© 2015 IJSRSET | Volume 1 | Issue 4 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Enhanced Data Security Model for Cloud Computing Platform

Ali I. Siam[1], Heba A. El-khobby[2], Hatem S. Abd Elkader[3], Mustafa M. AbdelNaby[4]
[1, 2, 4]Department of Electronics and Electrical Communication, Faculty of Engineering, Tanta University, Tanta, Egypt
[3]Department of Information Systems, Faculty of Computers and Information, Menoufia University, Menoufia, Egypt

ABSTRACT

Cloud Computing has continuously tremendous importance in the IT research field. As it is completely rely on sharing both physical and logical resources among numerous users, the concept of security must be considered, and deploying renewable security models must be taken into account. To obtain the highest level of security, some procedures must be implemented within a security model such as access control, authentication, authorization, data encryption, fast recovery, privacy, confidentiality, and attacks defending. This paper presents a proposed data security model for cloud computing. Some common attacks such as brute force attack and SQL injection are opposed. As data encryption plays the main role of data security, several modern encryption techniques are implemented and evaluated to deduce the most suitable technique to be used on cloud environment. The encryption techniques namely: AES, DES, 3DES, RC2, RC4, and Blowfish are evaluated based on randomness using NIST statistical testing. Also time consumed by each algorithm to decrypt the same amount of data is considered. Our model is deployed on Amazon Elastic Beanstalk cloud computing web server, and tests are run on two different environments; desktop environment and Amazon EC2 cloud computing environment. Our model is implemented using JSP language (Java Standard Edition), and NIST statistical tests are implemented using C# language.

Keywords: Cloud Computing, Security, Cryptography, NIST Tests, Google OAuth

I. INTRODUCTION

Nowadays, the usage of cloud computing for processing and storing data has become the most popular and reliable choice for both vast data amount foundation and simple personal email user. These data are transmitted over open and unsecured networks and almost opposed to attacks or even to be revealed by unwanted persons. When customers decide to move their data to the cloud, they can avoid the costs of building and maintaining a private storage infrastructure, this provides several benefits including availability (i.e., being able to access data from anywhere) and reliability (i.e., not having to worry about backups) at a relatively low cost [1]. But the issue must be considered is that they put their applications and data on remote servers that neither owned nor controlled by them. Cloud computing is based on sharing both physical and logical resources of the cloud among numerous users; therefore data from one user is stored and processed side by side data from other users, thus, it is important to secure data on the cloud to ensure safety of sensitive user data. In cloud computing the data is stored in an unknown place to the end user [2]. In order to achieve security and integrity, the place of data center is kept secret [3].

The concept of cloud computing can be simplified as it is a modern technology network model in which a person can make use of shared computing resources (applications, databases, storage, infrastructure) on the network from any place at any time by any device that can access the Internet (desktop, laptop, phone,…). The resources provider presents these computing resources as a service -by sharing them- for anyone to consume, the consumer pays for this service as much as he uses it. Cloud computing definition and reference architecture are described in NIST special publications [4, 5]. NIST has proposed a reference cloud computing model composed of three service models and four deployment models [4]. The service models are Software as a

Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). The deployment models are private cloud, community cloud, public cloud, and hybrid cloud. However, The ISO ruling has seven distinct cloud service categories, adding to the former service models: network as a service (NaaS), communication as a service (CaaS), computing as a service (CompaaS) and data storage as a service (DSaaS) [6]. The main advantages of cloud computing are: availability, quick deployment, simple backup and recovery, cost efficiency, lower power consumption than traditional infrastructure, and make customers focus on their own business rather than to have the knowledge necessary to develop and maintain the infrastructure and develop platforms and applications. On the other hand, the main disadvantages of cloud computing are: security issues which limits the growth of cloud computing, technical issues at which the customer has nothing to do as situation is controlled by cloud provider, provider dependent as one provider format may not work on another cloud provider, limited control, and internet connection dependent.

This paper presents a proposed security model and is applied on a real case study on cloud computing environment. Our case study is to implement and secure a cloud-based application for exam controls for post graduate studies in the Faculty of Engineering Tanta University, based on Amazon AWS public cloud platform and Google OAuth authentication protocol. Our application encrypts and decrypts students' data on runtime using six modern encryption techniques namely RC2, RC4, DES, 3DES, AES, and Blowfish, and then these encryption techniques results are evaluated to deduce the most suitable encryption technique to be used in cloud computing environment. NIST statistical tests are used to evaluate the randomness of the output of each encryption technique. Also, consumed time is a considerable factor in the evaluation process.

II. RELATED WORK

Many security models [7-14] have been proposed to enhance security in cloud computing platform. The main actor of most security models is data cryptography. Cryptography has two phases: Encryption and Decryption; Encryption is used to hide the meaning of a word by converting it into a meaningless text using encryption algorithm and secret key, the output of encryption process is called cipher text. Decryption is the reverse process in which the cipher text is supplied with the same secret key into decryption algorithm to give the original plain text.

Lo'ai Tawalbeh et al. [7] proposed a secure cloud computing model based on data classification, their model is used to secure data through using different security mechanisms with variable key sizes based on confidentiality level required for the data. According to Lo'ai Tawalbeh et al., data is encrypted according to its confidentiality degree through three levels: basic, confidential and highly confidential. Their proposed solution is based on the idea of manual classification, which means that the user will specify the confidentiality level of data. Eman M. et al. [8] proposed a cloud data security model that provides a single default gateway to encrypt sensitive data automatically in a real time before sending to the cloud storage. Eman M. et al. used OTP (One Time Password) to authenticate the user, and data is encrypted by some modern encryption techniques to get the highest security algorithm. Dai Yuefa et al. [9] proposed a data security model for cloud computing, this model used three-level defense system structure in which each floor performs its own duty to ensure the data security of cloud layers. These layers are authentication layer, data encryption layer, and fast recovery layer. Fernandez et al. [10] proposed a method to build a security reference architecture for clouds, and they validated their approach by showing that it can describe more precisely existing models and that it has a variety of uses. Rachna Jain et al. [11] used RSA and Triple DES cryptographic techniques to provide data security for users and studied the performance of the techniques on different aspects. The aspects are; avalanche effect, memory required, and simulation time. Jin-Mook Kim et al. [12] proposed a secure smart-work service model-based OpenStack for cloud computing. They showed that proposed model can be used to protect against DoS and SQL injection attacks. Zhao et al. [13] proposed five service deployment models to address security related issues in cloud computing. They argued that the proposed deployment models can address different issues that users are concerned about when deploying IT systems over cloud computing. These models are; separation model, availability model, migration model, tunnel model, and cryptography model. Lee [14] has proposed a secure authentication scheme for smart learning system in cloud computing

environment through User Authentication Service using two-factor method which mixed USIM_ID and login information of ID and password. Lee argued that the scheme proposed is safe to Password attack and replay attacks, and can be configured to protect the system from the man-in-the-middle attacks.

III. PROPOSED DATA SECURITY MODEL

Data security is the main concern of a cloud user, and the primary issue of cloud providers should be how to secure data within their clouds. Data in the cloud can be classified into three main categories. The first type is stored data which is data currently in rest in file storage or a database, the second type is data in transit that is currently transmitted within the cloud network, and the third type is processed data that is currently processed within the cloud computing engine. Each type of data should be secured by a different way.

Currently, Cloud providers use a fixed security model that based on authenticating the user requesting a service. Authentication assures that the user is the one he claims to be. The user firstly provides his credentials (username/password) to the cloud interface via secure connection. If the user is successfully authenticated, data is securely transferred (via SSL, HTTPS) from/to data centres. The main imprecation of that model is that data is processed, transferred, and then stored in plain form (not encrypted). However, recent researches [3, 8] have proposed to encrypt data before storing them in the cloud. The proposed security model is composed of four-level defence system structure. Each level performs a defensive task to complete the whole picture of the secured model as shown in Figure 1.

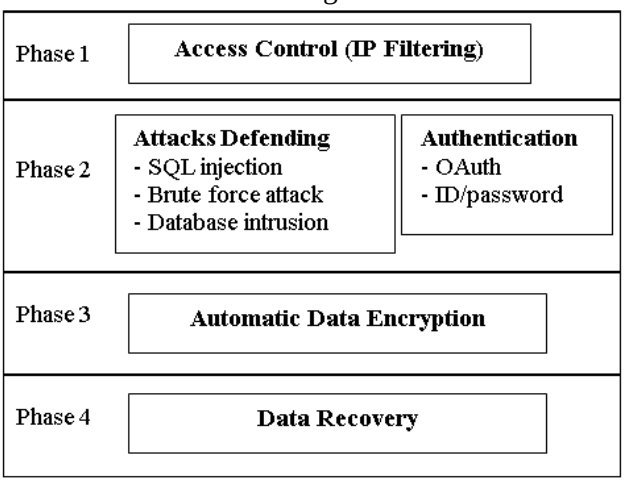


Figure 1: Proposed Data Security Model for Cloud computing

The first phase: Determines whether or not the user IP is permitted to access the network. A security group is created containing authorized IP addresses and ports to be allowed to access the network and/or unauthorized IP addresses to be denied to access the network.

The second phase: Has two responsibilities: Attacks defending and Authentication. Attacks defending layer is responsible for defending the system against software attacks (e.g. Brute force attack, SQL injection, Database intrusion). In the second layer, strong and safe authentication is achieved by using two approaches; Google OAuth [15] and the primary user credentials (username/password) authentication.

The third phase: Data is automatically encrypted (on real time) before going to storage, thus, the data is stored in the cloud in encrypted form.

The fourth phase: Assures fast recovery (decryption) of user data. Based on evaluation results of our research, the cloud user can select the most suitable decryption algorithm to achieve fast recovery of data.

IV. IMPLEMENTATION OF PROPOSED MODEL

Our case study is based on implementing an application that can be used as exam results control system in Faculty of Engineering Tanta University. The application is implemented using JSP language. This application is deployed on Amazon AWS cloud platform (an instance is created in EC2, S3 is used for file storage, RDS is the database engine, Elastic Beanstalk is used as web server) and secured by our proposed security model. Our case study interface is depicted in Figure 2.

Figure 2 shows the main interface of our application. First, the user provides the login type (staff or student), then he must successfully authenticated by both OAuth authentication and traditional ID/password authentication. At last, user selects technique to encrypt/decrypt data with.

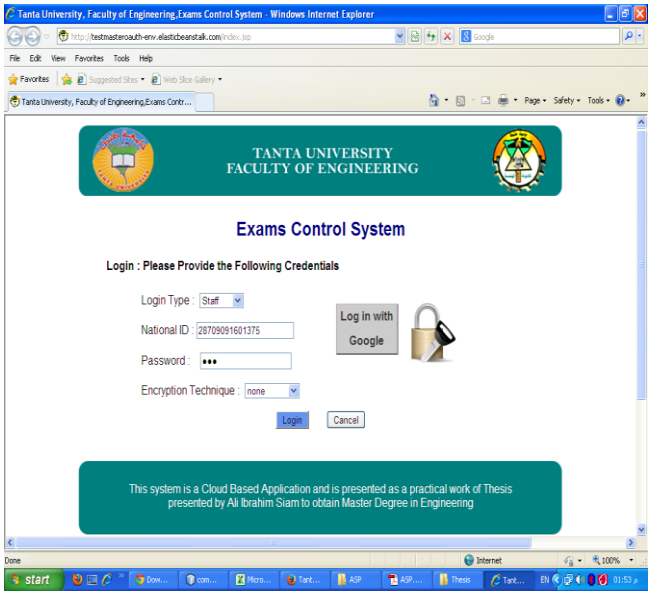


Figure 2: Case Study Interface

A. Implementation Details

Each phase of the security model is applied to the case study application as follow:

- **First phase: Access Control (IP Filtering)**

Access control procedure is implemented by creating a security group. A security group acts as a firewall that controls the traffic allowed to reach one or more EC2 instances [16]. Figure 3 shows a security group creation in EC2 instance.

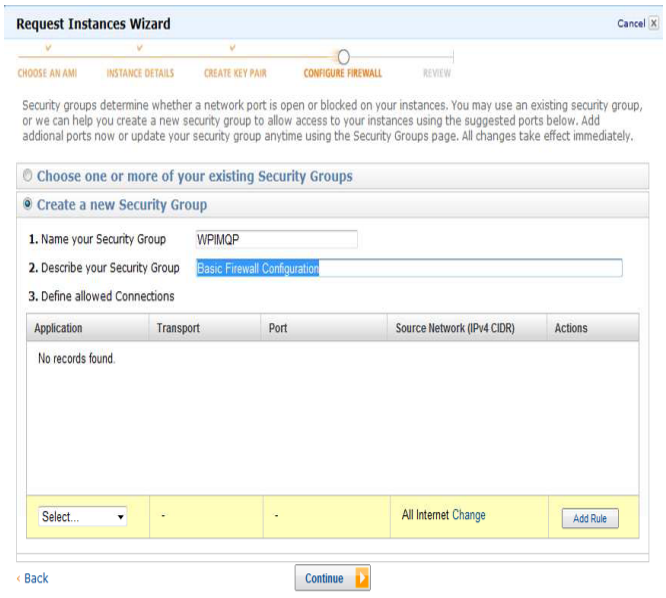


Figure 3: Creating a Security Group for EC2 Instance

The access control is achieved in the following sequence: as shown in Figure 4

1- The user requests to access our instance.
2- The cloud computing engine sends user's IP to the access control service that contains a list of trusted IP addresses that are allowed to access the instance and/or untrusted IP addresses that are denied to access the instance.
3- The access control service searches for user's IP in the IP lists to determine whether to allow or deny user's access.
4- If the user's IP exists in the trusted list, access is allowed, else, access is denied.

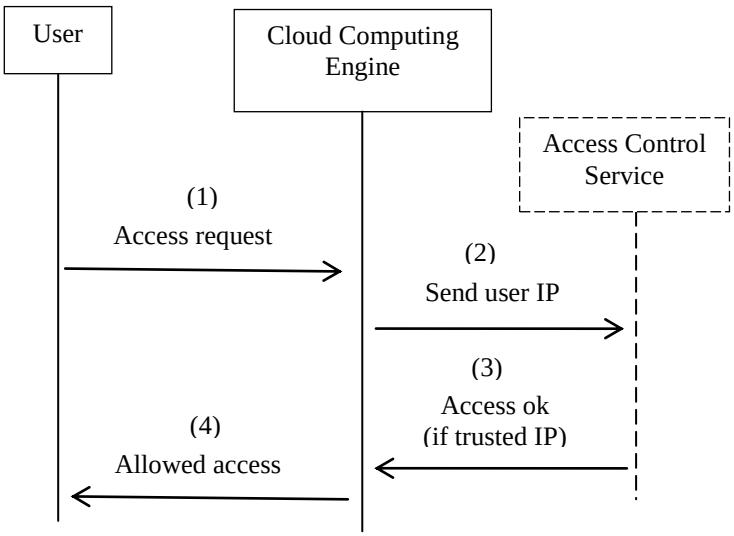


Figure 4: Access Control Phase of Proposed Model

- **Second Phase:**

a) Authentication

Two approaches are presented to fulfil strong authentication:

1. **Google OAuth:** OAuth is a trusted third party protocol developed by Google to authenticate its users. The user must have a valid Google account to proceed. When the user clicks "Log in with Google" button in Figure 2, he will be directed to Google authentication page as shown in Figure 5. If the user is not successfully authenticated, his access is denied; else, we read some information fields (email address) of the successfully authenticated account. This information is compared to a pre-entered value in our database, if the two values are matched, a valid OAuth authentication is declared. Figure 6

shows the application interface after valid OAuth authentication.

2. **Credentials Authentication:** in this step, the user provides his credentials (ID and password) obtained from system administrator. These credentials are compared to values in the database tables (staff users or student users) to check if they are belong to a valid user. If the user credentials are matched to a valid user, then a valid credentials authentication is declared. Once the user is authenticated, he can proceed to our application. If authentication failed, a message of wrong credentials will appear as shown in Figure 7.

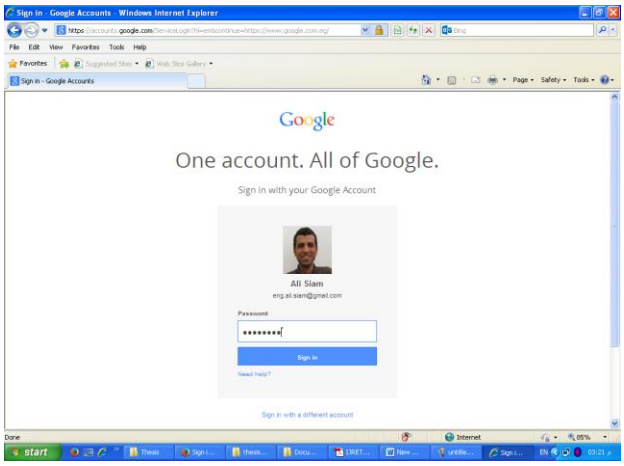


Figure 5: Google Login Page

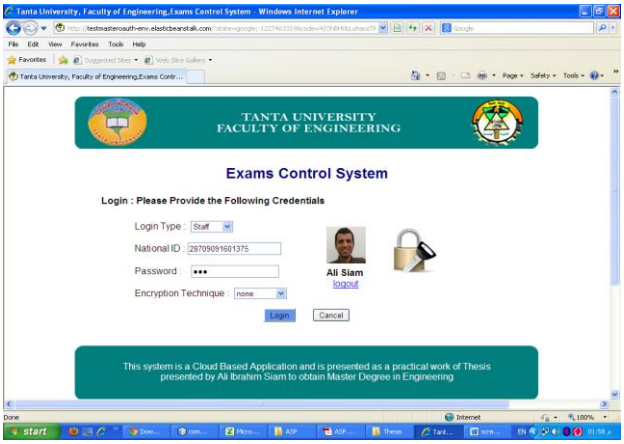


Figure 6: Valid OAuth Authentication

The user is authenticated in the following sequence; as shown in Figure 8.

1- The user requests OAuth authentication (first authentication)
2- The user is redirected to Google login page to enter his account credentials.
3- If the user has entered valid credentials for his Google account, Google authentication service replies with some information (email address) to indicate successful login.
4- The computing engine fetches the email address field relating to the user stored in the cloud storage database.
5- The cloud storage replies with the email address of the intended user.
6- The computing engine compares the two values (stored email address and Google email address). If they are matched, the user is successfully authenticated via OAuth authentication.
7- Successful OAuth authentication message to the user.
8- The user requests the primary authentication (second authentication), and sends his credentials (ID and password).
9- The computing engine fetches user's credentials stored in cloud storage.
10- The cloud storage database replies with the stored credentials of the user.
11- The computing engine compares sent credentials and stored credentials. If they are matched, the user is successfully authenticated.
12- Successful authentication message to the user and the user can pass through.

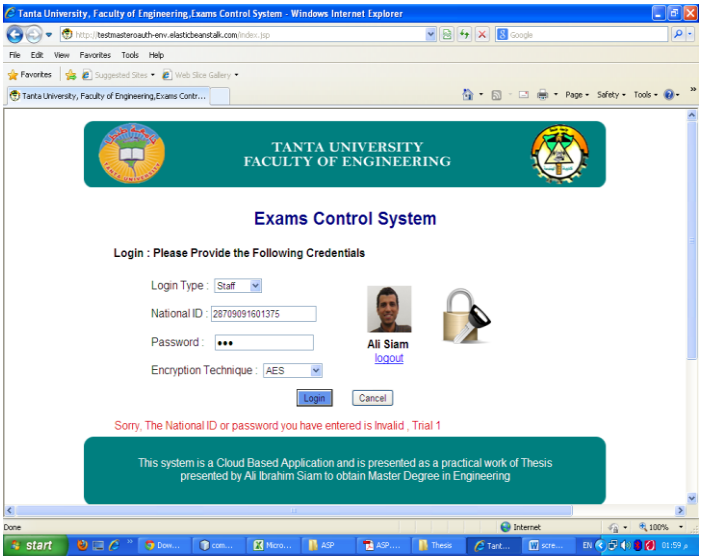


Figure 7: Failed Authentication

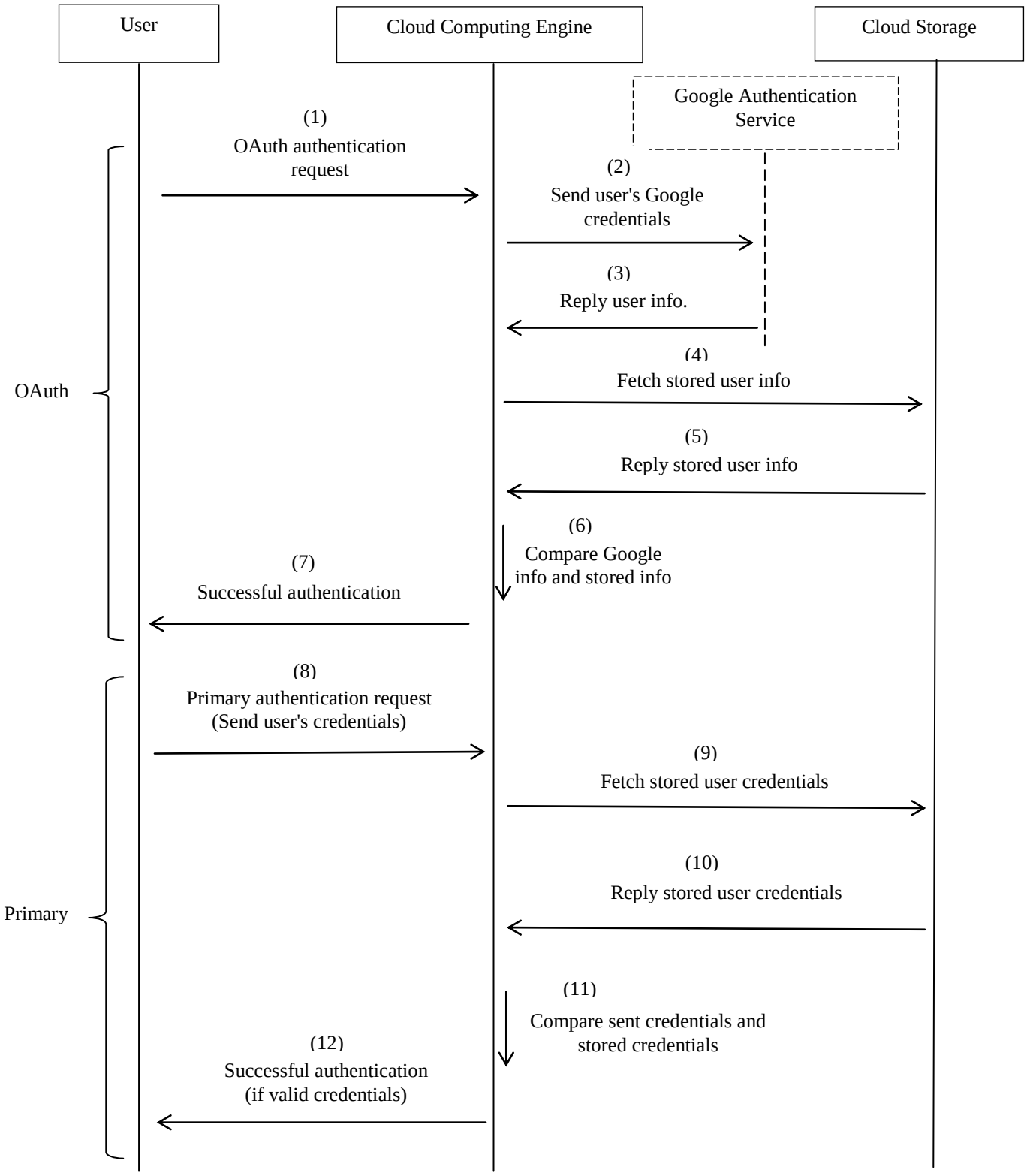


Figure 8: Authentication phase of proposed model

b) Attacks Defending

The following are samples of software attacks that have been prevented to threaten our application:

1. SQL Injection

SQL injection attempt is detected and prevented by our application as follows:
Consider the following SQL injection attempt:

i. The application prompts the user to enter his ID and password for authentication.
ii. The user enters (ID $_{provided}$, password $_{provided}$) as follows:
`ID provided = 123 OR 1=1`
`, password provided = 123 OR 1=1`
iii. The application executes the following query to check whether there is a user with ID= ID $_{provided}$ and password= password $_{provided}$.

```
query= "SELECT * FROM users WHERE
ID=ID provided
AND password=password provided"
```

User is successfully authenticated if the query returned a non-empty result set, else authentication failed.

iv. According to user input the query will be:

```
query= "SELECT * FROM users WHERE
ID=123 OR 1=1
AND password=123 OR 1=1"
```

This query is definitely true regardless id=123 or not as "1=1" check is almost true. And as a result, the query returns a non-empty result set indicating valid authentication despite wrong credentials.
To overcome this false authentication, an additional check is added to the returned result set (id $_{returned}$, password $_{returned}$) that is valid authentication is declared only if:
id $_{returned}$=ID $_{provided}$ and password $_{returned}$=password $_{provided}$.

If this condition is not achieved, the application records SQL injection attempt.
If the application detected SQL injection attempt, Figure 7 will be shown.

2. Brute Force Attack

Brute force attack is based on trying enormous attempts of reach valid credentials. This is almost done by robot programs to guess user password. In our application there is a limit of three wrong attempts to enter valid credentials, else, the user trying to authenticate is banned for 10 minutes. This is shown in Figure 9.

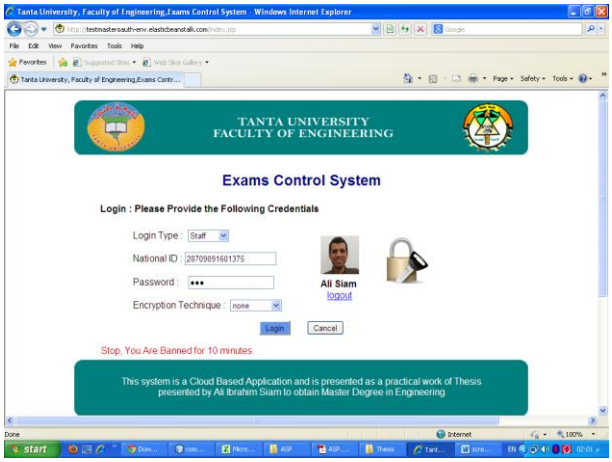


Figure 9: System bans user after three wrong attempts

3. Database hacking defending

Data in our database is encrypted before storage, thus, any one tries to hack the database he will get unreadable content.

- **Third phase: Data Encryption**

In this phase, data is automatically encrypted in real time using six modern encryption techniques namely: RC2, RC4, DES, 3DES, AES, and Blowfish. Students' data is encrypted and stored in separate tables in the database for each encryption technique. Data is encrypted in the following sequence as shown in Figure 10.

1. The user selects algorithm to encrypt/decrypt data with.
2. The user can update data, these data is encrypted with selected algorithm.
3. The encrypted content is updated in the database.

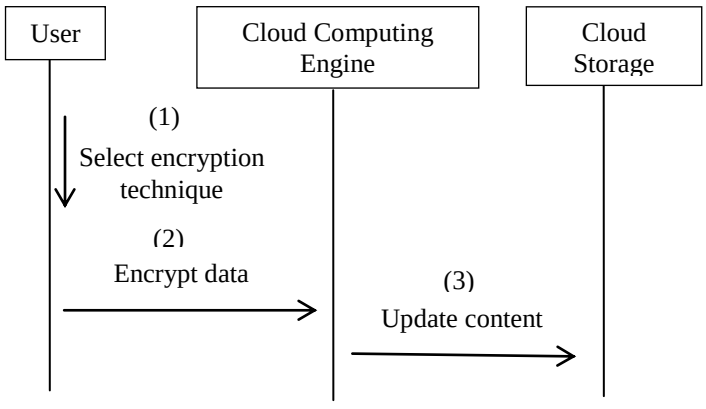


Figure 10: Data Encryption Phase

- **Fourth Phase: Data Recovery**

Fast recovery of data is achieved in this phase. In the initial application interface (Figure 3) the user selects the algorithm to decrypt data with to ensure fast recovery. Data recovered in the following sequence as shown in Figure 11.

1- The user queries some information from the database.
2- The computing engine fetches encrypted data from the database.
3- The database replies with the encrypted data.
4- The computing engine delivers encrypted data to the user interface.
5- The user decrypts data with the selected algorithm.
6- Data is handled in Encryption-Decryption flow upon request from the user.

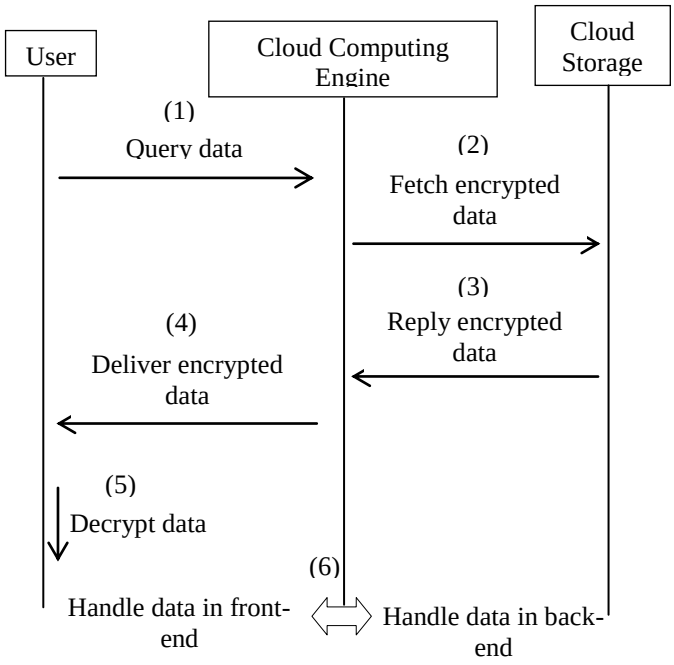


Figure 11: Data Recovery Phase

B. Proposed Model Evaluation

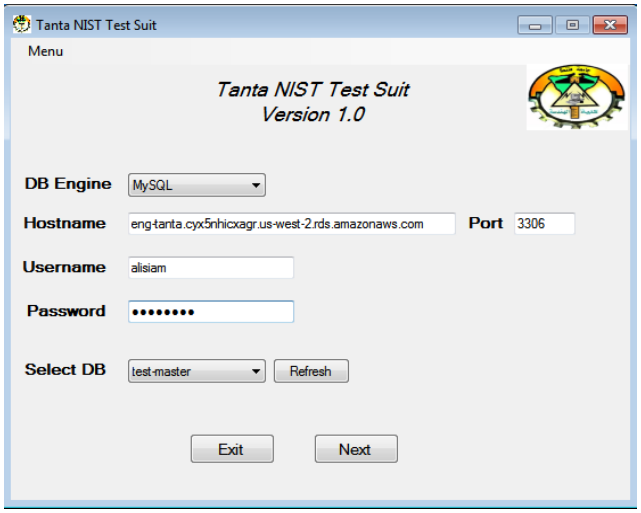
The evaluation of the model to deduce the most suitable encryption algorithm is based on three factors: randomness, rejection rate, and speed of each algorithm. The algorithms involved are RC2, RC4, DES, 3DES, AES, and Blowfish. NIST statistical tests are used to test the output sequence of each algorithm to produce a set of p-values indicating whether a particular sequence is random or not.

NIST statistical tests has 15 tests namely The Frequency Test, Frequency Test within a Block, The Runs Test, Tests for the Longest-Run-of-Ones in a Block, The Binary Matrix Rank Test, The Discrete Fourier Transform Test, The Non-overlapping Template Matching Test, The Overlapping Template Matching Test, Universal Statistical Test, The Linear Complexity Test, The Serial Test, The Approximate Entropy Test, The Cumulative Sums Test, The Random Excursions Test, and The Random Excursions Variant Test.

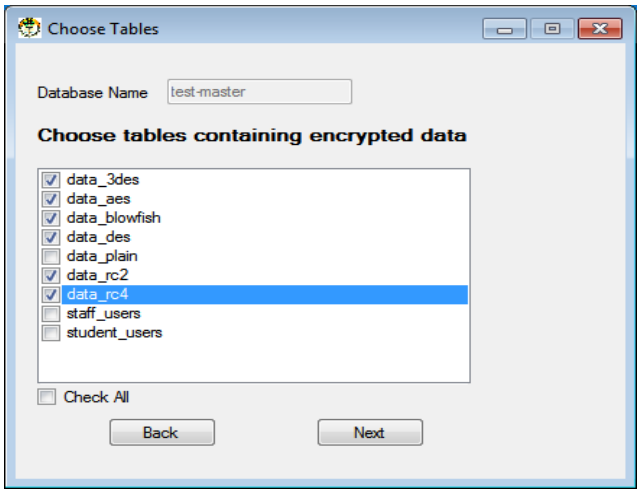
For a sequence to pass a NIST test, the following test must be verified: p-value $\geqslant$ 0.01. And to accept a sequence to be random, it must pass all 15 tests (i.e. all p-values corresponding to that sequence must be $\geqslant$ 0.01) else, reject the sequence. P-value is an indication of the randomness of the sequence, the higher p-value, the better the sequence and vice versa. Rejection rate is the ratio of number of rejected sequences to total number of sequences generated by algorithm. The lower rejection rate, the better the algorithm and vice versa.

The third factor of evaluation is the time taken by each algorithm to decrypt the same amount of data.
We have implemented a software program (Tanta NIST Test Suit) to implement the NIST tests. This software helped us to compute p-values, sequence rejection decision to compute the rejection rate, and plot visual cryptography for each algorithm. Tanta NIST Test Suit is shown in Figure 12.

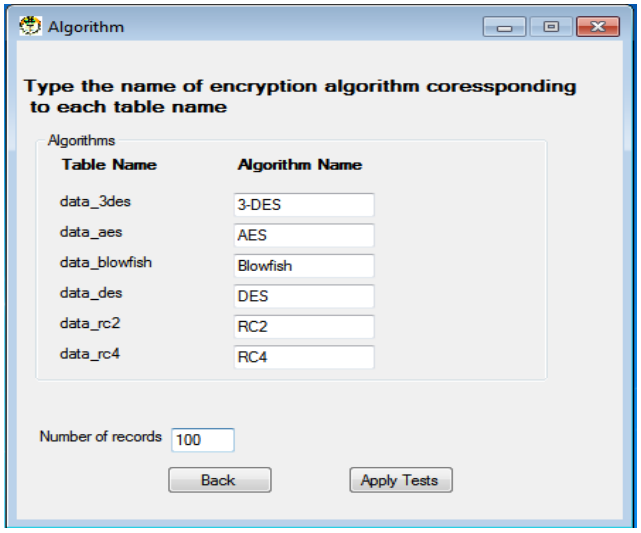
The experiments were performed on desktop (laptop, Microsoft Windows XP 32bit, Intel Celeron™ 2.13 GHz CPU, 3 GB RAM). We used in cloud computing, Amazon EC2 micro instance (Microsoft Windows Server 2012b,Xeon 2.4 GHz, 1 GB RAM, 64 bit operating system, 1 virtual core CPU).



a- Interface



b- Selecting Data Tables



c- Algorithms window

Figure 12: Proposed Evaluation Software

V. SIMULATION RESULTS

The proposed software will extract 100 cipher sequences from each algorithm related database table and then apply NIST statistical tests on them. The following results have been obtained:

A. Randomness Tests Results

After NIST tests have been run, there will be 41 p-values corresponding to each of 100 cipher sequences from each of six encryption algorithms. A sample output of NIST suit is depicted in Figure 13.

```
Applying NIST tests for sequence 100
Input sequence to test for randomness:
1101 1101 1111 1111 1001 1001 1010 0111 0011 1000 0011 0011 0001
1101 0010 0010 1110 0100 1110 1110 0110 1100 ……(truncated)
1. Testing input frequencies
SUCCESS:        pValue for Frequency test = 0.6166
2. Testing input blocks (block length = 8)
SUCCESS:        pValue for Block test = 0.7547
3. Testing input runs
SUCCESS:        pValue for Runs test = 0.5643
4. Testing Longest-Run-of-Ones in a Block
SUCCESS:        pValue for Longest-Run-of-Ones in a Block test =
0.5426
5. Testing Binary Matrix Rank ( M=8, Q=8 )
SUCCESS:        pValue for Binary Matrix Rank test = 0.4142
6. Testing Discrete Fourier transform
SUCCESS:        pValue for DFT test = 0.3973
7. Testing Non-overlapping Template Matching (N = 8, template =
111111111)
SUCCESS: pValue for Non-overlapping Template Matching test = 0.0572
(truncated output)
```

Figure 13: Sample Output of NIST Test Suit

The average p-values for all 15 NIST tests are shown in Figure 14. This is a clear comparison between the six modern encryption techniques that indicates the highest security algorithm based on randomness test. The results showed that RC2 has the greatest average p-value compared to other algorithms followed by Blowfish. Also, it can be noticed that RC4 has the lowest average p-value.

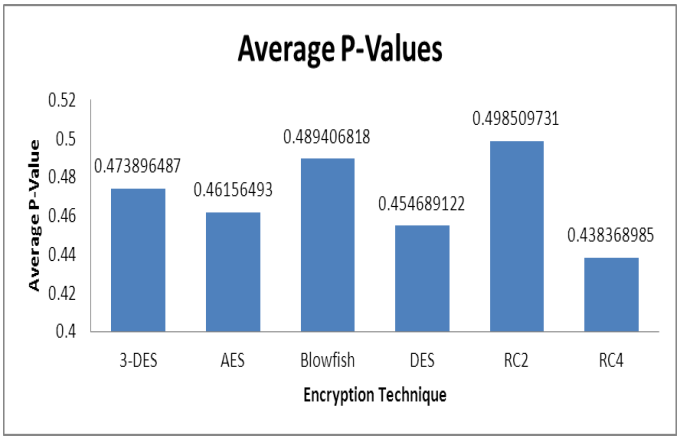


Figure 14: Average P-values in Amazon EC2

B. Rejection Rate Results

Table I below shows the number of rejected sequences for encryption techniques per NIST test (the sequence is rejected if the corresponding p-value is less than 0.01). The rejection rate for tested encryption techniques is depicted in Figure 15. It shows that Blowfish has the lowest average rejection rate compared to other algorithms. It can be noticed that RC4 algorithms has the highest average rejection rate.

TABLE I
REJECTED SEQUENCES

Test #		3-DES	AES	BLOWFISH	DES	RC2	RC4
1	Frequency (monobit)	0	1	0	0	0	81
2	Frequency (block)	1	0	0	1	1	0
3	Runs test	0	0	0	0	0	0
4	Longest run of ones in a block	0	3	0	1	0	0
5	Binary matrix rank	2	1	0	0	0	40
6	Discrete fourier transform (spectral)	5	2	2	7	4	48
7	Non-overlapping template matching	11	91	1	12	100	10
8	Overlapping template matching	1	28	1	1	13	0
9	Maurer's universal statistical	2	4	0	0	4	22
10	Linear complexity	27	0	4	24	0	5
11	Serial (1)	1	7	0	0	8	100
	Serial (2)	0	10	1	0	17	100
12	Approximate entropy	27	16	21	28	10	100
13	Cumulative sums (mode=0)	0	0	0	0	0	69
	Cumulative sums (mode=1)	0	4	0	0	0	96
14	Random excursions (1)	2	11	4	2	1	2
	Random excursions (2)	2	8	1	1	1	5
	Random excursions (3)	1	6	0	0	1	7
	Random excursions (4)	1	6	1	1	2	0
	Random excursions (5)	2	3	4	8	0	2
	Random excursions (6)	6	0	1	9	0	0
	Random excursions (7)	3	6	2	5	4	0
	Random excursions (8)	1	7	5	2	2	5
15	Random excursions variants (1)	1	5	3	1	0	0
	Random excursions variants (2)	1	3	1	0	0	0
	Random excursions variants (3)	1	3	0	0	0	0
	Random excursions variants (4)	1	3	0	1	0	0
	Random excursions variants (5)	1	2	0	0	0	0
	Random excursions variants (6)	1	1	0	0	0	0
	Random excursions variants (7)	0	0	0	0	0	0
	Random excursions variants (8)	0	3	0	0	0	0
	Random excursions variants (9)	3	2	0	1	0	2
	Random excursions variants (10)	7	3	1	6	3	2
	Random excursions variants (11)	10	2	1	10	1	0
	Random excursions variants (12)	9	1	1	9	0	0
	Random excursions variants (13)	4	0	2	7	0	0
	Random excursions variants (14)	2	0	5	3	0	0
	Random excursions variants (15)	2	0	8	4	1	0
	Random excursions variants (16)	3	0	6	2	1	0
	Random excursions variants (17)	2	0	6	2	2	0
	Random excursions variants (18)	3	1	6	3	1	0

Note: These rejected sequences are based on 100 tests.

C. Algorithm Speed Results

In this approach, a Java function is used to compute the time consumed by each algorithm to decrypt the same amount of data. Readings are taken for data of 100, 250, 400, 550 students. Figure 16 shows the time taken by tested algorithms in the cloud computing environment for different amount of data.

It can be showed that RC4 algorithm consumes little time to decrypt data in the cloud environment, while Blowfish algorithm consumes more time to decrypt the same amount of data. In desktop environment, RC4 still has the superiority over other algorithms in terms of computation time; this is shown in Figure 17. From Figures 16 and 17, it is clear that computation in EC2 environment needs lower time to accomplish the same task performed on desktop environment; this is fully agreeing with the fact that users with low-end stations can switch their computational tasks to more powerful servers in the cloud.

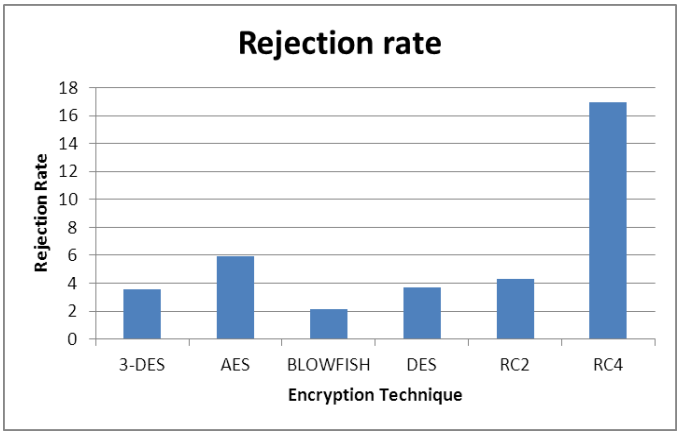


Figure 15: Average Rejection Rate in Amazon EC2

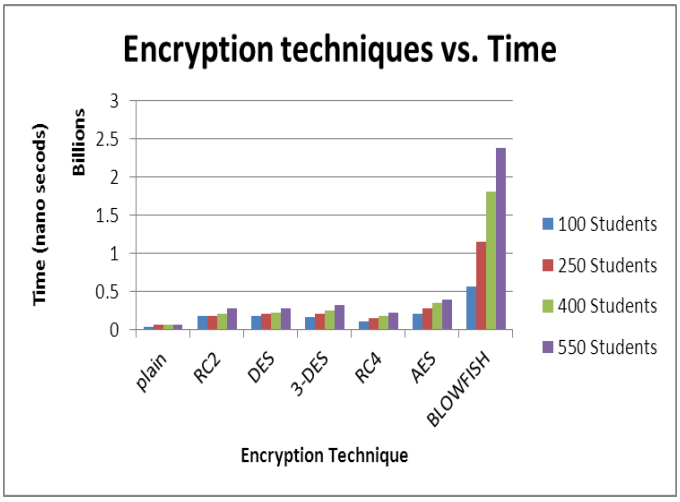


Figure 16: Encryption techniques versus time in Amazon EC2

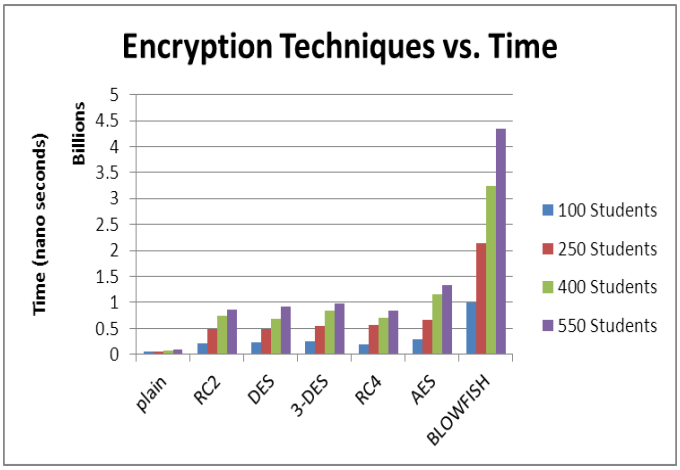


Figure 17: Encryption techniques versus time in desktop environment

VI. CONCLUSION

In this paper, we presented a proposed data security model for cloud computing based on four phases; access control phase, authentication and attacks defending phase, data encryption phase, and fast data recovery phase. We implemented a cloud-based application based on this model and is deployed on Amazon EC2 cloud environment. Access control is achieved through IP filtering. We used Google OAuth and credentials authentication to authenticate the user. Data encryption plays the main role in proposed model. Six modern encryption techniques, namely: RC2, RC4, DES, 3DES, AES, and Blowfish are implemented in our application and evaluated on the cloud environment to deduce the most suitable encryption technique to be used in cloud computing platform. Results show that RC4 consumes little computation time, while presents poor performance in randomness and rejection rate tests. Blowfish has the best results for randomness and rejection rate tests but works at more time in decryption. From the performance evaluation, user can select the best suited encryption technique to accomplish fast recovery of data.

VII. REFERENCES

[1] S. Kamara and K. Lauter, Cryptographic cloud storage, in Financial Cryptography and Data Security2010, Springer. p. 136-149.

[2] N.S. Kumar, G.R. Lakshmi, and B. Balamurugan, Enhanced Attribute Based Encryption for Cloud Computing. Procedia Computer Science, 2015. 46: p. 689-696.

[3] M. Christodorescu, R. Sailer, D.L. Schales, D. Sgandurra, and D. Zamboni, Cloud security is not (just) virtualization security: a short paper. in Proceedings of the 2009 ACM workshop on Cloud computing security. 2009. ACM.

[4] M. Mell and T. Grance, The NIST definition of cloud computing. 2011.

[5] F. Liu, J. Ting, J. Mao, R. Bohn, J. Messina, L. Badger, and D. Leaf, NIST cloud computing reference architecture. NIST special publication, 2011. 500: p. 292.

[6] ISO and IEC Publish Cloud Computing Standards. Information Standards Quarterly, 2014. 26(4): p. 24-24.

[7] L.a. Tawalbeh, N.S. Darwazeh, R.S. Al-Qassas, and F. AlDosari, A Secure Cloud Computing Model based on Data Classification. Procedia Computer Science, 2015. 52: p. 1153-1158.

[8] E.M. Mohamed, H.S. Abdelkader, and S. El-Etriby, Data Security Model for Cloud Computing. Journal of Communication and Computer, 2013. 10: p. 1047-1062.

[9] W.B. Dai Yuefa, G. Yaqiang, Z. Quan, and T. Chaojing, Data security model for cloud computing. in Proceedings of the 2009 International Workshop on Information Security and Application (IWISA 2009) Qingdao, China. 2009.

[10] E.B. Fernandez, R. Monge, and K. Hashizume, Building a security reference architecture for cloud systems. Requirements Engineering, 2015: p. 1-25.

[11] R. Jain, S. Madan, and B. Garg, Implementation and Comparison of RSA and Triple DES Algorithm For Encryption and Decryption In Cloud Environment. International Journal of Applied Engineering Research, 2015. 10(5).

[12] J.-M. Kim, H.-Y. Jeong, I. Cho, S.M. Kang, and J.H. Park, A secure smart-work service model based OpenStack for Cloud computing. Cluster Computing, 2014. 17(3): p. 691-702.

[13] G. Zhao, C. Rong, M.G. Jaatun, and F.E. Sandnes, Reference deployment models for eliminating user concerns on cloud security. The Journal of Supercomputing, 2012. 61(2): p. 337-352.

[14] A. Lee, Authentication scheme for smart learning system in the cloud computing environment. Journal of Computer Virology and Hacking Techniques, 2014: p. 1-7.

[15] Using OAuth 2.0 to Access Google APIs Available from: https://developers.google.com/identity/protocols/OAuth2.

[16] AWS Documentation, Getting Started with AWS,. Available from: http://docs.aws.amazon.com/gettingstarted/latest/wah/getting-started-applicationserver.html#create-security-group.