International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 06 | June -2017 www.irjet.net p-ISSN: 2395-0072

Improving Cloud Efficiency using ECDH, AES & Blowfish Algorithms

Rakesh Dogra[1], Anupam Sharma[2]

[1]*Department of Computer Science & Engineering, Desh Bhagat University, Gobindgarh, Punjab*
[2]*Assistant Professor, Department of Computer Science & Engineering, Desh Bhagat University, Gobindgarh, Punjab*

Abstract –*Cloud computing is being increasingly used in all spheres of business and industry but despite the convenience and cost effectiveness of the cloud, there have been concerns about the safety and security of the data stored in the cloud servers. Many people are averse to using cloud services due to the risks involved in handing over their confidential information to an unknown third party. Researchers have come up with several methods to minimize such risks. This research also tries to improve the security as well as the efficiency of cloud storage services through the use of combination of ECDH, AES and Blowfish algorithms. The methodology consisted of generated an ECDH key and using AES and Blowfish algorithms to encrypt data using those algorithms. Cloudsim was used to simulate the cloud environment based on NetBeans IDE and Java was used as the programming language. The results showed a significant improvement in encryption time and storage size using the proposed methodology.*

Key Words: Cloud Computing, ECDH, AES, Blowfish, RSA, Cloud Security, Encryption time

1. INTRODUCTION

One of the issues which cloud computing environment faces these days is the security of the data stored in the cloud servers. Since the users who submit data for storage, specifically over the public clouds which have a collection of data from various sources there is a need to develop tools, techniques and technologies which help to safeguard the confidential information stored in these servers. This study is an attempt to develop one such methodology wherein it has been tried to improve the efficiency of cloud storage through the reduction of storage size and improvement in the time taken for encryption and decryption. The research was based on a combination of the technique of using secret keys generated using elliptical curve diffie hellman algorithm and encypting data using those keys with the help of AES and then Blowfish algorithms. Cloudsim tool was used to provide the platform to generate the cloud server over the NetBeans IDE

1.1 Threats in Cloud Security

Although the cloud services are useful in terms of cost effectiveness and saving the trouble of deploying the necessary hardware and software, there could be issues related to data privacy [1] and integrity [2] in a cloud environment. It is obvious that since the data stored is handled by a third party, namely the cloud service provider, which is usually located in a far off geographical location, it is very difficult for a consumer to find out if the data is be handled securely and safely.

There are two main types of threats to data, from outside attacks and from insiders who could break or sneak into the data. These threats can be further divided into a number of areas [3] [4] [5] as shown below.

Data Breaches

It refers to the loss of data due to leakage or theft and in order to avoid breaches, data is encrypted. However this leads to another problem that in case the encryption key is lost, the data is unreadable and if it is stored in the offline mode then the risk of breach is increased further.

Hijacking

This refers to the loss of data when techniques such as phishing or social engineering are used to gain access to user credentials which in turn are used to misuse data

In-house Threats

One of the greatest threats to data is from insiders with wrong or fraudulent intentions. It is often easier for an insider to gain access to confidential data rather than an external attacker.

Shared Infrastructure

Since the infrastructure and resources are shared, this could lead to security issues in many cases. For example the shared infrastructure such as common storage space could be a source for potential threat to data.

Various techniques and tools have been used over the years to overcome and minimize the above risks and the technique proposed by the authors aims to improve the security and cloud efficiency.

1.2 Cryptography in Cloud Computing

Many users are hesitant to accept cloud computing services due to the fear of their data being lost or stolen [6] and therefore mostly the data stored on cloud servers is in encrypted format. Hence cryptography plays an important role in cloud computing and over the years researchers have been trying various combinations of algorithms [7] [8] [9] which can make the data secure from both internal and external threats. Researchers have been using different combinations of encryption techniques to try and improve the cloud security including

methodologies which split the data before uploading on the cloud [10] so that even if the information is accessed by any insider, it is not comprehensible or useful in any manner [11].

2. TOOLS & TECHNIQUES USED

ECDH or elliptical curve diffie hellman key generation has been used to generate the key for encryption. As the name itself suggests, it is a combination of EC and DH techniques each of which have been explained separately.

2.1 Elliptical Curves

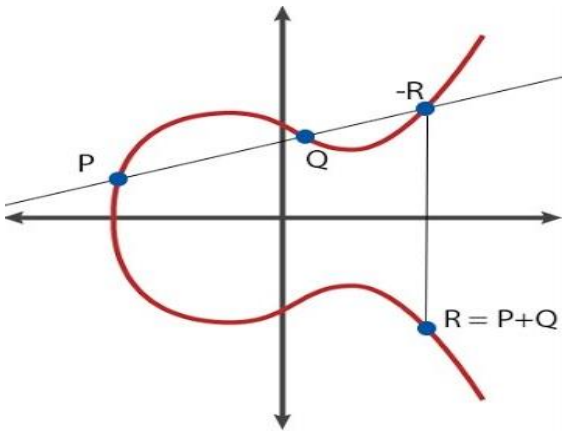

Fig - 1: Elliptic Curve

The above diagram shows a typical elliptical curve which is having a mirror image over the X-axis. An elliptic curve is defined by the equation

$$y^3 = x^3 + ax + b$$

Where a and b are real numbers.

The two points P and Q on the curve can be added by drawing a straight line passing through P and Q. The point of intersection of this line with the curve (-R) is the inverse of the sum of two points. In order to find the sum, the mirror image of –R across the x-axis is found by drawing a line parallel to y-axis.

In the particular case where P = Q, the sum is found by the same procedure by instead a tangent to the curve is drawn at the point. The point R in this case represents 2P or the double of P. Similarly the point 3P can be found by adding P to 2P.

The property of the elliptical curve is that given a point P and number n such that H is the result of n times P or

$$H = n * P$$

It is nearly impossible or practically infeasible to find out N even if H and P are known. This is called as the discrete logarithmic problem of the elliptical curve and is used in cryptography for key generation. The curves generated by professional mathematicians are used for the purposes of encryption and the keys used are normally the x coordinate of the chosen point on the curve.

2.2 Diffie Hellman Algorithm

The EC or elliptical curve is used in conjunction with the diffie hellman algorithm to jointly generate a secure key. The concept of DF is explained with the help of a diagram as follows.

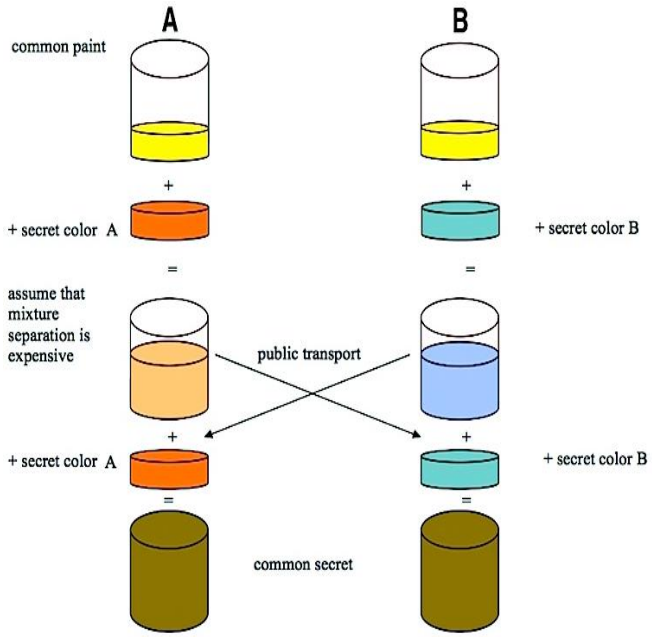

Fig - 2: Diffie Hellman Algorithm Explanation

The above diagram explains the concept of Diffie Hellman algorithm in a simple to understand and intuitive method. It consists of two users A and B who want to arrive at a common colour or a secret common key but they do not want to exchange it over the unsecure medium.

So they choose a common public colour and mix it with their own secret colours, then they pass on the mixture to each other.

Then they add their own respective private colours to this common mixed colour and arrive at a same colour which is not passed through the public media. Hence they are able to have a common shared colour without anyone knowing it.

This is the basic concept diffie hellman algorithm where the secret key is generated by exchanging information over the public network. A generator point is chosen by both

parties and multiplied by their own random numbers which are kept secret. The random multiples are again exchanged over the public network and the multiplied by own secret random numbers to get common secret key which is used to encrypt the communication messages.

2.3 AES Algorithm

The AES algorithm is also known has the Rijndael cipher based on the names of the persons who invented this in the year 1998 during a contest in America. It is a symmetric key algorithm which means that the key used for encryption and decryption is the same key. It has mostly replaced the traditional DES algorithm which was very popular earlier and is now been replaced by AES and is also used for encrypted confidential information used by the United States government.

The algorithm consists of 128 bit block size and the key size varies from 128, 192 or 256 bits and the number of repetitions of the cycle depends on the key size standing at 10, 12 and 14 cycles respectively corresponding to the three key sizes.

2.4 Blowfish Algorithm

The blowfish algorithm was invented by a person named Bruce in the year 1993 at a time when most encryption algorithms were paid and this was developed as a free and effective alternative for the purpose.

Basically it is a block cipher having a size of 4 to 56 bytes of block. The procedure for decryption is the same as for encryption which means to say that it is a symmetric algorithm. Apart from being open source, it also has the advantages of being fast and secure.

2.5 NetBeans IDE

The programming and development done for this project was done using the integrated development environment provided by NetBeans. Mainly there were two choices available for this, namely NetBeans and Eclipse but NetBeans was used because of its more user friendly interface and behaviour. NetBeans has been coded entirely in Java and used as a popular IDE for the same but it can also be used for other languages including but not limited to C++, PHP, HTML5 and so forth.

2.6 Cloudsim Tool

Cloudsim tool was used to simulate the cloud environment. It is an open source tool consisting mainly of Java libraries developed for various tasks which emulate a real cloud environment setup in the virtual mode.

The first step in using Cloudsim is to create a virtual data center which stores the data in the cloud. This is done through running the GUI created for the purpose and filling in the values of the different components as shown below.

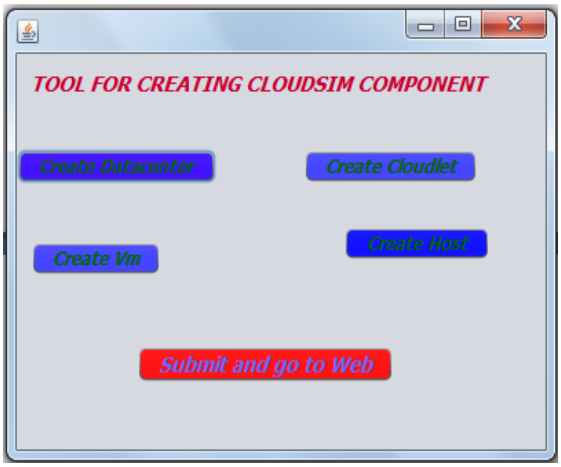

Fig – 3: Cloud Initialization Panel

The user needs to fill in the different details for the datacenter, cloudlet, virtual machine and host in order to use the cloud service simulation.

3.0 EXPERIMENTAL SETUP

Once the Cloudsim tool is up and running, the panel shown in fig 3 opens and the user needs to fill the details by clicking on the components one by one. One of these relating to the data center values has been shown below.

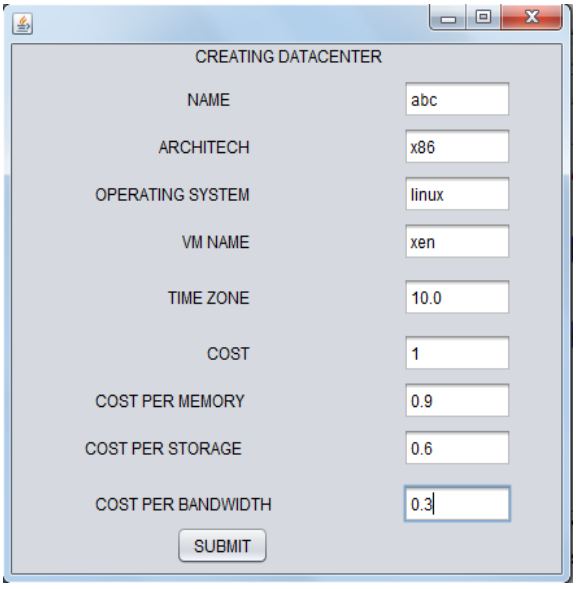

Fig – 4: Data Center Values Entry

After entering all the values for the four components of data center, cloudlet, virtual machine and host, the submit button is pressed. Once the input is received the Cloudsim tool generates the required data center virtually and a success message is shown as the output with the label written as "Datacenter created successfully". Once the OK button is pressed, the window disappears and the web page opens up which can be used for entering and viewing the data in the cloud.

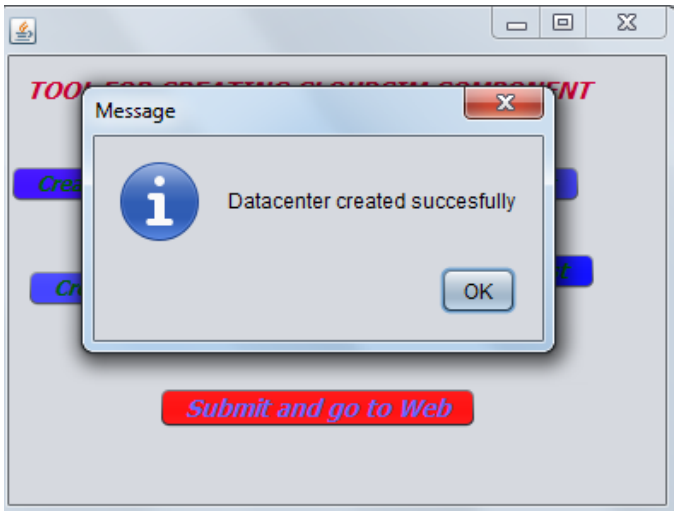

Fig – 5: Data Center Creation Message

3.1 Cloud Data Entry

Once the data center has been created, the webpage opens which asks for new user registration. The URL of the page that opens would be http://localhost:8080/CloudSecurity/welcome.html and page shows the options for home, user login, admin login and new user sign up. The user then has to click on sign up as new user. The data which is sent to the cloud server as the new user information is the one which is encrypted using ECDH keys. The screenshot of the home page is given as follows.

Fig – 6: Home Page for Cloud Server

When the new sign up is clicked the page for user sign up opens. This page along with other JSP and html pages have been created using the WYSIWYG or what you see is what you get editor in the NetBeans IDE using the tool of creating new file. Once the dialog box for the new file opens it has got the options to create a new file related to web, java, html and so forth. The appropriate files are created as the wizard goes ahead and the code can be written on the space created for the purpose within the framework of NetBeans IDE.

The new user registration page which opens after creation of the data center asks for several types of information such as the user name, mobile and so forth. Apart from that the page also shows a list of keys which have been generated using the ECDH algorithm as shown in the screen shot below. The user can choose any key as per choice and the data is encrypted using that key.

Fig – 7: Secure Data Entry

The data which is entered through this web form is encrypted with the strong ECDH key as selected during the sign up process by applying AES and Blowfish algorithms in the same order. The encrypted data can also be seen by logging in to the database where the data has been stored. The URL for the database can be found at http://localhost/phpmyadmin and the screenshot below shows the various entries which have been stored in encrypted form. When the data is fetched back from this database it is again decrypted using the same key and displayed on the webpage.

4.0 RESULTS AND DISCUSSION

The main purpose of the investigation was to find out if there could be improvements over the basic RSA encryption in terms of time and size. In order to do the comparison, a simple program was made which compared the size of the data and its time taken for encryption by manually inserting the value of each parameter of the entries used for cloud computing. The program compared the time for encryption and the size of the output for the

same set of data using the basic RSA algorithm as well as the combination of AES and Blowfish algorithm.

4.1 Results

The values inserted in the program are shown as follows and the time taken by the RSA and proposed algorithm has been tabulated as follows.

Table -1: Time Comparison

Parameter	RSA (milliseconds)	Proposed (milliseconds)
Rakesh	1589	621
9988776655	1219	517
Ludhiana	2092	532
Ricky@gmail.com	1918	586
Ab+	1286	504
550000	1704	533
6/12/2017	1644	530

A graph has been drawn for the above values in Excel which is shown in the chart below.

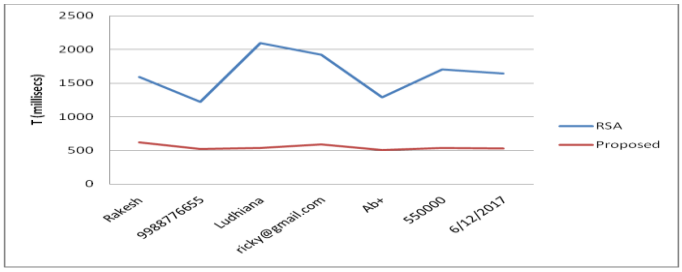

Chart -1: Time Comparison Chart

The values for size of output of RSA and proposed algorithm have been tabulated as follows.

Table -2: Output Size Comparison

Parameter	RSA	Proposed
Rakesh	930	44
9988776655	913	44
Ludhiana	934	44
ricky@gmail.com	925	64
Ab+	924	44
550000	908	44
6/12/2017	946	44

A graph of these values is shown in the chart given below.

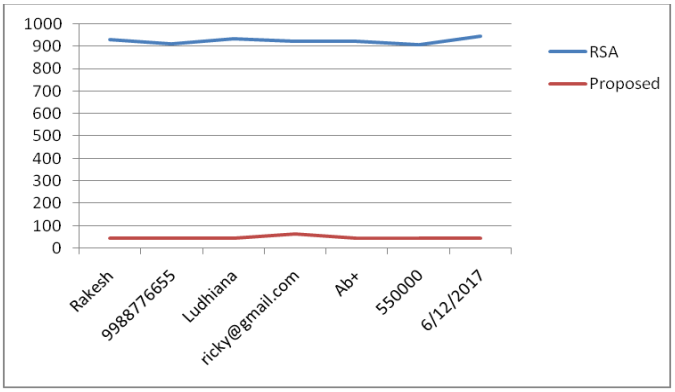

Chart – 2: Size Comparison Chart

4.2 Discussion

As can be seen from the results, there is savings in time as well as the size of the final encrypted output is much smaller than the pure RSA algorithm. This shows that even though the encryption is much stronger in the proposed case since it uses a combination of ECDH key along with AES and Blowfish algorithm, the time taken is relatively lesser.

The same can be seen with the space occupied by the algorithms, RSA is a public key encryption method and there is a limit to the maximum number of bytes that can be encrypted using this algorithm depending on the key size. For example when using a 1024 bit key or 128 bytes key, the maximum number of bytes that can be encrypted at a time are 117 bytes. The mathematical calculation of the number of bits or bytes of the final encrypted text is difficult to estimate as it involves a lot of calculations with exponents, factoring, modulus and primes. The simpler way is to simply measure the actual size of the encrypted text when using a real program as done in the calculations above.

As regards the time, the program calculates the time by measuring the system time when then process of encryption starts both in case of RSA as well as the proposed algorithm and it measures the system time when the process ends. The difference between the two is taken in milliseconds and it gives the time taken for the encryption. These values were then recorded in an Excel sheet and a graph was made which helped to visually identify the difference between the two methods. Hence it can be said that the proposed algorithm certainly gives better results than standalone RSA algorithm.

5.0 CONCLUSIONS

This research was carried out to enhance the efficiency of cloud storage by using a combination of different algorithms such as elliptic curve diffie hellman, advanced encryption standard and blowfish. The main aim of the research was to try and minimize the increase in size as well as the time taken for encryption. The base used for

comparison was the standard RSA algorithm which is a popularly used algorithm for encryption. The experiments were carried out using Cloudsim as a tool for simulating a virtual cloud environment running on the NetBeans IDE. The results were taken based on the output time and encrypted data size of the input data for both RSA and the proposed set of algorithms. It was found that there could be a significant increase in cloud efficiency using these algorithms.

However there are certain limitations on the research like this was done as a part of the Master's thesis and it requires a higher level work of PhD level to go deeper into the significance of these results and further improving the efficiency by comparing the improvements with the work done by other researchers.

ACKNOWLEDGEMENT

The authors would like to thank Er Khusbhoo Bansal, HOD, Department of Computer Science and Engineering, Desh Bhagat University, Punjab and members of the Research Cell, Desh Bhagat University, Punjab, specifically Dr Mrs Sidhu and Dr Rajan Manro for their invaluable guidance and support during this study.

REFERENCES

[1] B. Kaur, "Cloud computing and security issues: A survey,". International Journal of Computer Science Trends and Technology, vol. 3, no. 2, 2015, pp. 168-171.

[2] Y. Li, K. Gai, L. Qiu, M. Qiu and H. Zhao, "Intelligent cryptography approach for secure distributed big data storage in cloud computing," Information Sciences, vol. 8, no. 5, 2016, pp. 1-13.

[3] M. Mamatha and P. Kanchan, "use of digital signature with diffie hellman key exchange and hybrid cryptographic algorithm to enhance data security in cloud computing." International Journal of Scientific and Research Publication, vol. 5, no. 6, 2015, pp. 1-4.

[4] S. Manjula, M. Devi, and R. Swathiya, "Division of data in cloud environment for secure data storage," 2016 IEEE International Conference on Computing Technologies and Intelligent Data Engineering, 2016, pp. 265-269.

[5] R. Manro, T.D.S. Dua, and A.S. Joshi, "Ensures Dynamic access and Secure E-Governance system in Clouds Services – EDSE," International Journal of Applied Engineering Research, vol. 11, no. 1, 2016, pp. 731-737.

[6] P. Mell and T. Grance, "The NIST definition of cloud computing,", Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, 2011, pp. 1-7.

[7] M.A. Nadeem, "Cloud Computing: Security Issues and Challenges," Journal of Wireless Communications, vol. 1, no. 1, 2016, pp. 10-15.

[8] R. Patil and V. Kulkarni, "Hybrid Cryptosystem Approach for secure communication," IOSR Journal of Computer Engineering, vol. 17, no. 1, 2016, pp. 21-24.

[9] J. Cui, Y. Qi, B. Hong and Q. Chen, "Research on Cloud Computing Data Security based on ECDH and ECC," International Journal of Simulation Systems, Science & Technology, Vol. 17, no. 35, 2013, pp. 201-207

[10] M. Singh, G.M. Singh, A. Kumar and S. Bhargava, "Security in Cloud Computing," Open International Journal of Technology Innovations and Research, vol. 7, no. 1, 201a4, pp. 1-9.

[11] N. Tirthani and R. Ganesan, "Data Security in Cloud Architecture Based on Diffie Hellman and Elliptical Curve Cryptography." IACR Cryptology ePrint Archive, 20114, pp. 4-9.

BIOGRAPHIES

Rakesh Dogra is currently studying for M.Tech in Computer Science & Engineering (final semester) from Desh Bhagat University, Punjab. A former marine engineer with nearly a decade of experience on commercial foreign going and coastal vessels in companies such as Selandia (India), Essar (India) and Tschudi & Eitzen (Denmark), has travelled far and wide as a sailor finally to settle on land as an IT consultant in the coming times.

Anupam Sharma is currently working as Assistant Professor in Department of Computer Science and Engineering at Desh Bhagat University Punjab. Her main area of interest is in mobile adhoc networks where she has published research papers related to improving performance of infested networks and reducing jamming effects.