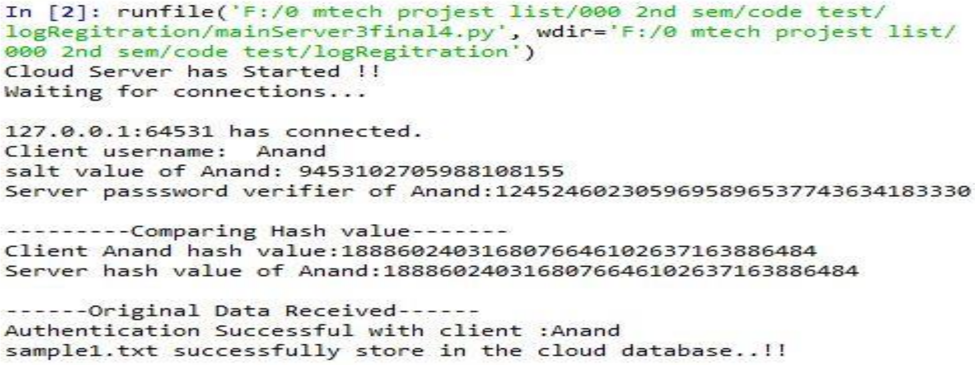
Fig 4.3 Output result of client successfully uploading data.



```
In [1]: runfile('F:/0 mtech projest list/000 2nd sem/code test/
logRegitration/mainCloud3final2.py', wdir='F:/0 mtech projest list/000
2nd sem/code test/logRegitration')
Username:  Anand

Enter password: 123kumar
Your password Verifier is:  1245246023059695896537743634183330
```

Fig 4.4 Client obtaining Password Verifier using Password

<u>Server Side</u>



```
In [2]: runfile('F:/0 mtech projest list/000 2nd sem/code test/
logRegitration/mainServer3final4.py', wdir='F:/0 mtech projest list/
000 2nd sem/code test/logRegitration')
Cloud Server has Started !!
Waiting for connections...

127.0.0.1:64531 has connected.
Client username:  Anand
salt value of Anand: 9453102705988108155
Server passsword verifier of Anand:1245246023059695896537743634183330

---------Comparing Hash value-------
Client Anand hash value:1888602403168076646102637163886484
Server hash value of Anand:1888602403168076646102637163886484

------Original Data Received------
Authentication Successful with client :Anand
sample1.txt successfully store in the cloud database..!!
```

Fig 4.6 Output result in the Server storing client data to the cloud database.

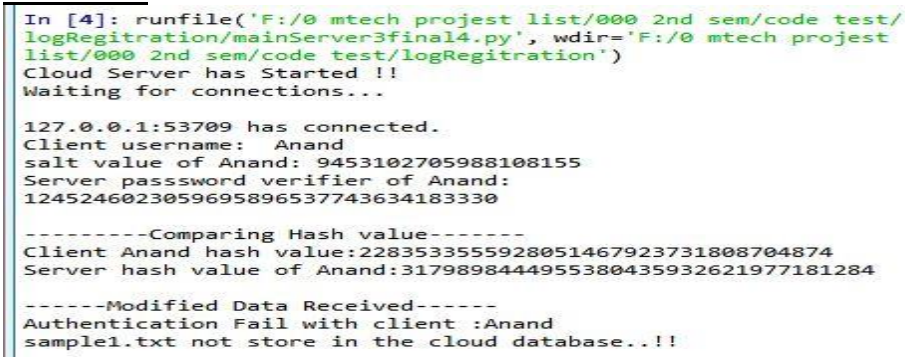
4.2.2 Client-Server Authentication fail scenerio

There are two case of Client-Server Authentication fail.

1. Client entering incorrect Password which results in obtaining incorrect Password Verifier (Vc)
2. Any modification of the Encrypted data while sending from client to server.

The “Data tempering” button is used to send the modified encrypted data to the server and to test whether the Server accept this modified encrypted data or not.

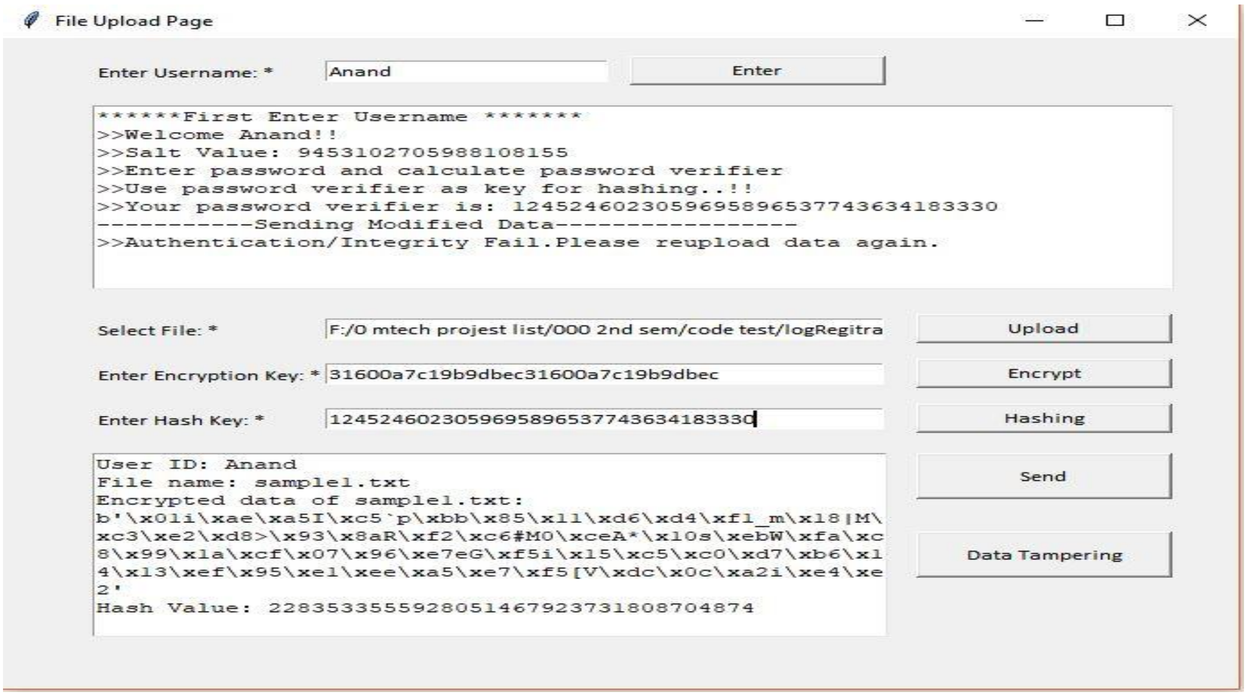
<u>Server Side</u>



```
In [4]: runfile('F:/0 mtech projest list/000 2nd sem/code test/
logRegitration/mainServer3final4.py', wdir='F:/0 mtech projest
list/000 2nd sem/code test/logRegitration')
Cloud Server has Started !!
Waiting for connections...

127.0.0.1:53709 has connected.
Client username:  Anand
salt value of Anand: 9453102705988108155
Server passsword verifier of Anand:
124524602305969589653774363418333O

---------Comparing Hash value-------
Client Anand hash value:2283533555928051467923731808704874
Server hash value of Anand:3179898444955380435932621977181284

------Modified Data Received------
Authentication Fail with client :Anand
sample1.txt not store in the cloud database..!!
```

Fig 5.8 Output result showing authentication fail due to client Hash value (Hc) ≠ Server Hash value (Hs)

Client Side



Enter Username: * Anand

```
******First Enter Username *******
>>Welcome Anand!!
>>Salt Value: 9453102705988108155
>>Enter password and calculate password verifier
>>Use password verifier as key for hashing..!!
>>Your password verifier is: 124524602305969589653774363418333O
-----------Sending Modified Data------------------
>>Authentication/Integrity Fail.Please reupload data again.
```

Select File: * F:/0 mtech projest list/000 2nd sem/code test/logRegitra

Enter Encryption Key: * 31600a7c19b9dbec31600a7c19b9dbec

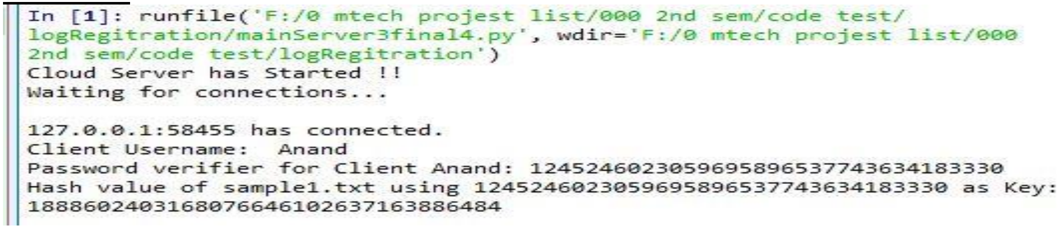
Enter Hash Key: * 124524602305969589653774363418333O

```
User ID: Anand
File name: sample1.txt
Encrypted data of sample1.txt:
b'\x01i\xae\xa5I\xc5`p\xbb\x85\x11\xd6\xd4\xf1_m\x18|M\
xc3\xe2\xd8>\x93\x8aR\xf2\xc6#M0\xceA*\x10s\xebW\xfa\xc
8\x99\x1a\xcf\x07\x96\xe7eG\xf5i\x15\xc5\xc0\xd7\xb6\x1
4\x13\xef\x95\xe1\xee\xa5\xe7\xf5[V\xdc\x0c\xa2i\xe4\xe
2'
Hash Value: 2283533555928051467923731808704874
```

Fig 5.8 Output result showing Authentication fail

4.3. Downloading data from the cloud

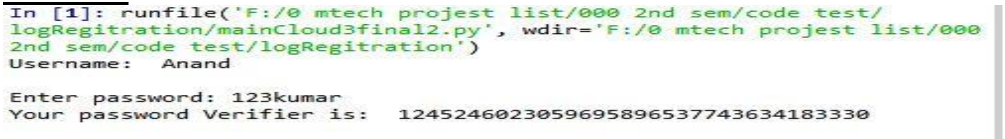
<u>Server side</u>



```
In [1]: runfile('F:/0 mtech projest list/000 2nd sem/code test/
logRegitration/mainServer3final4.py', wdir='F:/0 mtech projest list/000
2nd sem/code test/logRegitration')
Cloud Server has Started !!
Waiting for connections...

127.0.0.1:58455 has connected.
Client Username:  Anand
Password verifier for Client Anand: 124524602305969589653774363418333O
Hash value of sample1.txt using 124524602305969589653774363418333O as Key:
1888602403168076646102637163886484
```

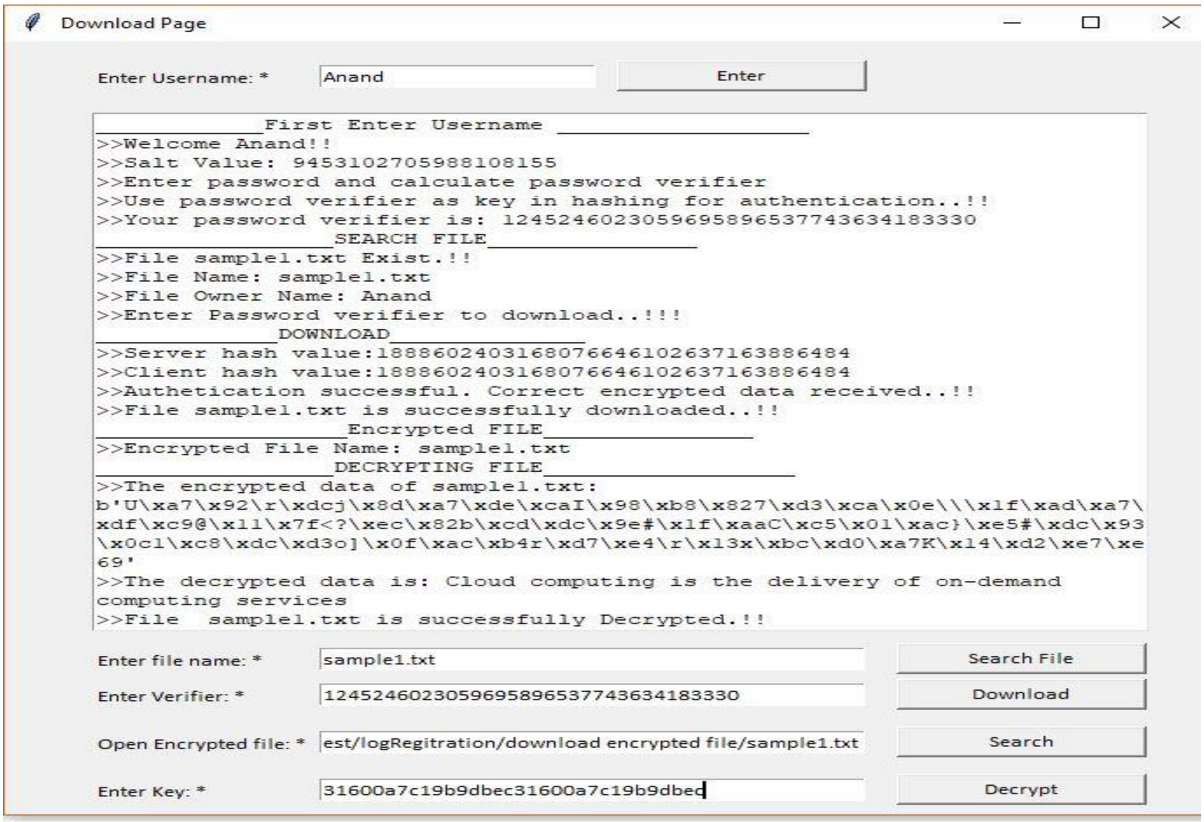
Fig 5.11 Output result of the server when Client request to download Encrypted file.

<u>Client Side</u>



```
In [1]: runfile('F:/0 mtech projest list/000 2nd sem/code test/
logRegitration/mainCloud3final2.py', wdir='F:/0 mtech projest list/000
2nd sem/code test/logRegitration')
Username:  Anand

Enter password: 123kumar
Your password Verifier is:  124524602305969589653774363418333O
```

Fig: Client used password to calculate password verifier V_C.



Enter Username: * Anand

```
_____________First Enter Username ____________________
>>Welcome Anand!!
>>Salt Value: 9453102705988108155
>>Enter password and calculate password verifier
>>Use password verifier as key in hashing for authentication..!!
>>Your password verifier is: 12452460230596958965377436341833330
_________________SEARCH FILE_________________
>>File sample1.txt Exist.!!
>>File Name: sample1.txt
>>File Owner Name: Anand
>>Enter Password verifier to download..!!!
_____________DOWNLOAD______________
>>Server hash value:18886024031680766461026371638864 84
>>Client hash value:18886024031680766461026371638864 84
>>Authetication successful. Correct encrypted data received..!!
>>File sample1.txt is successfully downloaded..!!
__________________Encrypted FILE_______________
>>Encrypted File Name: sample1.txt
_________________DECRYPTING FILE__________________
>>The encrypted data of sample1.txt:
b'U\xa7\x92\r\xdcj\x8d\xa7\xde\xcaI\x98\xb8\x827\xd3\xca\x0e\\\x1f\xad\xa7\
xdf\xc9@\x11\x7f<?\xec\x82b\xcd\xdc\x9e#\x1f\xaaC\xc5\x01\xac}\xe5#\xdc\x93
\x0c1\xc8\xdc\xd3o]\x0f\xac\xb4r\xd7\xe4\r\x13x\xbc\xd0\xa7K\x14\xd2\xe7\xe
69'
>>The decrypted data is: Cloud computing is the delivery of on-demand
computing services
>>File  sample1.txt is successfully Decrypted.!!
```

Enter file name: * sample1.txt

Enter Verifier: * 12452460230596958965377436341833330

Open Encrypted file: * est/logRegitration/download encrypted file/sample1.txt

Enter Key: * 31600a7c19b9dbec31600a7c19b9dbed

Fig 5.10 Various output result while downloading data from server database.

Table 4 Comparison between Proposed Model and Previous Model [4,5]

Services	Blowfish-Md5 model	Proposed Model (Blowfish-Blake2b)
Data privacy	Yes	Yes
Data integrity in the database	Yes	Yes
Data integrity in transit	No	Yes
Authentication while uploading data to server	No	Yes
Authentication while downloading data from server	No	Yes
Storage technique	Data sent to the cloud are directly store in the database	Data sent to the cloud are store in the database only after data integrity is maintain and Client-Server authentication is successful

5. CONCLUSION AND FUTURE WORK

In our proposed model, security is achieved through a technique of encryption using blowfish and blake2b algorithm. The proposed model allow client to upload encrypted file to the cloud only if no modification of the encrypted file occur during the transmission of data between the user and the cloud server. Hence this proposed model allow to utilized the Cloud storage more efficiently. The proposed model also provided authentication which allow server to except data only from the exact User and not from other entity. Authentication is also provided to client while downloading the data from the cloud database.

Thus the proposed model provided a lot of improvement compare to the existing model such as enhancing the integrity of data in transit, providing authentication while uploading and downloading the file and also efficiently utilization of cloud storage

In future, we can work on different approach of authentication with different cryptographic model and compare with the authentication method used our proposed model in term of efficiency and security.

REFERENCES:

[1] Prashant et al .(2013). “Use of Digital Signature with Diffie Hellman Key Exchange and AES Encryption Algorithm to Enhance Data Security in Cloud Computing”.
2013 International Conference on Communication Systems and Network Technologies.

[2] Priyanka et al. (2015)."Data Security and Integrity in Cloud Computing Based On RSA Partial Homomorphic and MD5 Cryptography". IEEE International Conference on Computer, Communication and Control (IC4-2015).

[3] Harpreet et al. (2017),"A Novel Technique of Data Security in Cloud Computing based on Blowfish with MD5 method ". International Journal of Advance Research, Ideas and Innovations in Technology (Volume 3, Issue 6) 2017.

[4] Adviti et al. (2017)."A Novel Technique of Cloud Security Based on Hybrid Encryption by Blowfish and MD5". 4th IEEE International Conference on Signal Processing ,Computing and Control(ISPCC 2k17),Sep 21- 23, 2017,solan , India

[5] Divya et al. (2017). "A Hybrid Cryptography Algorithm for Cloud Computing Security" . Vellore Institute of Technology Vellore (T. N), India, 2017 IEEE.

[6] Rohini et al. (2018)."Proposed hybrid RSA algorithm for cloud computing". Second International Conference on Inventive Systems and Control (ICISC 2018).

[7] Chen et al. (2012, March). Data security and privacy protection issues in cloud computing. In Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on (Vol. 1, pp. 647-651). IEEE.

[8] Kaaniche et al. (2013, June). ID based cryptography for secure cloud data storage. In CLOUD 2013: IEEE 6th International Conference on Cloud Computing (pp. 375-382). IEEE.

[9] Tirthani et al. (2014). Data Security in Cloud Architecture Based on Diffie Hellman and Elliptical Curve Cryptography. IACR Cryptology ePrint Archive, 2014, 49.

[10] Saarinen et al.(2015)."The BLAKE2 Cryptographic Hash and Message Authentication Code (MAC)",(ISSN: 2070-1721), November 2015.

[11] Yao et al.(2017)." Improved Client-to-Client Password-Authenticated Key Exchange Protocol", Second International Conference on Availability, Reliability and Security (ARES'07),(0-7695-2775-2/07).

[11] Taylor, David; Tom Wu; Nikos Mavrogiannopoulos; Trevor Perrin (November 2007). "Using the Secure Remote Password (SRP) Protocol for TLS Authentication" (http://tools.ietf.org/html/rfc5054). RFC 5054 .