IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
Vol. 3, Issue 2, February 2016

An Optimized and Secure Data Sharing Approach for Cloud Computing with Proxy Re-Encryption and Chaotic Standard Map

Vishwas Srivastava[1], Sumeet Dhillon[2], Yogendra Kumar Jain[3]

Research Scholar, Computer Science & Engg., Samrat Ashok Technological Institute, Vidisha (M.P.), India [1]

Asst. Professor, Computer Science & Engg., Samrat Ashok Technological Institute, Vidisha (M.P.), India [2]

Head of the Department, Computer Science & Engg., Samrat Ashok Technological Institute, Vidisha (M.P.), India [3]

Abstract: Cloud computing provides an opportunity to scale the computing resources dynamically on the internet. Sharing these resources, in a secure and efficient manner with others, is an open challenge. In order to overcome these issues, we proposed an optimized and secure data sharing (OSDS) approach for data outsourcing onto the cloud. In our proposed work, we used chaotic standard map with proxy re-encryption to provide fast and easy data sharing. Chaotic standard map reduce the computational time cost of encryption and decryption, while proxy re-encryption (PRE) reduce heavy computation at the side of data holder and make the data sharing flexible and easy. Earlier, to preserve the confidentiality of user's data against cloud server, generally cryptographic methods are applied. However, these cryptographic methods possess computation overhead on the data holder for key distribution and data management. An existing approach named as EFADS, provide good solution for this but its computational time cost is more as compared to our proposed OSDS approach. We compared proposed OSDS method with EFADS data sharing approach and found that our proposed approach is 1.5 times efficient than EFADS. Our OSDS approach consumes approximately 66% less computational time cost as compared to previous approach.

Keywords: Chaotic Standard Map, Cloud computing, EFADS, OSDS, Proxy Re-Encryption.

I. INTRODUCTION

Cloud computing is a solely internet dependent technology and growing very rapidly. It reduces the cost and management of hardware and software resources. In cloud computing different systems are connected with each other in a complex way to share computing resource on the policy of pay and use. Cloud computing offer better services for management of software and hardware; however it is more important to secure the data which are going to outsource to the cloud. To preserve the confidentiality of user's data, data should be encrypted before outsourcing onto the cloud. In general data holder encrypt the data for a particular data sharer by using data sharer's public key and send it to the cloud server. Data sharer downloads the data from cloud server and decrypts it by using his private key. If data holder wants to share that data with other sharer, he has to download the data again from cloud and encrypt for other sharer and upload it again, or if data is not generated by himself but received from others, which is encrypted by data holder's public key; data holder has to decrypt and encrypt it again for other sharer. To minimize the work load of decryption and encryption over and over, and make the data sharing flexible and easy we used proxy re-encryption (PRE). Concept of proxy re-encryption (PRE) is that, a third party can change cipher text which is encrypted for one party so that another party can decrypt it by using his own private key as said Blaze et al.[1].

In above case data holder does not require to decrypt and encrypt the data for other sharers. Data holder generate re-encryption key by computing $REKey_{gen} \leftarrow KeyGen(PrKey_H , PuKey_S)$ where $REKey_{gen}$ is re-encryption key, $PrKey_H$ is data holder's private key and $PuKey_S$ is data sharer's public key and send it to the cloud. Cloud server will decrypt the already decrypted data for other data sharer by using re-encryption key.

For optimizing the computational time complexity in encryption and decryption, chaotic standard map is used. Chaotic maps are much appropriate for cryptosystem, it is easy to implement on personal computer. It is fast and have low cost, which make it better than the traditional cryptosystem. In chaotic cryptosystem a key set is generated by a chaotic map and encrypt the data bit by bit. This chaotic cryptosystem used repetition of chaotic map many times to make encrypted data more randomized.

Standard map is described by

$$x_{k+1} = (x_k + y_k + r_x + r_y) mod\ N,$$

$$y_{k+1} = \left(y_k + r_y + K_C sin\frac{x_{k+1}}{2\pi}\right) mod\ N,$$

where (x_k , y_k) and (x_{k+1} , y_{k+1}) is the original and permutated position of $N \times N$ matrix, (r_x , r_y) random scan couple and K_C is positive integer.

A. Paper Organization

We organized our paper as follows. In section 2, we present the related work and difficulty of achieving optimal time in encryption and decryption. Section 3; describe the preliminaries definitions of PRE and ECC

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 3, Issue 2, February 2016

elliptic curve cryptography. In section 4, we proposed our OSDS approach. In section 5, we describe the performance evaluation. In section 6, we give the conclusion of our proposed approach.

II. RELATED WORK

Our proposed OSDS approach fills the area of "proxy re-encryption" and "chaos based cryptosystem". Here we review some research works, which are related to our proposed approach.

Proxy re-encryption: In recent years, cryptographic file sharing system is good area in the research field. There are many researchers has presented the work [2-7] till now. The concept of Proxy Re-Encryption (PRE) is given by Blaze et al. [1]. They introduced a general way in which an intermediate one between the participants of a two party might transform the data without "harming" the protocol and the scheme is only bidirectional. Ivan and Dodis [8] introduced simple definitions of the bidirectional and unidirectional PRE, and implemented the functions based on cryptosystem primitives such as IBE; their formal demonstrating of proxy cryptography significantly generalizes, simplifies and concurrently illuminates the model of "atomic proxy" proposed by Blaze.

Later on, Ateniese et al. proposed the first unidirectional PRE scheme and demonstrated some practical applications, with key-private PRE construction and demonstrate its CPA-security with simple Decisional Bilinear Diffie Hellman assumption and its key-privacy under the Decision Linear assumption in the standard model. In 2007, Canetti and Hohenberger [9] presented the CCA security model for PRE, and demonstrated the first IND-CCA2 secure bidirectional PRE. Tang and Weng solely presented the definition of Type-based PRE and conditional PRE in Tang et al., 2008 [10] and Weng et al., 2009[11] respectively. A lot of research work has been carried out [12-19] on different PRE schemes with different security properties. The proxy re-encryption with anonymity is demonstrated by Ateniese et al. [16]. Later on, Shao et al. [20 and 21] enhanced the idea of anonymity in [16], and presented diverse anonymous PRE scheme. However, all of the existing anonymous PRE schemes need the time-consuming operation–pairings.

However, the proxy in all these scheme provide the facility of transforming all the delegator's original cipher text and enable the decryption rights to delegatee, on the other hand "a proxy can transform the cipher text which is encrypted for one party so that another one can decrypt it by using his own secrete key."

Chaos based cryptosystem: Designing a chaos based cryptosystem has become a hot area for researchers. In recent years many researchers contributed lot of work on chaos based cryptosystem. Lian et al. [22] introduced a block cipher based on the chaotic standard map, which have composition of three parts: a confusion process based on chaotic standard map, a diffusion process, and a key generator. Fridrich introduced Symmetric Ciphers Based on Two-dimensional Chaotic Maps, and suggested that a chaos-based image encryption scheme should compose of two processes: chaotic confusion and pixel diffusion [23]. Later on, Desai et al. proposed a Chaos-Based System for image encryption [24]; it provides encryption and decryption of image of any type, size and shape.

In the literature so far, an approach proposed by Wei et al. [25] (EFADS protocol) are representative. EFADS protocol is a PRE-based data sharing protocol that is motivation of our proposed OSDS approach. However, EFADS protocol is not optimized, i.e., It consumes more time in encryption and decryption. Furthermore, The AFGH05 [4] and YWRL10 [7] protocol are motivation of EFADS protocol. AFGH05 perform time-consuming operation–pairing. YWRL10 is attribute-based encryption (ABE) and slow re-encryption protocol. However, YWRL10 protocol is not flexible. Because the data holder should know the characteristics values of intended data sharer's before encrypting the shared data (i.e., the pre-decided sharer list is required). It is not good that the data holder can share the data only generated by him. EFADS protocol provide flexibility of sharing data i.e. data holder can shared data with sharer even the data is not generated but receive from others without disclosing his own private key. At last, their scheme is not efficient as expected due to the requirement of pairings.

III. PRELIMINARIES

This section describes some basic knowledge about definitions of Proxy re-encryption and Elliptic curve cryptography which are used in our paper.

A. Proxy re-encryption

PRE a unidirectional proxy re-encryption scheme has following probabilistic polynomial time algorithm.

- (PuKey, PrKey) ← Setup (p, q, n).On input security parameter, this algorithm outputs a public key and private key.
- REKey ← KeyGen ($PrKey_H$, $PuKey_s$). On input a private key $PrKey_H$ and public key $PuKey_s$, the key generation algorithm outputs a re-encryption key REKey.
- C ← PRE_Enc (PuKey, M). On input a plaintext M and symmetric key K, the symmetric encryption algorithm outputs a ciphertext C for plaintext M.
- K' ← PRE_ReEnc (K, REKey). On input a key K and re-encryption key REKey, the REnc algorithm generate ciphertext K' for key K.
- M ← PRE_Dec ($PrKey_S$, C). On input a ciphertext and key K, the symmetric key decryption algorithm outputs original plaintext M.

Correctness

The correctness property required following conditions.

PRE_Dec (PrKey, PRE_Enc (PuKey, M)) = M, and

PRE_Dec (PrKey, PRE_ReEnc (KeyGen ($PrKey_H$, $PuKey_s$), C)) = M,

Where C is ciphertext for plaintext M under PuKey from PRE_Enc.

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 3, Issue 2, February 2016

B. Elliptic Curve Cryptography

ECC is a public-key cryptography based on the algebraic structure of elliptic curves over finite fields. One of the main advantage of ECC cryptography is that, it provide the same level of security with smaller key non-ECC cryptography with larger key (with plain Galois fields as a basis). The use of elliptic curves in cryptography was suggested independently by Koblitz and Miller [26] in 1985. Elliptic curve cryptography algorithms entered wide use in 2004 to 2005 [27].

Elliptic Curve

An elliptic group E_p (a, b) is determined by following computations

$$x^3 + ax + b \bmod p$$
[for $0 \leq x < p$, a and $b < p$.

Where a and b are integers and p is a prime number and must satisfy the following condition.

$$4a^3 + 27b^2 \bmod p \neq 0$$

For each value of x, we are required to find that whether it is in quadratic residue or not. If it is in quadratic residue then elliptic group have two values else the point is not in the $E_p(a, b)$.

Now one need to obtained quadratic residue by computing $x^2 \bmod p$ and $(p-x)^2 \bmod p$ therefore the quadratic residue is Q_p = set of $\frac{p-1}{2}$.

Operation over elliptic

Let the points P = (x_1, y_1) and Q = $(x_2, y_2) \in E_p(a, b)$ and O is point on infinity. The rules over $E_p(a, b)$ are:

1. P + O = O + P = P
2. If P = (x_1, x_2) and Q = $(x_1, -y_1)$ = -P, then P + Q =O
3. If Q ≠ -P, then P + Q = (x_3, y_3)

$$x_3 = \lambda^2 - x_1 - x_2 \bmod p$$
$$y_3 = \lambda(x_1 - x_3) - y_1 \bmod p$$

where

$$\lambda \triangleq \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & if\ P \neq Q \\ \frac{3x_1^2 + a}{2y_1} & if\ P = Q \end{cases}$$

Elliptic Curve Cryptography

It can be used to encrypt plaintext P, into ciphertext C by encoding plaintext P into a point P_M from the elliptic group $E_p(a, b)$. Then choose a generator point, $G \in E_p(a, b)$, such that smallest value of n for which nG = O is a very large prime number. Then make $E_p(a, b)$ and G public.

User select a private key, $PrKey_H < n$ and determine the public key $PuKey_H = n_A G$. To encrypt the message point P_M for data sharer, data holder choose a random integer k and encrypt the message and get the ciphertext points part P_C using data sharer's public key PuKey.

$$P_C = [kG, (P_M + k\ PuKey_S)]$$

At receiver side data sharer got pair of points, P_C , then data sharer calculate the following

$$(P_M + k\ PuKey_S) - [PrKey_S(kG)] = (P_M + k\ PrKey_S G) - [PrKey_S(kG)] = P_M$$

By computing the above equation data sharer got the plaintext information P_M, corresponding to message M.

IV. PROPOSED SCHEME

In our proposed OSDS scheme, an optimized and secure data sharing approach for cloud computing with proxy re-encryption and chaotic standard map.

The developed approach provides a method for fast data encryption and decryption. For chaotic based cryptography we transform our data into a matrix form by using 2D array. Place the text data into tabular form or matrix form so that each array element is a character. And then transform the table or matrix by using chaotic map function, with random couple coordinates and specified key.

We determine random couple(r_x, r_y), by calculating cumulative sum of

$$r_x = rand\left(sum\left(key(1, 1{:}5)\right)\right),$$
$$r_y = rand\left(sum\left(key(1, 6{:}10)\right)\right)$$

Now determine the new coordinates for some previous and original coordinates by computing the following

$$x_1 = mod(i + r_x + r_y + j, N),$$
$$y_1 = mod\left(j + r_y + 1 * sin\left(\frac{x_1 * 256}{2\pi}\right), N\right)$$

where x_1, y_1 are the new generated coordinates,
i, j are the previous and original coordinates,
and N is the size of matrix.

We have modified matrix or encrypted matrix with a specified key. The decryption process is inverse of encryption and equally simple for those who hold the key.

At description level

1. To convert the data into 2D matrix
   n_1 = count all character of data or message including space.
2. Check whether n_1 is a perfect square, if not choose a smallest integer value m, which is greater than n_1. m must be a perfect square.
3. Initialize n = sqrt(m).
4. Create a matrix of n × n order.
5. Put the value of n_1, character by character into n × n matrix.
6. If there are empty cells at the end of the matrix put the space in that empty cell.
7. Apply chaotic standard map method on the matrix and obtained a new modified or encrypted matrix.

To decrypt the encrypted matrix, inverse this process. Then read the character from the matrix and put that into a sequential data form.

At algorithm level

1. Data = plaintext;
2. N_1 = count data;
3. Initialize integer temp & N = 0;

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 3, Issue 2, February 2016

4. Repeat
   {
   temp = sqrt (N_1);

   if temp * temp == N_1
   {
   N = temp;
   exit loop and go to step 5;
   }
   else
   N_1++;
   }
5. Mat [N]×[N] = Data;

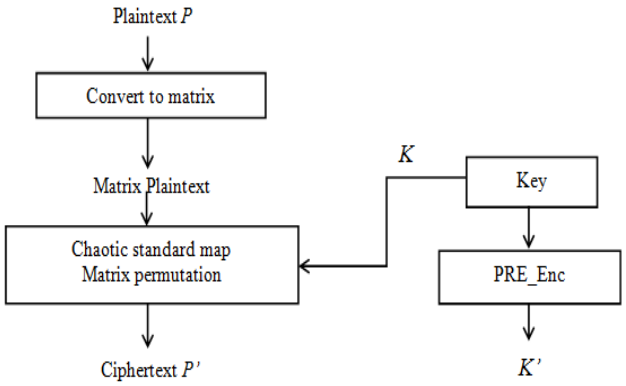

Fig. 4.1 Encryption process by chaos based cryptosystem.

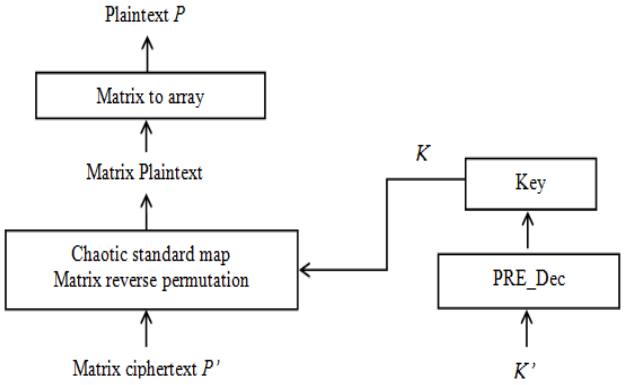

Fig. 4.2 Decryption process by chaos based cryptosystem.

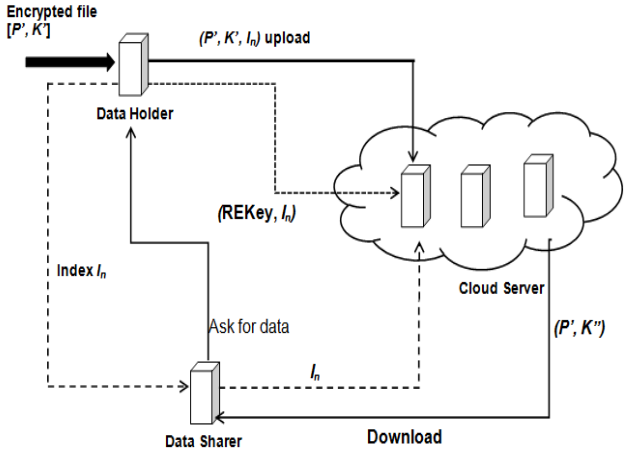

Fig. 4.3 Describe the process of PRE in our proposed Method.

where ***P'*** is encrypted plaintext,
K' is encrypted key,
K" is re-encrypted key,
I_n is index number related to corresponding data
and
REKey is the re-encryption key.

- Here data holder upload the encrypted file [P', K', I_n] (encrypted by executing P' ← Chaos_Enc (K, P).) and index number related to corresponding encrypted file to the cloud server.
- Then data holder send the re-encryption key (REKey, I_n) with index number to cloud.
- If data sharer wants the data from data holder, data sharer ask for data to data holder, then data holder send the index number (I_n) for that particular data.
- Data sharer send the index number (I_n) to the cloud server after that cloud server find the associated data (P', K', I_n) and re-encryption key REKey.
- Cloud server transform the encrypted data file (P', K') into (P', K"), and send re-encrypted data file (P', K") to that perticular data sharer.
- At last data sharer determine the key K with his private key $PrKey_S$ by executing PRE decryption algorithm, K ← PRE_Dec ($PrKey_S$, K").
- Then obtain the original data content P (plaintext) by executing M ← Chaos_Dec (K, P').

V. PERFORMANCE EVALUATION

A cloud computing system has to resolve diverse hurdles security measure, reliability and many more. Sometimes unexpected behaviour of demands and other various issues which are consider that why experiments cannot be done on the real computing environment, if it is done then it will cost to the customer.

By using MATLAB 2012b, we implemented our OSDS approach and EFADS protocol (simple with only time complexity) on a personal computer Dell Inspiron N4010 equipped with the Intel(R) Core(TM) i3 CPU M350 at frequency of 2.27 GHz that is runs on operating system Windows 7 Home Basic.

Based on the above tentative outcomes, we compare our proposed OSDS approach with EFADS approach in terms of computational cost in encryption and decryption and results are plotted in Fig. 5.1 and 5.2.

For evaluating performance, we applied both previous EFADS and our proposed OSDS approach on following number of data, and found that our proposed approach provide optimal cost as compared to existing EFADS protocol.

In below table we evaluate the performance of individual approach applying on some information message or data message.

On the basis of these evaluations we compare our proposed OSDS approach with EFADS protocol and found that computational time cost of our OSDS approach is less as compared to EFADS protocol.

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 3, Issue 2, February 2016

Table 5.1 The experimental results of computational time of EFADS protocol.

S.No.	Plaintext	Ciphertext	Encryption time (in sec)	Decryption time (in sec)
Data1.	PROXY RE-ENCRYPT	a˘¤ ìmÌPòPN g	0.0086194	0.0084704
Data2.	SATI	Qy	0.0069089	0.0069129
Data3.	Chaotic Standard Map	Â¦¸.¢,$ • ¿d- t	0.0093142	0.0093776
Data4.	Elliptic Curve Cryptography	ÉÓ ZlÒÚ÷ ö t mHfZÆeáÈ d rl	0.010236	0.010219
Data5.	RSA Cryptography	d fe & m r # 5 5	0.0088101	0.0087797

Table 5.2 The experimental results of computational time of our proposed OSDS approach.

S.No.	Plaintext	Ciphertext	Encryption time (in sec)	Decryption time (in sec)
Data1.	PROXY RE-ENCRYPT	È°í ü¸dX •Q fi	0.0035679	0.0034927
Data2.	SATI	¥ ,	0.0016811	0.0019248
Data3.	Chaotic Standard Map	Øã 4 3-I ëSÅ'Í	0.0042273	0.004225
Data4.	Elliptic Curve Cryptography	y<C$eÍ À æ×ÇÞ)à Á Q? •	0.0052114	0.0053364
Data5.	RSA Cryptography	M± í°TPE¸Q•±a ÅÁ	0.0037513	0.0036426

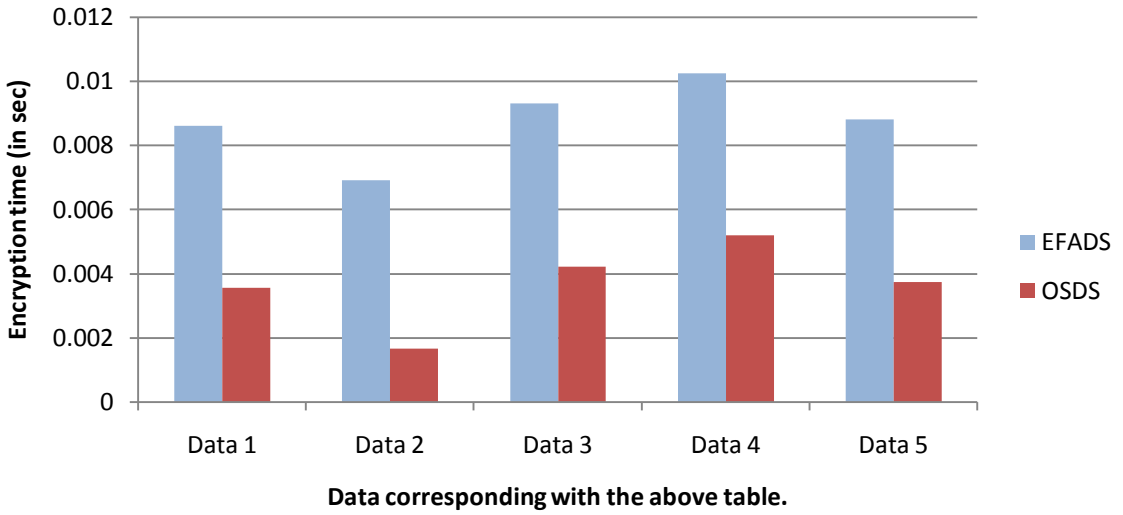

Fig. 5.1 Computational cost in encryption between our proposed OSDS approach and EFADS protocol.

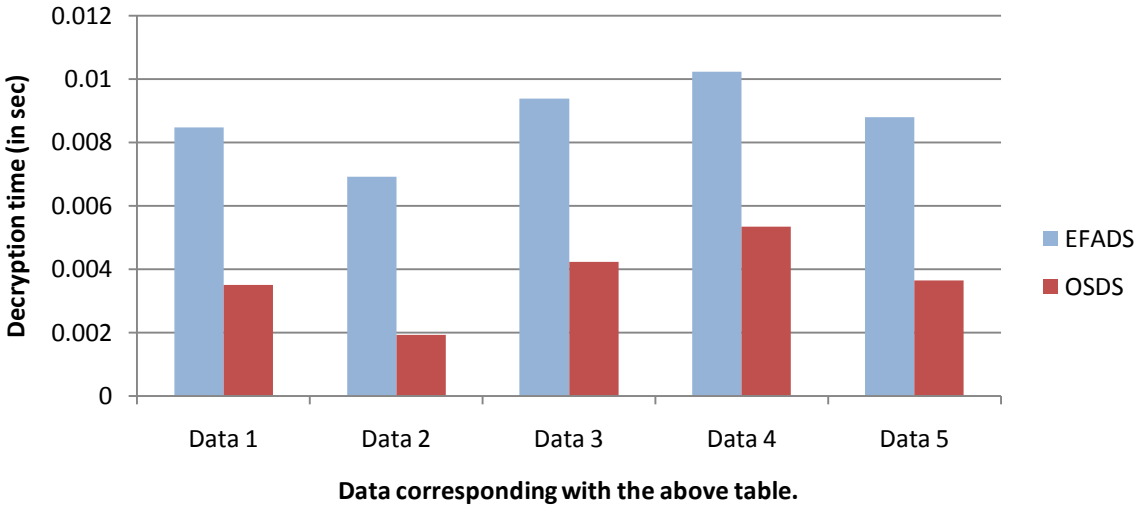

Fig. 5.2 Computational cost in decryption between our proposed OSDS approach and EFADS protocol.

For obtaining computational time cost in cryptography of particular data with proposed algorithm discussed in section 4. First we converted the plaintext data into two dimensional arrays to apply chaotic standard map function and got ciphertext by using encryption keys, for decryption we applied chaotic standard map in bottom up manner. We calculate the computational time cost for encryption and decryption. The whole process consume less time as compared to traditional cryptosystem, even length of data is more, reason is that it does not require to perform XOR operation with 128 bits key with many iterations.

As shown in Table 5.1 and 5.2, we computed the encryption time and decryption time (in seconds) for particular sets of data with proposed OSDS approach and EFADS approach. EFADS approach generates some ciphertext "a˘¤ìmÌPòPN g" for the plaintext "PROXY RE-ENCRYPT" in 0.0086194 seconds. Decryption process takes 0.0084704 seconds to decrypt the ciphertext in original plaintext. Whereas proposed OSDS approach encrypt that particular plaintext into ciphertext "È°í ü¸dX •Q fi" in 0.0035679 seconds only.

From the above analysis, we can observe that our proposed OSDS approach is truly flexible and efficient. The reason is that our Proposed OSDS approach does not require time consuming cryptographic computation.

VI. CONCLUSION

In this paper, we revisit the concept of EFADS proposed by Wei et al. (2014). In EFADS their primitive allows user to share their data with others in a secure and flexible manner. However, its computational time cost is more. Thus we introduced an optimized and secure data sharing (OSDS) approach for cloud computing by using proxy re-encryption (PRE) and chaotic standard map. Our proposed OSDS approach provides the optimal solution for computational time cost for sharing data with secure and flexible style. Proposed chaos based OSDS approach has some distinct properties. The plaintext size of OSDS approach is not fixed. As if the length of the plaintext increases, difficult to breach the security. The proposed OSDS technique also works well with heterogeneous type of data. The performance evaluation, in which we compared, proposed OSDS approach with EFADS approach and found that our proposed approach is 1.5 times efficient than EFADS.

REFERENCES

[1] Blaze M., Bleumer, G., Strauss M, "Divertible protocols and atomic proxy cryptography" In: EUROCRYPT 1998., LNCS, vol. 1403, pp. 124-144, (1998).
[2] Matt Blaze, "A cryptographic file system for UNIX" ACM Conference on Computer and Communications Security, ACM, pp. 9-16, (1993).
[3] Adya A., William J. Bolosky, Castro M., Cermak G., Chaiken R., John R. Douceur, Howell J., Jacob R. Lorch, Theimer M., Wattenho fer R., "FARSITE: Federated, Available, and Reliable Storage for an Incompletely Trusted Environment" In: Proceedings of the 5th Symposium on Operating Systems Design and Implementation, OSDI, pp. 1-14, (2002).
[4] Ateniese G., Fu K., Green M., Hohenberger S., "Improved proxy re-encryption schemes with applications to secure distributed storage" In: ACM NDSS 2005, pp. 29–43, (2005).
[5] Eu-Jin Goh, Shacham H., Modadugu N., Boneh D., "SiriUS: securing remote untrusted storage " In: NDSS, The Internet Society, (2003).
[6] Kallahalla M., Riedel E., Swaminathan R., Wang Q., Fu K., "Plutus: scalable secure file sharing on untrusted storage" In: FAST, USENIX, (2003).
[7] Shucheng Yu, Wang C., Ren K., Lou W., "Achieving secure, scalable, and fine-grained data access control in cloud computing" In: INFOCOM, IEEE Press, pp. 534-542, (2010).
[8] Ivan A., Dodis Y., "Proxy cryptography revisited" In: NDSS, The Internet Society, (2003).
[9] Canetti R., Hohenberger S., "Chosen-ciphertext secure proxy re-encryption" In: Proceedings of the 14th ACM Conference on Computer and Communications Security, CCS' 07, ACM, New York, NY, USA, pp. 185-194, (2007).
[10] Tang, Q., "Type-based proxy re-encryption and its construction" In: INDOCRYPT 2008. LNCS, vol. 5365, pp. 130-144, (2008).
[11] Weng, J., Deng, R.H., Chu, C., Ding, X., Lai, J., "Conditional proxy re-encryption secure against chosen-ciphertext attack" In: ACM ASIACCS 2009, pp. 322-332, (2009).
[12] Hohenberger S, Rothblum G. N., Shelat A., Vaikuntanathan V., "Securely obfuscating re-encryption" In: Salil P. Vadhan (Ed.), TCC, LNCS, Springer, vol. 4392, pp. 233-252, (2007).
[13] Libert B., Vergnaud D., "Unidirectional chosen-ciphertext secure proxy re-encryption" In: Ronald Cramer (Ed.), PublicKey Cryptography, LNCS, Springer, vol. 4939, pp. 360-379, (2008).
[14] Shao J., Cao Z., "Cca-secure proxy re-encryption without pairings" In: Stanislaw Jarecki, Gene Tsudik (Eds.), PublicKey Cryptography, LNCS, Springer, vol. 5443, pp.357-376, (2009).
[15] Chu C. K., Weng J., Sherman S. M. Chow, Zhou J., Robert H. Deng, "Conditional proxy broad cast re-encryption" In: Colin Boyd, Juan Manuel González Nieto (Eds.), Information Security and Privacy, 14th Australasian Conference, ACISP, LNCS., Springer, vol. 5594, pp. 327-342, (2009).
[16] Ateniese G., Benson K., Hohenberger S., "Key-private proxy re-encryption" In: Marc Fischlin (Ed.), Topicsin Cryptology, CT-RSA 2009, The Cryptographers' Track at the RSA Conference, LNCS, Springer, vol. 5473 pp. 279-294, (2009).
[17] Shao J., Cao Z., Liu P., "Cca-secure pre scheme without random oracles" J. Cryptol., 112, (2010).
[18] Matsuda T., Nishimaki R., Tanaka K., "Cca proxy re-encryption without bilinear maps in the standard model" In: Phong Q. Nguyen, David Pointcheval (Eds.), Public Key Cryptography, LNCS, Springer, vol. 6056, pp. 261-278, (2010).
[19] Weng J., Chen M. R., Yang Y., Deng R. H., Chen K., Bao B., "Cca-secure unidirectional proxy re-encryption in the adaptive corruption model without random oracles" Sci. China, Ser. F53 (3) , pp. 593–606, (2010).
[20] Shao J., Liu P., Wei G., Ling Y., "Anonymous proxy re-encryption" J. Secur. Commun. Netw. (5), pp. 439-449, (2012).
[21] Shao J., Liu P., Zhou Y., "Achieving key privacy without losing CCA security in proxy re-encryption" J. Syst. Softw. 85 (3), pp. 655-665, (2012).
[22] Lian SG, Sun J, Wang Z., "A block cipher based on a suitable use of chaotic standard map" Chaos, Solitons and Fractals;26(1):117-29, (2005).
[23] Fridrich J., "Symmetric Ciphers Based on Two-dimensional Chaotic Maps" Int. J. Bifurcat Chaos ;8(6): 1259-84, (1998).
[24] Desai D., Prasad A., Carsto J., "Chaos-Based System for Image Encryption" International Journal of Computer Science and Information Technologies, vol. 3 (4), pp. 4809-4811, (2012).
[25] Wei G., Lu R., Shao J. "EFADS: Efficient, flexible and anonymous data sharing protocol for cloud computing with proxy re-encryption" Journal of Computer and System Sciences 80, pp. 1549-1562, 2014.
[26] Miller V. "Use of elliptic curves in cryptography". CRYPTO. LNCS 85: pp. 417–426, (1985).
[27] https://en.wikipedia.org/wiki/Elliptic_curve_cryptography#cite_note-5.

BIOGRAPHY

Vishwas Srivastava received the B.Eng. degree in Computer Science and Engineering from Rajiv Gandhi Proudyogiki Vishwavidyalaya Bhopal, India. He is currently a M. Tech. candidate in the Department of Computer Science, Samrat Ashok Technological Institute Vidisha, India. His research interest is in security in cloud computing.