IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 15, Issue 2 (Nov. - Dec. 2013), PP 86-94
www.iosrjournals.org

Automatic DNA Sequence Generation for Secured Effective Multi -Cloud Storage

D.Sureshraj[1], Dr.V.Murali Bhaskaran[2]
[1] *PhD Research Scholar, M.S.University, Tirunelveli District, TN*
[2] *Principal, Dhirajlal Gandhi College of Technology, Salem District, TN*

Abstract: *The main target of this paper is to propose an algorithm to implement data hiding in DNA sequences to increase the confidentiality and complexity by using software point of view in cloud computing environments. By utilizing some interesting features of DNA sequences, the implementation of a data hiding is applied in cloud. The algorithm which has been proposed here is based on binary coding and complementary pair rules. Therefore, DNA reference sequence is chosen and a secret data M is hidden into it as well. As result of applying some steps, M′′′ is come out to upload to cloud environments. The process of identifying and extracting the original data M, hidden in DNA reference sequence begins once clients decide to use data. Furthermore, security issues are demonstrated to inspect the complexity of the algorithm. In addition, providing better privacy as well as ensure data availability, can be achieved by dividing the user's data block into data pieces and distributing them among the available SPs in such a way that no less than a threshold number of SPs can take part in successful retrieval of the whole data block. In this paper, we propose a secured cost-effective multi-cloud storage (SCMCS) model in cloud computing which holds an economical distribution of data among the available SPs in the market, to provide customers with data availability as well as secure storage.*

Keywords*: DNA sequence; DNA base pairing rules; complementary rules; DNA binary coding; cloud service provider.*

I. Introduction

In order to protect data through the unsecure networks like the Internet, using various types of data protection is necessary. One of the famous ways to protect data through the Internet is data hiding. Because of the increasing number of Internet users, utilizing data hiding or Steganographic techniques is inevitable. Therefore, the role of data hiding has become more eminent nowadays. Before employing biological properties of DNA sequences, the common way of embedding a secret data into the host images was the traditional way of data hiding. The most important ones was the detection of the distortions of the image when the host image changed to some degrees. By advent of biological aspects of DNA sequences to the computing areas, new data hiding methods have been proposed by researchers, based on DNA sequences In order to convert binary data into amino acids as a DNA sequence, the base pairing rules must be used. Synthesizing nucleotides in real environment (biology) [1] is done in constant rules:

- ✓ Purine Adenine (A) always pairs with the pyrimidine Thymine (T)
- ✓ Pyrimidine Cytosine (C) always pairs with the purine Guanine (G)

Always, those rules are done naturally because the opportunities to synthesize hydrogen bonds between A and T (two bonds), and also between C and G. These concepts are named Watson-Crick base pairing rules.

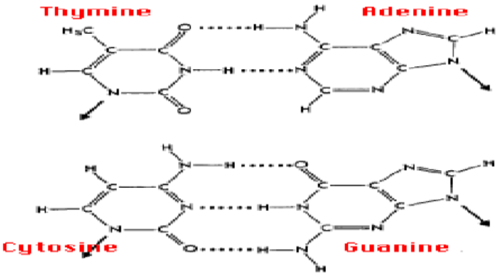


Figure 1. Synthesizing nucleotides in real environment

In binary computing area, it is possible to change the natural rules by own decision. For example, in biology A is synthesized to T while we can assume A to C or A to G, and so on, as we prefer. Increasing the complexity of the algorithm is the main purpose of the changing the rules. Consider A=00, T=01, C=10, and G=11 to convert binary data to DNA sequences. A way to increase the complexity is complementary pair rule. Complementary pair rule is a unique equivalent pair which is assigned to every nucleotides base pair. There are four basic alphabets therefore four likelihood of complementary rule for every DNA sequences. So, the final number of possible those rules are 4×3×2×1=24. On the other hand, the possibility to happen a correct guess is 1/24.

II. Associated Works

The most important part of each DNA base data hiding algorithm is, manipulating four letters which has been called as nucleotides in biology. The letters are A, C, G, and T. Any composition from them will make a sequence. For instance, two DNA sequences have been taken and this DAN sequences from European Bioinformatics Institute (which is known as EBI Database) for the purpose of extracting DNA sequences of Litmus and Balsaminaceae. So, Litmus with 154 nucleotides and Balsaminaceae with 2283 are shown in below, respectively:

Litmus:
"ATCGAATTCGCGCTGAGTCACAATTCGCGCTG
AGTCACAATTCGCGCTGAGTCACAATTGTGACTCA
GCCGCGAATTCCTGCAGCCCCGAATTCCGCATTGC
AGAGATAATTGTATTTAAGTGCCTGCTCGATACAA
TAAACGCCATTTGACC".

Balsaminaceae:
"TTTTTATTATTTTTTTCATTTTTTTCTCAGTTTT
TAGCACATATCATTACATTTTATTTTTTCATTACTTC
TATCATTCTATCTATAAAATCGATTATTTTTATCAC
TTATTTTTCTAATTTCCATATTTCATCTAATGATTAT
ATTACATTAAAGAAATCG".

All binary sequences are illustrated in form of s{0|1}e. Primer has key role to decrypt a coded strand. Using a public DNA strand was utilized as a reference sequence in their DNA based encryption technique. In this scheme, the receiver must also be informed about the reference sequence. Namely, the receiver will receive a selected primer and an encrypted strand. The intruder is not able to decrypt the binary data without knowing about both of primer and reference strand, certainly. A primer is a complementary subset from a sort of DNA strand. Normally, the primer is called a short substring.

For example, assume S is a DNA strand:
S= "ATGCTTAGTTCCATCGGAGACTAATGGCCTA" [2] and "two primers ATCAA and GATTAC". So "ATCAA and GATTAC are complementary substring of TAGTT and CTAATG", correspondingly. Definitely, a complementary rule is needed to handle the manipulations, correctly. For this reason, they defined a complementary rule which is AT, T-A, C-G, and finally G-C. So, according to the above, the proper output of the hybridization is:

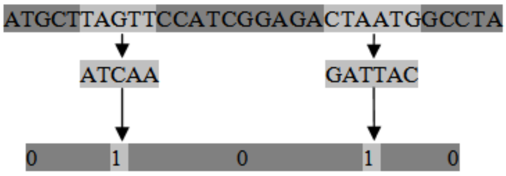


Figure 2. Binary representation

Therefore, the final secret message in form of binary is "01010".

III. Concepts

In this method, there is a cloud environment and its clients in a same company. The clients (client1 and client 2) want to upload data on cloud in such a manner that confidentiality of data be in highest point [3]. Therefore, the clients need to employ a method to increase the level of confidentiality of data so that no one can see data when someone intentionally or unintentionally accessed to them.

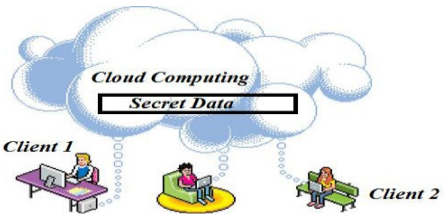


Figure 3. System Architecture

First, client1 must apply the method of data hiding on its data which it wants to hide to the cloud computing environments.

Data Hiding is divided into two phases.

- Embedding data [4].
- Extracting the original data [5].

3.1 Embedding Data - Client 1

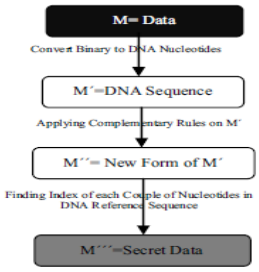

M= Data
Convert Binary to DNA Nucleotides
M´=DNA Sequence
Applying Complementary Rules on M´
M´´= New Form of M´
Finding Index of each Couple of Nucleotides in DNA Reference Sequence
M´´´=Secret Data

Figure 4. Embedding Data

Assume original data M=100111000011[6] should be uploaded to the cloud. DNA Reference Sequence:

$AT_1CG_2AA_3TT_4CG_5CG_6CT_7GA_8GT_9CA_{10}CA_{11}AT_{12}TC_{13}$
$GC_{14}GC_{15}TG_{16}AG_{17}TG_{18}AA_{19}CC_{20}$

M=100111000011

Sub-phase$_1$ *(A= 00, T= 01, C= 10, G= 11)*: M´= CTGAAG

Sub-phashe$_2$ *((AC) (CG) (GT) (TA))*: M´´= GATCCT

Sub-phase$_3$ *(Indexes)*: M´´´=8137

Figure 6. Original data convert to Secret data.

Now, embedding phase is finally completed. Then, sender sends 8137 to the cloud.

3.2 Extracting Data - Client 2

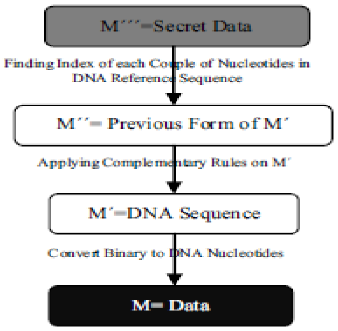
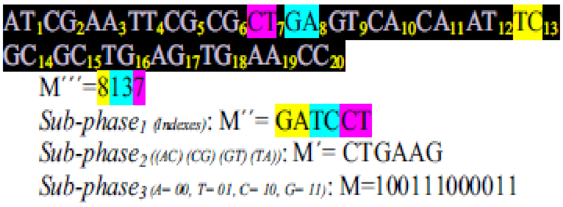

M´´´=Secret Data
Finding Index of each Couple of Nucleotides in DNA Reference Sequence
M´´= Previous Form of M´
Applying Complementary Rules on M´
M´=DNA Sequence
Convert Binary to DNA Nucleotides
M= Data

Figure 5. Extracting Data

Client2 takes the secret data in form of some numbers. For the purpose of extracting the original data from DNA reference sequence, phase two with its sub phases will extract the original data, correctly. Assume secret data M=8137 [6] should be downloaded from the cloud. DNA Reference Sequence:



$AT_1CG_2AA_3TT_4CG_5CG_6CT_7GA_8GT_9CA_{10}CA_{11}AT_{12}TC_{13}$
$GC_{14}GC_{15}TG_{16}AG_{17}TG_{18}AA_{19}CC_{20}$

M'''=8137

Sub-phase$_{1\ (Indexes)}$: M''= GATCCT

Sub-phase$_{2\ ((AC)\ (CG)\ (GT)\ (TA))}$: M'= CTGAAG

Sub-phase$_{3\ (A-00,\ T-01,\ C-10,\ G-11)}$: M=100111000011

Figure 6.Secret Data convert to Original data

So, the receiver extracted the original data, accurately by using a simple algorithm.

IV. Security Measures

In terms of security, each intruder must be aware from the following information, correctly. Without this fundamental information, possibility of extracting original data is near to zero, scientifically.

4.1 DNA reference sequence: There are 163 million DNA reference sequence on EBI database. Therefore, the likelihood of making a doing well conjecture by attacker is 1 /24.

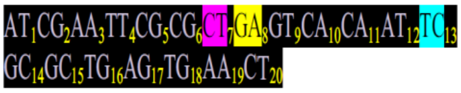
4.2 Binary coding rule: Clients are free to select any equivalent binary form for every nucleotide. It means that, A can be '00', '01', '10', or '11'; C can be '00', and so on. In other words, all the binary coding rules are 4×3×2×1=24. So, the likelihood of making correct guess by attacker is 1 /24.

4.3 Complementary pairing rule: Like binary coding rule, there is 4×3×2×1=24 complementary alphabet among basic nucleotides. Therefore, the possibility of making successful attack is 1 / 24. Eventually, the final probability of making a correct and successful guess by attacker is

V. Drawbacks in this System

5.1 The DNA Reference Sequence is selected from the EBI database, where 163 million DNA reference are stored.

The problem in the above concept is this system use huge size of Data base and we need the algorithm to pick the DNA sequence. It is time consuming process to pick the one DNA sequence out of 163 million sequences. Consider a Reference DNA Sequence [7].

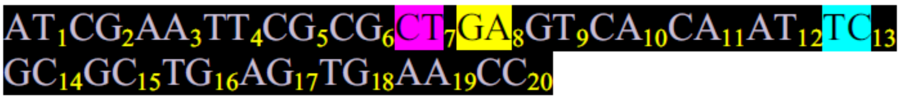


$AT_1CG_2AA_3TT_4CG_5CG_6CT_7GA_8GT_9CA_{10}CA_{11}AT_{12}TC_{13}$
$GC_{14}GC_{15}TG_{16}AG_{17}TG_{18}AA_{19}CT_{20}$

Figure 7. DNA sequence

The problem in above sequence is receptiveness CG is mentioned in three places 2, 5 and 6, CA is mentioned in two places 10 and 11. This will create problems in implementation.

1. Consider a Reference DNA Sequence



$AT_1CG_2AA_3TT_4CG_5CG_6CT_7GA_8GT_9CA_{10}CA_{11}AT_{12}TC_{13}$
$GC_{14}GC_{15}TG_{16}AG_{17}TG_{18}AA_{19}CC_{20}$

a. Some time system may fetch the sequence similar to the above sequence, The problem in this sequence is CC is missing. That means there is no guarantee that all the 16 combinations are available in the stored sequence. If this is happen system cannot able to fulfill the encryption.

2. Compulsory 16 Combinations
AA, AC, AT, AG,
CA, CC, CT, CG,
TA, TC, TT, TG,
GA, GC, GT, GG
3. In this system the extracting data has been uploaded without Encryption in cloud. Moreover, a single cloud service provider offered.
4. Even though in this system messages are spited and stored in different places, since it is not encrypted there is chance for the others to easy the part of the data and they can able to guess the content of the entire file.

VI. Proposed System

In the base paper DNA sequence is taken from the database where as in this application our program will create the DNA sequence based on the user id and shift key, which is more secure than the method specified in the paper.

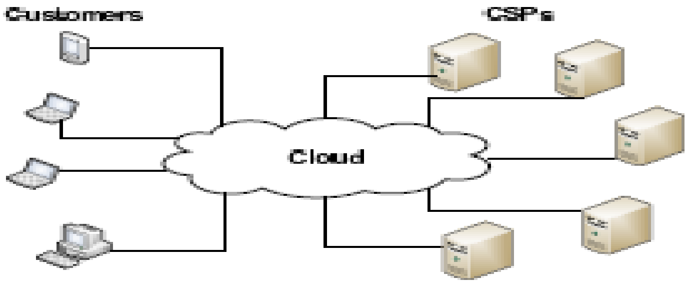
In terms of security, each intruder must be aware from the following information, correctly. Without this fundamental information, possibility of extracting original data is near to zero, scientifically. For each user system will assign 4 character DNA key eg: "ACTG" and it will generate the shift key range from 1 to 16 randomly.

So 16 combinations can be shifted based on the key give so we cab able to produce 16^{16} combinations (i.e.) there are more than 163 million DNA unique reference sequence system can generate. Only with small two inputs, Therefore, the likelihood of making a doing well conjecture by attacker is 1 / 24.

Binary coding rule: as mentioned, the clients are free to select any equivalent binary form for every nucleotide. It means that, A can be '00', '01', '10', or '11'; C can be '00', and so on. In other words, all the binary coding rules are 4×3×2×1=24. So, the likelihood of making correct guess by attacker is 1 / 24. Complementary pairing rule: like binary coding rule, there is 4×3×2×1=24 complementary alphabet among basic nucleotides. Therefore, the possibility of making successful attack is 1 /24. Eventually, the final probability of making a correct and successful guess by attacker is

$$\frac{1}{163\times10^{6}} \times \frac{1}{24} \times \frac{1}{24}$$

Also we can overcome the other two disadvantages mention previously without changing the probability of making guess by attackers. Now we have to use the Encrypted data and pushed to Cloud. One of the prominent services offered in cloud computing is the cloud data storage, in which; subscribers do not have to store their own data on their servers, where instead their data will be stored on the cloud service provider's servers. In cloud computing, subscribers have to pay the providers for this storage service. This service does not only provides flexibility and scalability data storage, it also provides customers with the benefit of paying only for the amount of data they needs to store for a particular period of time, without any concerns of efficient storage mechanisms and maintainability issues with large amounts of data storage. Customers can easily access their data from any geographical region where the Cloud Service Provider's network or Internet can be accessed



To providing better privacy as well as ensure data availability, can be achieved by dividing the user's data block into data pieces and distributing them among the available SPs in such a way that no less than a threshold number of SPs can take part in successful retrieval of the whole data block.

In our model the customer divides his data among several SPs available in the market, based on his available budget. Also we provide a decision for the customer, to which SPs he must chose to access data, with respect to data access quality of service offered by the SPs at the location of data retrieval. This not only rules out the possibility of a SP misusing the customers' data, breaching the privacy of data, but can easily ensure the data availability with a better quality of service.

From the business point of view, since cloud data storage is a subscription service, the higher the data redundancy, the higher will be the cost to be paid by the user. We provide an optimization scheme to handle the tradeoff between the cost that a cloud computing user is willing to pay to achieve a particular level of security for his data.

Privacy preservation and data integrity are two of the most critical security issues related to user data. But in case of cloud computing, the data is stored on an autonomous business party that provides data storage as a subscription service. The users have to trust the cloud service provider (SP) with security of their data. One bigger concern that arises in such schemes of cloud storage services, is that, there is no full-proof way to be certain that the service provider doe not retains the user data, even after the user opts out of the subscription. With enormous amount of time, such data can be decrypted and meaningful information can be retrieved and user privacy can easily be breached the user might not be availing the storage services from that service provider, To provide users with better and fair chances to avail efficient security services for their cloud storage at affordable costs, our model distributes the data pieces among more than one service providers, in such a way that no one of the SPs can retrieve any meaningful information from the pieces of data stored on its servers, without getting some more pieces of data from other service providers. We will describe our system model and the threat model.

Advantage of the proposed system is a secured cost-effective multi cloud storage (SCMCS) in cloud computing, which seeks to provide each customer with a better cloud data storage decisions, taking into consideration the user budget as well as providing client with the best quality of service (*Security and availability of data) offered by available cloud service* providers. By dividing and distributing customer's data, this model has shown its ability of providing a customer with a secured storage under affordable budget.

System Overview

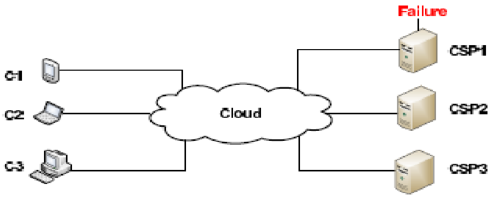
We consider the storage services for cloud data storage between two entities, *cloud users* (U) and *cloud service providers* (SP). The *cloud storage service* is generally priced on two factors, how much data is to be stored on the cloud servers and for how long the data is to be stored. In our model, we assume that all the data is to be stored for same period of time. We consider p number of cloud service providers (SP), each available cloud service provider is associated with a QoS factor, along with its cost of providing storage service per unit of stored data (C). Every SP has a different level of quality of service (QoS) offered as well as a different cost associated with it. Hence, the cloud user can store his data on more than one SPs according to the required level of security and their
affordable budgets.

Threat Model

Customers' stored data at cloud service providers is vulnerable to various threats. Cloud service provider can be a victim to Denial of service attacks or its variants. We consider two types of threat models.

1. **Single point of failure**

This will affect the data availability that could occur if a server at the cloud service provider failed or crashed, which make it hard for the costumer to retrieve his stored data from the server. Availability of data is also an important issue which could be affected, if the cloud service provider (SP) runs out of business. Such worries are no more hypothetical issues; therefore, a cloud service customer can not entirely rely upon a solo cloud service provider to ensure the storage of his vital data.

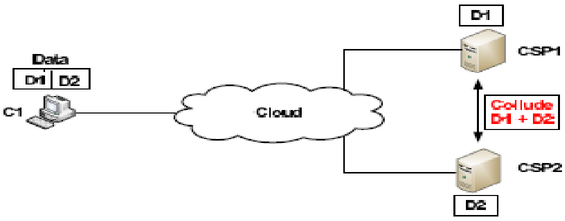


Assume that three customers (C1, C2 and C3) stored their data on three different service providers (CSP1, CSP2 and CSP3) respectively. Each customer can retrieve his own data from the cloud service provider who it has a contract with. If a failure occur at CSP1, due to internal problem with the server or some issues with the cloud service provider, all C1's data which was stored on CSP1's servers will be lost and cannot be retrieved. One solution for this threat is that, the user will seek to store his data at multiple service providers to ensure better availability of his data.

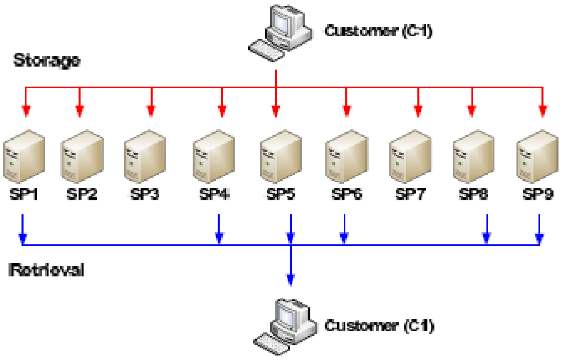
2. **Colluding service providers**

Colluding service providers which the cloud service providers might collude together to reconstruct and access the user stored data.

Let us assume that two cloud service providers are available for customer (C1), who wants to store his own data securely. In here he will divide his data into two parts (D1 and D2) and distribute these parts on the two available CSPs (CSP1 and CSP2) respectively. The two cloud service providers might collude with each other, and exchange the parts of data that the customer has stored on their server and reconstruct the whole data without being detected by the user.



Assume that we have 9 cloud service providers (SP1, SP2, ..., SP9). Let us assume that a customer (C1) has divided his own data he wish to store on some SP's servers into 9 data pieces. A customer required to retrieve at least 6 data pieces from different SPs to reconstruct his own data to get the full information, where in our example, six SPs will participate in the data retrieval (SP1, SP4, SP5, SP6, SP8 and SP9).



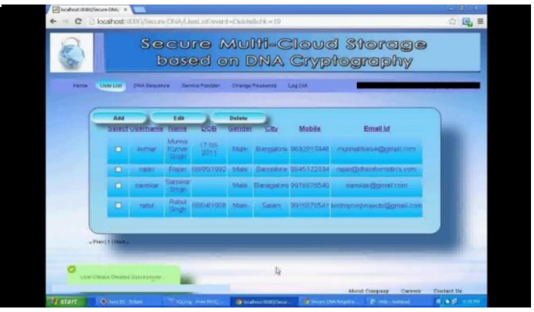
This will provide each customer with a better cloud data storage decisions, taking into consideration the user budget as well
6 as providing him with the best quality of service (Security and availability of data) offered by available cloud service providers. By dividing and distributing customer's data, our model has shown its ability of providing a customer with a secured storage under his affordable budget.

REPORTS

Admin Module reports

[redacted]

Press Esc to exit full screen mode.

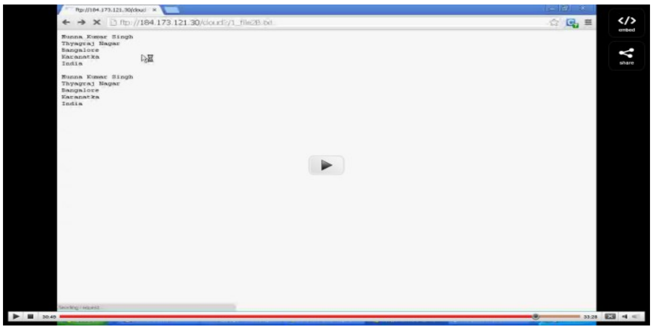


[redacted]

Select	Username	Name	DOB	Gender	City	Mobile	Email Id
□	kumar	Munna Kumar Singh	17-08-2011	Male	Bangalore	9632917446	munnabharse@gmail.com
□	rajan	Rajan	08/05/1992	Male	Bangalore	9945122334	rajan@dhsinformatics.com
□	sanskar	Sanskar Singh		Male	Banagalore	9916876540	sanskar@gmail.com
□	rahul	Rahul Singh	08/04/1988	Male	Salem	9916876541	testmyownprojects@gmail.com

User Details Deleted Successfully

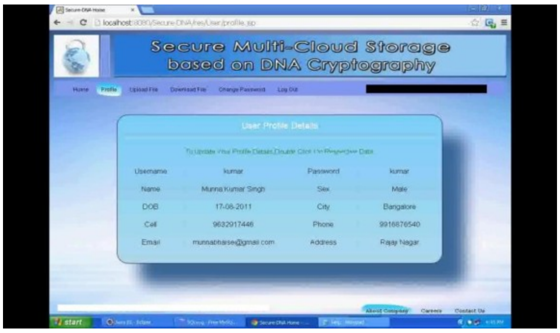
User Module reports



ftp://184.173.121.30/cloud?/1_file2B.txt

```
Munna Kumar Singh
Thyagraj Nagar
Bangalore
Karanatka
India

Munna Kumar Singh
Thyagraj Nagar
Bangalore
Karanatka
India
```

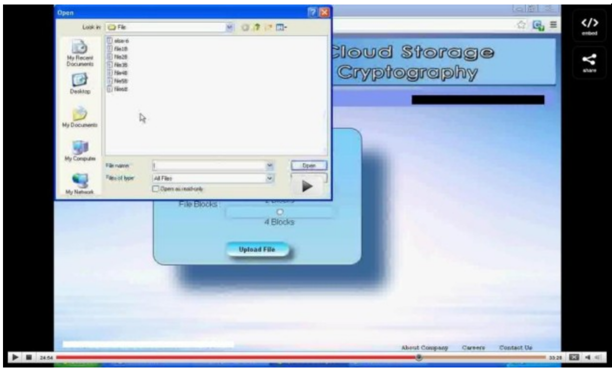


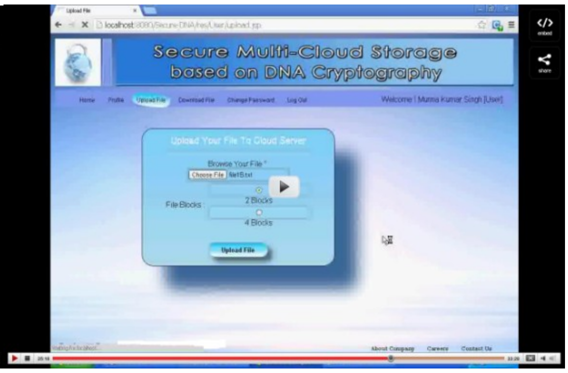
[redacted]

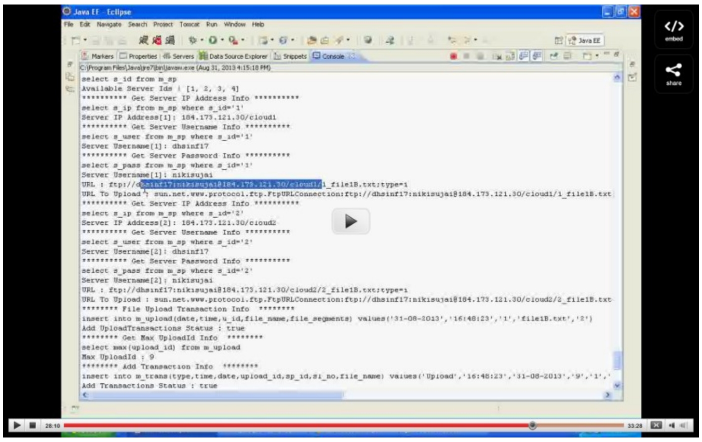
User Profile Details

To Update Your Profile Details,Double Click On Respective Data

Username	kumar	Password	kumar
Name	Munna Kumar Singh	Sex	Male
DOB	17-08-2011	City	Bangalore
Cell	9632917446	Phone	9916876540
Email	munnabharse@gmail.com	Address	Rajaji Nagar

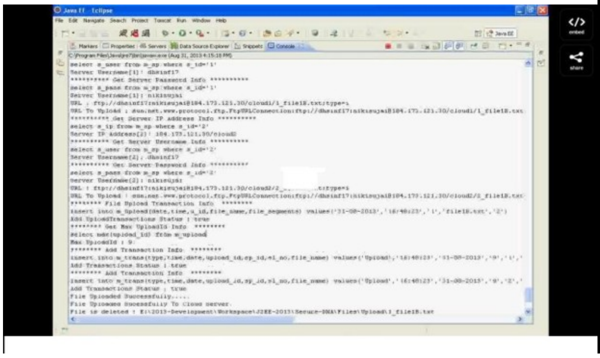




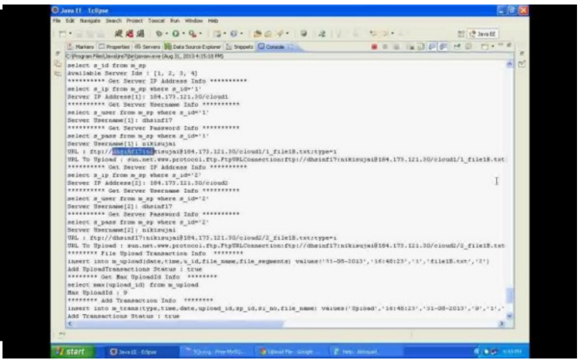


```
C:\Program Files\Java\jre7\bin\javaw.exe (Aug 31, 2013 4:15:18 PM)
select s_id from m_sp
Available Server Ids : [1, 2, 3, 4]
********** Get Server IP Address Info **********
select s_ip from m_sp where s_id='1'
Server IP Address[1]: 184.173.121.30/cloud1
********** Get Server Username Info **********
select s_user from m_sp where s_id='1'
Server Username[1]: dhsinf17
********** Get Server Password Info **********
select s_pass from m_sp where s_id='1'
Server Username[1]: nikisujai
URL : ftp://dhsinf17:nikisujai@184.173.121.30/cloud1/1_file1B.txt;type=i
URL To Upload : sun.net.www.protocol.ftp.FtpURLConnection:ftp://dhsinf17:nikisujai@184.173.121.30/cloud1/1_file1B.txt
********** Get Server IP Address Info **********
select s_ip from m_sp where s_id='2'
Server IP Address[2]: 184.173.121.30/cloud2
********** Get Server Username Info **********
select s_user from m_sp where s_id='2'
Server Username[2]: dhsinf17
********** Get Server Password Info **********
select s_pass from m_sp where s_id='2'
Server Username[2]: nikisujai
URL : ftp://dhsinf17:nikisujai@184.173.121.30/cloud2/2_file1B.txt;type=i
URL To Upload : sun.net.www.protocol.ftp.FtpURLConnection:ftp://dhsinf17:nikisujai@184.173.121.30/cloud2/2_file1B.txt
******** File Upload Transaction Info  ********
insert into m_upload(date,time,u_id,file_name,file_segments) values('31-08-2013','16:48:23','1','file1B.txt','2')
Add UploadTransactions Status : true
******** Get Max UploadId Info  ********
select max(upload_id) from m_upload
Max UploadId : 9
******** Add Transaction Info  ********
insert into m_trans(type,time,date,upload_id,sp_id,sl_no,file_name) values('Upload','16:48:23','31-08-2013','9','1','
Add Transactions Status : true
```

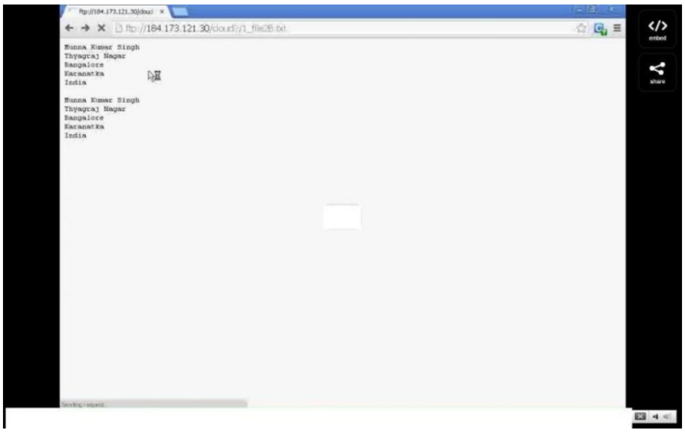
Sample Program Report



```
C:\Program Files\Java\jre7\bin\javaw.exe (Aug 31, 2013 4:15:18 PM)
select s_user from m_sp where s_id='1'
Server Username[1]: dhsinf17
********** Get Server Password Info **********
select s_pass from m_sp where s_id='1'
Server Username[1]: nikisujai
URL : ftp://dhsinf17:nikisujai@184.173.121.30/cloud1/1_file1B.txt;type=i
URL To Upload : sun.net.www.protocol.ftp.FtpURLConnection:ftp://dhsinf17:nikisujai@184.173.121.30/cloud1/1_file1B.txt
********** Get Server IP Address Info **********
select s_ip from m_sp where s_id='2'
Server IP Address[2]: 184.173.121.30/cloud2
********** Get Server Username Info **********
select s_user from m_sp where s_id='2'
Server Username[2]: dhsinf17
********** Get Server Password Info **********
select s_pass from m_sp where s_id='2'
Server Username[2]: nikisujai
URL : ftp://dhsinf17:nikisujai@184.173.121.30/cloud2/2_[illegible]t;type=i
URL To Upload : sun.net.www.protocol.ftp.FtpURLConnection:ftp://dhsinf17:nikisujai@184.173.121.30/cloud2/2_file1B.txt
******** File Upload Transaction Info  ********
insert into m_upload(date,time,u_id,file_name,file_segments) values('31-08-2013','16:48:23','1','file1B.txt','2')
Add UploadTransactions Status : true
******** Get Max UploadId Info  ********
select max(upload_id) from m_upload
Max UploadId : 9
******** Add Transaction Info  ********
insert into m_trans(type,time,date,upload_id,sp_id,sl_no,file_name) values('Upload','16:48:23','31-08-2013','9','1','
Add Transactions Status : true
******** Add Transaction Info  ********
insert into m_trans(type,time,date,upload_id,sp_id,sl_no,file_name) values('Upload','16:48:23','31-08-2013','9','2','
Add Transactions Status : true
File Uploaded Successfully.....
File Uploaded Sucessfully To Cloud server.
File is deleted : E:\2013-Development\Workspace\J2EE-2013\Secure-DNA\Files\Upload\1_file1B.txt
```



```
C:\Program Files\Java\jre7\bin\javaw.exe (Aug 31, 2013 4:15:18 PM)
select s_id from m_sp
Available Server Ids : [1, 2, 3, 4]
********** Get Server IP Address Info **********
select s_ip from m_sp where s_id='1'
Server IP Address[1]: 184.173.121.30/cloud1
********** Get Server Username Info **********
select s_user from m_sp where s_id='1'
Server Username[1]: dhsinf17
********** Get Server Password Info **********
select s_pass from m_sp where s_id='1'
Server Username[1]: nikisujai
URL : ftp://dhsinf17:nikisujai@184.173.121.30/cloud1/1_file1B.txt;type=i
URL To Upload : sun.net.www.protocol.ftp.FtpURLConnection:ftp://dhsinf17:nikisujai@184.173.121.30/cloud1/1_file1B.txt
********** Get Server IP Address Info **********
select s_ip from m_sp where s_id='2'
Server IP Address[2]: 184.173.121.30/cloud2
********** Get Server Username Info **********
select s_user from m_sp where s_id='2'
Server Username[2]: dhsinf17
********** Get Server Password Info **********
select s_pass from m_sp where s_id='2'
Server Username[2]: nikisujai
URL : ftp://dhsinf17:nikisujai@184.173.121.30/cloud2/2_file1B.txt;type=i
URL To Upload : sun.net.www.protocol.ftp.FtpURLConnection:ftp://dhsinf17:nikisujai@184.173.121.30/cloud2/2_file1B.txt
******** File Upload Transaction Info  ********
insert into m_upload(date,time,u_id,file_name,file_segments) values('31-08-2013','16:48:23','1','file1B.txt','2')
Add UploadTransactions Status : true
******** Get Max UploadId Info  ********
select max(upload_id) from m_upload
Max UploadId : 9
******** Add Transaction Info  ********
insert into m_trans(type,time,date,upload_id,sp_id,sl_no,file_name) values('Upload','16:48:23','31-08-2013','9','1','
Add Transactions Status : true
```


ftp://184.173.121.30/cloud
ftp://184.173.121.30/cloud1/1_file2B.txt

```
Munna Kumar Singh
Thyagraj Nagar
Bangalore
Karanatka
India

Munna Kumar Singh
Thyagraj Nagar
Bangalore
Karanatka
India
```

References

[1]. C.C. Chang, C.C. Lin, C.S. Tseng, W.L. Tai, Reversible hiding in DCTbased compressed images, Information Sciences 177 (2007).
[2]. C.C. Chang, W.C. Wu, Y.H. Chen, Joint coding and embedding techniques for multimedia images, Information Sciences 178 (2008).
[3]. A. Leier, C. Richter, W. Banzhaf, H. Rauhe, Cryptography with DNA binary strands, BioSystems 57 (2000).
[4]. I. Peterson, Hiding in DNA, Muse (2001).
[5]. Chunye Gong; Jie Liu; Qiang Zhang; Haitao Chen; Zhenghu Gong; , "The Characteristics of Cloud Computing," Parallel Processing
[6]. Workshops (ICPPW), 2010 39th International Conference on , vol., no., pp.275-279, 13-16 Sept. 2010.
[7]. Amazon.com, "Amazon s3 availablity event: July 20, 2008", Online at http://status.aws.amazon.com/s3-20080720.html, 2008
[8]. .http://csrc.nist.gov/groups/SNS/cloud-computing/index.html, 2009. [16] P. F. Oliveira, L. Lima, T. T. V. Vinhoza, J. Barros, M. M´edard, "Trusted storage over untrusted networks", IEEE GLOBECOM 2010, Miami, FL. USA.