Asian Journal of Information Technology 15 (5): 928-932, 2016
ISSN: 1682-3915
© Medwell Journals, 2016

Ecids: Elliptic Curve Inspired Data Scrambling Method for Securing Data in Hybrid Cloud Using N-Cloud Architecture

K.S. Vijayanand and T. Mala
Department of Information Science and Technology,
College of Engineering Guindy, Anna University, Chennai, India

Abstract: Cloud computing is becoming popular day by day in the computing paradigm and in the business world. Most of the enterprises, business people, academia and even individuals set up private cloud by consolidating their available resources for an optimized and effective usage of the resources. When private cloud find shortage of resources, it is easy and flexible to hire services from public cloud. Such a model is referred as the hybrid cloud. Among the various services offered by cloud, storage as a service is getting popularity since cloud helps its users to store and manipulate huge amount of data in an easy, flexible and pay-per-use manner. Security issues with cloud computing still hinders people to store their sensitive and business critical data in cloud. The commonly used solution for data security is encryption but most of the cases cryptographic methods consumes large amount of time. More than that the size of the data will be increased when it is encrypted. The increase in the size of the data, while encrypting, will increase cost for storing the data. In this study, an Elliptic Curve inspired Data Scrambling (ECiDS) method is proposed to provide data security in the hybrid cloud environment with the help of data splitting and N-Cloud architecture. The ECiDS provides fast and efficient data scrambling mechanism which exploits a two level data dispersal plan to reinforce the security of information. The data is split into multiple chunks and data bytes is each chunk are scrambled and then stored in to N number of CSPs (Cloud Service Provider). The use of N-cloud architecture along with ECiDS makes exceptionally troublesome for the malicious user to get data since the task of reproducing the original data needs more effort than they can endure. The performance of the proposed system is analyzed with various security metrics and found giving better results and the execution time of the proposed algorithm consumes less time than various block cipher encryption algorithms.

Key words: Cloud computing, cloud service providers, data security, data scrambling, elliptic curve, hybrid cloud, N-cloud architecture

INTRODUCTION

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction. The NIST has defined hybrid cloud as “a composition of two or more distinct cloud infrastructures (private, community or public) that remain unique entities but which are bound by standardized or proprietary technology that enables data and application portability”. The intension of Hybrid cloud is to consolidate the advantages of both private cloud and public cloud. Hybrid cloud gives operational adaptability by running mission critical and sensitive information in private cloud and development and testing activities in public cloud. Furthermore, it incorporates a pay-for-your-installment highlight as a feature of its open cloud administrations (Parsi and Laharika, 2013; Rajkumar and Balamurugan, 2014).

These cloud deployment models provide storage as a service as one of their services. Since, huge data is stored in a public CSP, the private data is under the risk of being accessed by unauthorized entities. No Public CSP will guarantee the complete security for the data being stored with them.

The private cloud that uploads their private data never wants any unauthorized entities access the data. Unauthorized entities can be a lethal hacker, a competitor and sometimes the CSP itself (Khan *et al.*, 2013). There are many state of the art mechanisms proposed for data security in cloud computing like data partitioning techniques (Venkataramana and Padmavathamma, 2013), Encryption techniques (Han *et al.*, 1996; Hacigumus *et al.*,

Corresponding Author: K.S. Vijayanand, Department of Information Science and Technology, College of Engineering Guindy, Anna University, Chennai, India

2002; Kamara and Lauter, 2010; Mahajan and Sachdeva, 2013), Homomorphic encryption techniques (Ahmad and Khandekar, 2014; Bajpai and Srivastava, 2014), Multi-cloud architecture (Bohli *et al.*, 2013; Vijayanand and Mala, 2015).

The conventional cryptographic systems are performing well because to break that system an attacker may need information about plain text, cipher text, key, time and computational power. The size of encrypted data will be more than that of the original data which can be an overhead as the payment is based on the size of the data to be stored.

Our research aims to overcome the above scenario where the CSP itself is the attacker. Just because the entire data is encrypted and stored with one CSP, the CSP may get access of the data if the encryption algorithm fails. To avoid this, we use multiple CSPs for storing the data. The data is fragmented into multiple chunks and stored into multiple CSPs.

MATERIALS AND METHODS

Elliptic curves: Elliptic curves are widely used for encryption, digital signatures, pseudo-random generators and other tasks. In this study, the properties of elliptic curve are used for data splitting and data scrambling. Elliptic curves which are not directly related to ellipses are cubic equations in two variables that are similar to the equations used to calculate the length of a curve in the circumference of an ellipse. The general equation for an elliptic curve is (Enge, 1999):

$$y^2 + b_1xy + b_2y = x^3 + a_1x^2 + a_2x + a_3 \qquad (1)$$

Three types of elliptic curves are mainly used in cryptosystems and in this work, elliptic curves over GF (p) is used. Modular arithmetic is used in such type of elliptic curves. The addition operation of the elliptic curve is performed in modulo p. The resulting elliptic curve is expressed as E_p (a, b) where p defines the modulus and a and b are the coefficient of the equation:

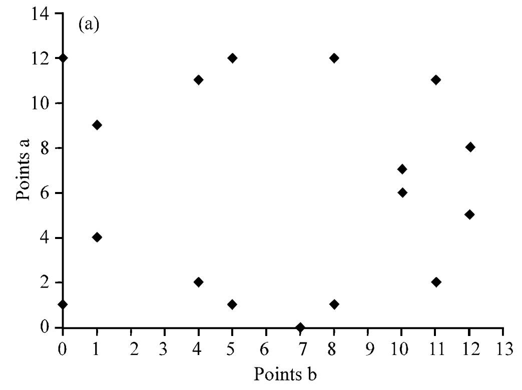
$$y^2 = x^3 + ax + b \qquad (2)$$

Figure 1 shows a table and graph that shows the points generated and plotted for the elliptic curve E_{13} (1, 1):

Uploading the data in to public cloud

Elliptic curve inspired data splitting: The original data in the private cloud is split into multiple chunks using any sequential file splitting program. Each data chunk is placed in the points generated on the elliptic curve. Then, the chunks will be read again in a row-wise manner from the top of the graph towards bottom in left to right fashion and the chunks will be re-arranged to form the original data. Thus, the original data will be scrambled and will not convey the original information to anyone who tries to read it in the re-arranged format. The process is illustrated with the elliptic curve points given in Fig. 1. The elliptic curve selected for the data to be stored is E_{13} (1, 1) and the number of points generated is 17. Hence, the original data is divided in to 17 chunks as shown in Fig. 2.

First chunk will be placed in to the first point, (0,1), the second chunk is placed in point (0,12) and so on. Once all the chunks are placed in the 17 points in the graph, the points (the chunks) are read again in a row-wise manner from the top of the graph towards bottom in left to right fashion. That is the chunk placed in the point (0,12) will be read first and then the chunk in the point (5,12), then in the point (8,12) and so on. Finally, the original data will be scrambled as the data chunks will be



(b)

(0, 1)	(0, 12)
(1, 4)	(1, 9)
(4, 2)	(4, 11)
(5, 1)	(5, 12)
(7, 0)	(7, 0)
(8, 1)	(8, 12)
(10, 6)	(10, 7)
(11, 2)	(11, 11)
(12, 5)	(12, 8)

Fig. 1: Points on elliptic curve E_{13} (1, 1): a) Graph description and b) Their points description

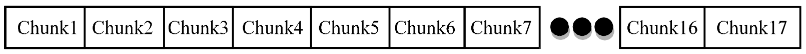


Fig. 2: Data split into chunks

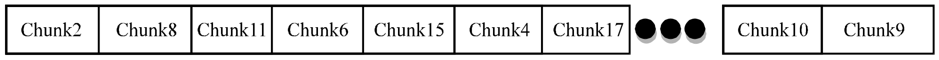


Fig. 3: Scrambled data chunks

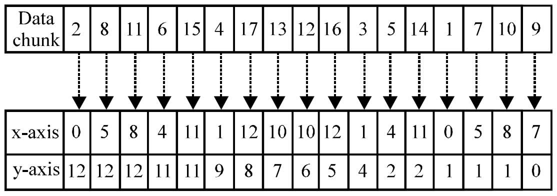
arranged in a different order, as shown in the Fig. 3. This level of scrambling reduces the amount of information that can be retrieved from this form of scrambled data samples.

To increase the level of security, N-cloud architecture is adopted in the public side of this hybrid model. Instead of storing the whole data in a single CSP, the data in the form of multiple chunks are stored in N number of CSPs. The multiple chunks are packed into N number of groups and each group of data chunks will be stored into each CSP. First group of chunks in the scrambled data set, say group1, will be stored in CSP1 and second group of data chunks, say group2, will be stored in CSP2 and Nth group of data chunks will be stored in CSP_N. The number of chunks in the each group will be determined by the total number of the data chunks and the number of CSPs identified for a particular data set.

The number of chunks in a group, n = Round (# data chunks/N, 0) where N is the number of public CSPs. The (N-1)×n number of chunks will be stored in N-1, number of CSPs and the remaining number of chunks will be stored in the Nth CSP. Each group will be stored in each CSP.

Elliptic curve inspired data scrambling: The data scrambling refers to the scrambling of bytes within each data chunks that are re-arranged in the above mentioned manner. The data bytes of each chunk will be scrambled again before uploading into CSPs. Each data chunk will be split into multiple blocks of bytes. The block size is a user defined value which can be 8, 16, 32, 64, 128 or 256. Since, the number of blocks within a chunk is huge number, an elliptic curve E_p (a,b) with a larger p value will be selected for the bytes scrambling process. The scrambling of bytes will be done in the similar fashion as explained above with an elliptic curve of large number of points.

Thus, the two levels of data scrambling before uploading into public CSPs, increase the security of the data. Even the data fragments in the public CSPs are maliciously accessed by an attacker, the chances of retrieving correct information out of it will be very less. In this method, the details that are to be stored in the private cloud are very minimal and they are: elliptic curves E_p (a,b) defined for scrambling of data chunks and scrambling of data bytes within each data chunk. One to one mapping



Data chunk	2	8	11	6	15	4	17	13	12	16	3	5	14	1	7	10	9
x-axis	0	5	8	4	11	1	12	10	10	12	1	4	11	0	5	8	7
y-axis	12	12	12	11	11	9	8	7	6	5	4	2	2	1	1	1	0

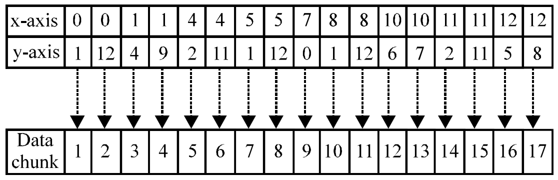
Fig. 4: Mapping between scrambled chunks and the points on the curve

information regarding the groups of data chunks and the CSPs to which the groups are stored (i.e., group1 is stored in CSP1, group2 is stored in CSP2 and so on).

Downloading the data to private cloud: The data chunks that are stored in N number of CSPs will be downloaded into private cloud and based on the information kept in the private cloud, the first group of data chunks which are downloaded from CSP1 will be arranged first which will be followed by the group of data chunks that are downloaded from the CSP2 and this arrangement continues till the group of chunks from the Nth CSP is arranged. Thus, the original data set in the scrambled form is re-created in the private cloud., i.e., the data in the scrambled form as shown in Fig. 3 is re-created in the private cloud. Next task is to unscramble the data bytes within each chunk and then arrange the data chunks in the original order and fuse together to construct the original data set.

The process to unscramble the data bytes within each chunk and to unscramble the data chunks to form the original data set follows the same methods as described below.

The same elliptic curve E_p (a,b) used to scramble the data bytes is drawn again and the points over the curve will be found out. The unscrambling process is explained using the example mentioned above. The points generated for the elliptic curve E_{13} (1,1) are sorted in descending order of the y-coordinate value and then in ascending order of x-coordinate value of each y . Then, the data chunks in the scrambled data set are mapped to the each point in the previously sorted list as show in Fig. 4.



x-axis	0	0	1	1	4	4	5	5	7	8	8	10	10	11	11	12	12
y-axis	1	12	4	9	2	11	1	12	0	1	12	6	7	2	11	5	8
Data chunk	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

Fig. 5: Original data set re-created after sorting the points

To unscramble the data chunks, sort the points in the ascending order of x-values and then in the ascending order of y-values for each x-value. The data chunks that are mapped to each point will also be arranged in the same order that the points are sorted as shown in Fig. 5. Thus, the scrambled data chunks will be unscrambled to form the original data when it is fused together.

RESULTS AND DISCUSSION

Security: Security of the mechanism is ensured from the methods like data fragmentation, bytes scrambling within the fragment and the use of N number of CSPs for storing the data fragments. The use of N number of CSPs makes it impossible for an attacker to get all the data fragments of a data set, in his lifetime. The strength of this security mechanism lies in the ignorance, of an attacker or even a CSP, regarding 'how many CSPs are totally used' and 'which CSPs are used' for storing a particular data set. As explained above, even a CSP is compromised the probability of getting all the fragments belonging to the particular set is very less.

Let M be the number of nodes in a CSP, S be the number of fragments of a single data file stored in that CSP and I be the number of nodes successful intruded and if I>S, then the probability that I number of victim nodes contain all the data fragments represented by P (I,S) is given as:

$$P(I,S) = \frac{C_S^I \times C_{(I-S)}^{(M-1)}}{C_I^M} \tag{3}$$

M = 30; I = 10; S = 7; P (I, S) = 0.004578

Worst case: Let, the n be the number of CSPs selected for storing the private data, from the N number of CSPs altogether available in the market. An attacker need to find out all the n number CSPs among the total number of CSPs available in the market N. The probability that the attacker gets access to all the n number of CSPs in the first n attempts is given as:

$$P(n,N) = \left(\frac{n}{N}\right)\times\left(\frac{n-1}{N-1}\right)\times\left(\frac{n-2}{N-2}\right)\times \left(\frac{n-3}{N-3}\right)\cdots\left(\frac{n-(n-1)}{N-n}\right) \tag{4}$$

For, e.g., If n = 10 and N = 100, then the probability. In worst case, the probability for an attacker gets all the fragments from a CSP, based on the above mentioned scenarios can be given as:

$$P = P(I,S)\times P(n,N) \tag{5}$$

For the given scenario, the probability value will be 2.6446×10^{-16}.

Average case: Let, the attacker could successfully access h number of CSPs among the N number of CSPs altogether available in the market. If the private data is stored in d number of CSPs and if h>d, then the probability that all the d come within the h is given as:

$$P(h,d) = \frac{{}^{h}C_{d}\times{}^{(N-h)}C_{(h-d)}}{{}^{N}C_{h}} \tag{6}$$

For, e.g., if h = 15, d = 10 and N = 100, then the P (h, d) = 3.88×10^{-17} that also a very less value. In both cases as N increases the probability of successful attack will become less. In addition to that, the probabilities discussed above are in the assumption that the attacker found out the information regarding the number of CSPs used for storing the private data. If it is unknown, the difficulty of the attacker increases exponentially. The entropy or the uncertainty of n is given as:

$$\text{Entropy}, H(n) = P(n)\times\log_2(1/P(n)) \tag{7}$$

where, P (n) is the probability of n events to occur, i.e., the probability of selecting n number of CSPs for storing the private data. Since, the original data set is split into multiple number of fragments, each fragment may not carry enough information for an attacker.

Performance: The execution time for the proposed ECiDS algorithm is compared with DES, Triple DES and AES in ECB (Electronic Code Book) mode and CBC (Cipher Block Chaining) mode. The experimental results show the ECiDS consumes very less time compared to the block cipher algorithms in ECB mode and CBC mode. Table 1 shows the values obtained during the experiments conducted to compare the execution time of ECiDS with the block cipher algorithms in EBC mode and CBC mode.

Table 1: Comparison of execution time of ECiDS with block cipher algorithms, in ECB and CFC modes

Data chunk	ECB mode				CBC mode		
size (kb)	DES	3DES	AES	ECiDS	DES	3DES	AES
20	0.67	2.33	1.33	0.36	5.67	19.33	20.67
36	1.33	4.33	2.00	0.66	10.00	33.00	31.33
49	1.67	5.67	2.67	0.89	13.67	43.33	41.67
59	2.33	7.67	3.67	1.07	17.33	60.33	58.00
69	3.00	8.67	4.33	1.26	23.00	67.00	66.67
138	5.67	17.00	8.67	2.51	43.00	133.67	136.33
158	6.67	20.00	10.00	2.88	50.33	157.33	157.67
166	7.00	20.67	10.33	3.02	53.00	162.67	163.00
192	8.00	24.00	12.00	3.49	61.67	189.33	189.00
232	10.00	29.00	14.67	4.22	76.33	227.00	229.00

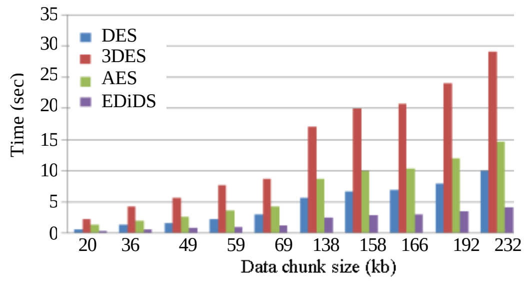


Fig. 6: Execution time results with ECB mode

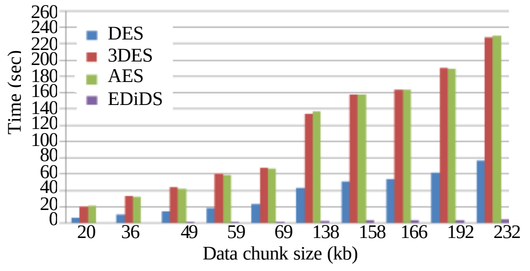


Fig. 7: Execution time results with CBC mode

Figure 6 and 7 show the graphical representation of the values given in Table 1. The results seal the fact that in terms of execution time the ECiDS outperforms the standard block cipher algorithms.

CONCLUSION

In this research, multiple public CSPs are used to spread the risk. Since, the data is split into multiple fragments and stored in multiple CSPs, even a CSP is compromised or even a CSP has become malicious, the chance of data leakage will be very less. ECiDS is used for data chunking and scrambling of data bytes within each chunk which consumes very less time and resources. In addition, the amount of information that need to be stored in the private cloud side are very minimal and less informative. The strength of the security provided by the proposed work is analyzed with cryptographic algorithm metrics like randomness, uncertainty and exposure probability. The time complexity and space complexity of the proposed work is less than the most commonly used cryptographic algorithms (DES and AES). This work has attempted to innovate an out of the box method for providing data security in cloud environment which can be optimized further for better results.

REFERENCES

Ahmad, I. and A. Khandekar, 2014. Homomorphic encryption method applied to cloud computing. Int. J. Inf. Comput. Technol., 4: 1519-1530.

Bajpai, S. and P. Srivastava, 2014. A fully homomorphic encryption implementation on cloud computing. Int. J. Inf. Comput. Technol., 4: 811-816.

Bohli, J.M., N. Gruschka, M. Jensen, L. Iacono and N. Marnau, 2013. Security and privacy enhancing multicloud architectures. IEEE Trans. Dependable Secure Comput., 10: 212-222.

Enge, A., 1999. Elliptic Curves and Their Applications to Cryptography: An Introduction. Springer-Verlag, Berlin, Germany, ISBN-13: 9780792385899, Pages: 164.

Hacigumus, H., B. Iyer and S. Mehrotra, 2002. Providing database as a service. Proceedings of the 18th International Conference on Data Engineering, February 26-March 01, 2002, San Jose, CA., pp: 29-38.

Han, S.J., H.S. Oh and J. Park, 1996. The improved data encryption standard (DES) algorithm. Proceedings of the IEEE 4th International Symposium of Spread Spectrum Techniques and Applications, September 22-25, 1996, Mainz, Germany, 1310-1314.

Kamara, S. and K. Lauter, 2010. Cryptographic Cloud Storage. In: Financial Cryptography and Data Security, Sion, R., R. Curtmola, S. Dietrich, A. Kiayias, J.M. Miret, K. Sako and F. Sebe (Eds.). Springer, New York, pp: 136-149.

Khan, A.N., M.L.M. Kiah, S.U. Khan and S.A. Madani, 2013. Towards secure mobile cloud computing: A survey. Future Gener. Comput. Syst., 29: 1278-1299.

Mahajan, P. and A. Sachdeva, 2013. A Study of encryption algorithms AES, DES and RSA for Security. Global J. Comput. Sci. Technol. Network, Web Secur., 13: 15-22.

Parsi, K. and M. Laharika, 2013. A comparative study of different deployment models in a cloud. Int. J. Adv. Res. Comput. Sci. Software Eng., 3: 511-515.

Rajkumar, B. and K. Balamurugan, 2014. Service and data security for multi cloud environment. Int. J. Innov. Res. Comput. Commun. Eng., 2: 1131-1138.

Venkataramana, K. and M. Padmavathamma, 2013. Multi-tenant data storage security in cloud using data partition encryption technique. Int. J. Sci. Eng. Res., 4: 6-10.

Vijayanand, K.S. and T. Mala, 2015. Securing data in multicloud hybrid model using data splitting and encryption. Int. J. Applied Eng. Res., Vol. 10, No.75.