Cloud Data Assured Deletion Based on Information Hiding and Secondary Encryption with Chaos Sequence

Yijie Chen and Wenbin Yao(✉)

Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing 100876, China
yaowenbin_cdc@163.com

Abstract. The strategic point of data assured deletion in cloud storage is how to avoid unauthorized users accessing or backing up data. The main choice for this issue is how to improve the data access strategy or how to manage the encryption keys better. However, these methods are still put whole encrypted data in the cloud. In case that the encryption key is pick up by an attacker, the data is not secure. A new scheme named AD-IHSE is proposed in this paper. In this method, the bit stream of encrypted data will be divided into two parts. Big part of encrypted data is uploaded to the cloud directly, while the small part of encrypted data is embedded in the carrier and then uploaded to the cloud after secondary encryption with logistic chaotic sequence encryption. This method can even guarantee the security of the data when the encryption key is lost. In addition, the scheme does not bring other third parties to lessen the risk of data leakage. Both safety certificates and experimental results show that it not only realizes the certainty of the cloud data assured deletion but also has good security.

Keywords: Data assured deletion · Bit stream extraction
Data hiding · Secondary encryption · Logistic chaotic mapping

1 Introduction

In recent years, cloud storage has brought opportunities in improving information technology efficiency costs and green computing, as well as some security challenges. In cloud storage mode, data is stored in a third-party cloud storage platform. Data owners lose direct control of their personal data. The security of data highly depends on the cloud service provider (CSP). Data assured deletion is one of those security issues. CSP can not guarantee faithful deletion of data user's data when he received this request. When a data owner intends to delete the data, he wants his data can not be accessed forever once the request is sent out. Considering that CSP is not credible, the most direct way to protect the confidentiality of data is to encrypt the data first, and then outsource the encrypted data to the cloud. The encryption key will be protected by data owner itself. Due to the unreliability of CSP, the data stored in the cloud is at risk of being leaked. For example, for the sake of improve the dependability of the service,

I. Romdhani et al. (Eds.): CollaborateCom 2017, LNICST 252, pp. 613–623, 2018.
https://doi.org/10.1007/978-3-030-00916-8_56

CSP may have a lot of data backups and store them onto several servers. In this case, once the data is expired and the data owner commands CSP to make its data deleted, CSP may not completely delete all of the data and the backup. In the case that an attacker uses an encrypted key and encrypted data that is not deleted from CSP through a brute force attack, the security of the data will be influenced. While the data owner wants to delete the data, if CSP does not execute the actual data deletions, he is not sure that the deleted data will not appear in the future. This is what this article will discuss.

2 Related Works

In this section, we have proposed several proposed solutions to solve the issue of data assured deletion.

The previous work [1] completed a file assured delete system that can restore encrypted data at the appropriate time to destroy the encryption key. But they only bring up a conceptual method that has not been implemented, and their theoretical models are not very convincing. In a cloud environment, the ownership and management of the data are separated. For the sake of the security of data, it must be encrypted before it is outsourced to the cloud. To some extent, this method makes the issue of data assured deletion into another issue, that is, how to delete the encryption key. However, this is a risk that an attacker may get an encryption key through cryptanalysis or brute force. The scheme in [2] points out a kind of strategy based on the approach. The key idea is that the data is encrypted with the data encryption key at the very beginning, and then encrypts the data encryption key with the control key related to the policy. The last but not the least, deleting the control key in the file is equal to the deletion of the data. Although, the control key used in [2] is managed by third party, namely centralized management. There is a security risk. The untrusted key manager may delete or abandon the control key. Distributed management obtains higher security than key centralized management [3].

In addition, document [4] points out that if the scheme destroys the encryption key only without encrypting data, there will be potential security risks, which may cause encrypted data to be attacked by cryptanalysis or brute force attack. In its scheme, they distribute encryption keys and partially encrypted data into the DHT. Although this method has prevented some attacks mentioned above, if someone has already obtained the key and has some backup, someone can also decode encrypted data. Moreover, [5] improves the Shamir secret sharing algorithm and the length of the key is extended to resist the presence of jump attacks in the Vanish system. Resistance to sniffing attacks, [5] uses public-private key encryption and decryption. However, the cloud stores complete encrypted data, the scheme can not resist brute force attacks and cryptographic attacks.

Next, [6] points out a kind of solution that divides the very beginning encrypted data into the sampling data and the remaining data. Then, they deliver the remaining data to the cloud, which makes it impossible for the unreliable CSP to get all the encrypted data. The drawback in document [6] is that it introduces the third party that should be trusted to save the sampled data. But, this premise is too idealized. As mentioned in [7], if we can't totally trust the cloud service provider, should not we have

the same doubt to the other third parties? When the government has a court order to force the cloud and third parties to hand over the data and key of the investigated company, no matter how hard the data owner tries to delete the data, it will be useless [8, 9].

In order to reduce the risk of data leakage, we propose AD-IHSE, which not only allows the encryption key to be stolen, but also does not need any other third party's management or storage. In AD-IHSE, we firstly do some data extraction for the encrypted data at the bit stream level. The algorithm of extraction guarantees the randomness of data extraction position. When data owner needs to delete data, the only thing to do for him is to delete the random sequence that is used as the location number of the extraction. This can guarantee the security even if CSP does not delete the encrypted data as normal. Without the location information, the extraction can not be recovered by anybody. After data extraction, the encrypted data is divided into two parts. The big part of encrypted data will be sent to the cloud service provider directly. The small one is embedded into default carrier by using information hiding technology. After being hidden, the Stego is encrypted for the second time. Finally, the encrypted Stego is also uploaded to the cloud. Moreover, this method does not need any other third party. The content stored in the cloud is not the entire encrypted data, but the two parts of the encrypted data. The above process ensures that even if the cloud data and encryption keys are stolen by an attacker, the deleted data will not be accessed again.

3 Security Assumptions and Threat Model

3.1 Security Assumptions

Our security intention is to complete reliable data deletion when the encryption key has been stolen by an attacker. This method has three hypotheses. First, the encryption operation is safe, in the situation that an attacker can not recover encrypted data without a decryption key. Secondly, although attackers can decrypt the encrypted data of cloud into corresponding data, they cannot recover and merge encrypted data without random sequence. Third, because of the high cost of storage, data users will not backup the original data after access.

In this article, we consider an idea which has three main entities: the data owner, the cloud service provider (CSP) and the data user.

Data owners are responsible for data encryption, data extraction, information hiding, and two level encryption.

The cloud service provider is responsible for storing the uploaded data. But CSP is not trusted. It works faithfully to data storage, but it is also curious about sensitive information and wants to get sensitive data.

Data users are responsible for data decryption and data recovery.

The data access process is shown in Fig. 1. The red line is used to represent the process for data owner uploading the data. The blue line is used to represent the process for data user visiting the data.

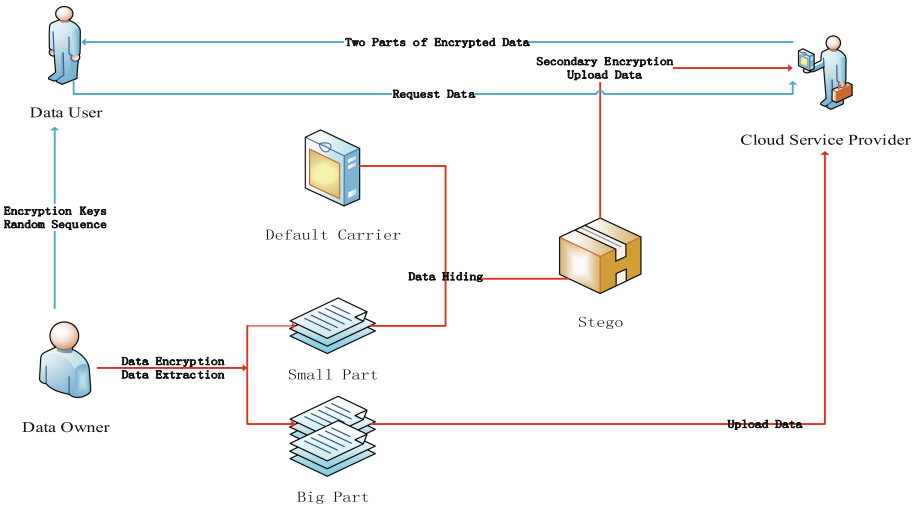

Fig. 1. Data access (Color figure online)

3.2 Threat Model

According to the above security hypothesis, several conditions are defined as follows: CSP is not trusted. It will divulge encrypted data to other people or some malicious users. The contents stored in CSP are not safe. The attacker's behavior is not real, but a subsequent act. Therefore, an attacker will not know what useful data is available before the data is accessed. Data users are trusted. It does not retain the original data or use for transmission.

4 Design and Security Analysis

In this method, it is not the original complete encrypted data that is retained on the cloud. It is based on the following conditions: if an attacker leaks the encryption key in the active and passive case, the data can also be secure. In addition, this method has only data owners and data users, without any other third parties performing some transmission or storage work, which improves safety.

In this design, the chaotic system generates a random sequence as the location information for the next step in the preparation of information extraction. Besides, the random sequences are just stored by the data owner. Choosing the appropriate parameter values, the logical mapping can enter the chaotic state. When you need to delete data, the only thing to do is to delete a random information that is used as the number of the transformation positions. If there is no position numbers, the extraction for encrypted data can not be recovered by anyone. Then, data is split into two fragments according to the logistic chaotic sequences. The small part is hidden into a default gray image by using the modified LSB algorithm. The image with the small fragment of the encryption data is encrypted by logistic algorithm for the second time.

Both the big part cipher text and the encrypted image containing the small fragment of encrypted data are uploaded to CSP. For formalization, we now modify our symbols as follows. Let {M} be the original data. Let q be the bit number of extracting bits. Let {K} be the small part of encrypted data. Let $\{S_i\}(1 \leq S_i \leq q)$ be the position number. Let $\{C_i\}$ be the encrypted data. Let k be the encryption key. Let *Extract*() be the function of bit stream extraction. The detail will be explained in the following parts.

4.1 Data Extraction

If the hiding position is selected in order, it will cause the image of the various parts of the statistical characteristics of inconsistencies leading to serious security problems. The modified part of carrier and the unmodified part have different statistical characteristics, which increases the possibility that the attacker has doubts the secret communication.

For the sake of solving this issue, the random sequences are used to select the sequence of pixel by logistic chaotic mapping to product chaos sequences. It is the random sequence that makes the conversion safer. According to the traditional definition, the expression of logistic chaotic mapping is (1):

$$X_{i+1} = f(X_i) = \mu X_i(1 - X_i) \tag{1}$$

In this formula: $X_i \in [0, 1]$ is a logistic mapping state. From the Chaotic Sequence definition, logical mapping parameter $\mu \in [3.5699456, 4]$, the logistic mapping will enter into chaotic state [10]. There are n irregular numbers generated by logistic mapping. In order to ensure data security without bringing additional third party management operations, the encrypted data will be extracted. The random sequence is generated by using the logistic chaotic system as the data extraction position, and the cipher text is divided into two parts.

$$K_i = Extract(M, S_i) \tag{2}$$

The small portion of the extracted data is called the extraction of data, and the other is called the change of data. After extracting, the location information extracted from the extracted data and data will be saved only by the data owner and not sent to anyone else. The sketches extracted by the bitstream are shown in Fig. 2.

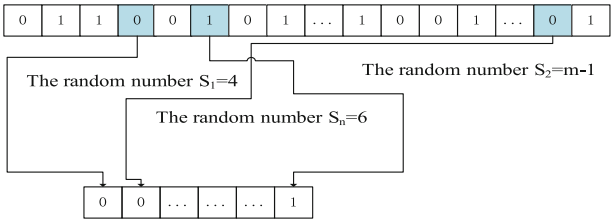

Fig. 2. Data extraction

4.2 Extracted Data Hiding and Secondary Encryption

After extracting the cipher text, the small fragment is hidden as secret information into the vector image by using LSB algorithm. This hidden algorithm is generally in the BMP format images, with a good concealment, hidden capacity, low computational complexity. However, we need to make some improvements to strengthen its security.

Firstly, select the encrypted information rather than the plaintext as a secret to hide into vector image. Secondly, choose the random position rather than a sequential position to be the hidden location. Data owner uses logistic chaotic sequence to generate an random index sequence $\{j_1, j_2, \ldots, j_{L(m)}\}$. The k_{th} secret extracted cipher text bit is hidden in the lowest bit of the index carrier element j_k. Moreover, in order to prevent the random sequence from colliding, the generated index values are recorded. If a collision occurs, the currently used value is discarded and the next index value will be used. From a mathematical point of view, the grayscale image is a matrix whose mathematical expression is as follows:

$$\mathrm{I} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1N} \\ g_{21} & g_{22} & \cdots & g_{2N} \\ \vdots & \vdots & & \vdots \\ g_{M1} & g_{M2} & \cdots & g_{MN} \end{bmatrix} \tag{3}$$

$$\mathrm{I} = \{g_{xy} | 1 \le \mathrm{x} \le \mathrm{M}; 1 \le \mathrm{y} \le \mathrm{N}\} \tag{4}$$

M and N denote the image samples as M rows and N columns, g_{ij} is the image after sampling and quantization of the pixel position of the gray value. In order to improve the anti-privacy performance of dense images, it is necessary to change the gray value of the pixel that hides the bit when embedding the secret information.

$$\mathrm{T} = \sum\nolimits_{u=\mathrm{i}-1}^{i+1} \sum\nolimits_{v=j-1}^{j+1} g(u,v) - 9g(i,j) \tag{5}$$

The secret information has been hidden into the vector image in the previous section. For higher security, this paper uses two-dimensional logistic mapping for image encryption processing. The dynamic equations of the two-dimensional logistic mapping are as follows:

$$\begin{cases} x_{i+1} = \mu\lambda_1 x_i(1 - x_i) + \gamma y_i \\ y_{i+1} = \mu\lambda_2 x_i(1 - y_i) + \gamma x_i \end{cases} \tag{6}$$

Using the generated real-valued chaotic sequence $\{x_i\}$, the symbol sequence is generated by defining the threshold Symbol (x_i). The symbol matrix is generated in row by Symbol (x_i). Using the generated real-valued chaotic sequence $\{y_i\}$, the sequence is generated by the following transformation to generate the corresponding gray-scale matrix G.

$$\text{Symbol}(x_i) = \begin{cases} -1, & -1 \le x_i < 0 \\ 1, & 0 \le x_i \le 1 \end{cases} \tag{7}$$

$$\text{Gray}(y_i) = \text{Round}\left(y_i \times \frac{225}{2} + \frac{225}{2}\right) \tag{8}$$

$$I_{encrypt} = \text{I} \oplus \text{G} \cdot \text{S} \tag{9}$$

4.3 Data Decryption and Recovery

While the data owner wants to access the data, its request is firstly sent to CSP. After checking data user's legal identity, CSP will send both big part of cipher text and the encrypted image with small part of cipher text to data user. Besides, data owner will send the encryption key and the chaotic sequence of this file to data user. Next, the encrypted image is decrypted by encryption key and a small cipher text is extracted from the image. The cipher text is synthesized according to the extracted position of the chaotic sequence.

4.4 Data Assured Deletion

While the data owner expects to delete the data, the only thing that he need to do is to delete the location sequences of the bit stream extraction process. This information is saved by the data owner after the data is extracted. If CSP deletes the encrypted data, it will be more secure. However, as CSP is trustless, it is not guaranteed to operate as expected. When the location has been deleted, the attacker can not restore the data even if the encryption key and any cloud encrypted data have been extracted at the same time, changed and encrypted for the second time. The way to delete data is to delete it in real time. And the data can be guaranteed to be deleted. Without deleting the request, the location part extracted from the data will be saved by the data owner.

4.5 Security Analysis

In this paper, information hiding is used to encrypt the part of cipher text for the second time. This solution does not need other third parties, nor does it put the original cipher text directly up to the cloud. In this case, for the sake of getting the original data, an attacker needs to not only get two encryption keys for both first and second encryption, but also has to combine the two parts of cipher text into the original one. Or, if he only gets the encryption key, the cloud does not have a complete cipher text ready for the attacker to decrypt.

Theorem: in the condition of the same data length, when attackers acquire the encryption key, it is more difficult for attackers to get encrypted data by AD-IHSE than to recover raw data in traditional way.

Proof: From the above mentioned procedure, it is not hard to see that the key process is to obtain the function of extracting bit stream. This scheme has better security compared with the scheme that an attacker has no encrypted key to get the

complete encrypted data. Though attackers already have encryption keys, it is more difficult for him to know the real raw data instead of converting the functions of bit streams, rather than using complete encrypted data and encryption keys to guess the original data. The security of the method is determined on the security of the conversion code flow algorithm. Let D_t become a difficult problem for an attacker to guess the original data in the scenario described in this article. Let D become an attacker's difficulty in guessing original data with complete encrypted data without encryption keys. That is. $D = 2^P$, p is the length of the encryption key of the symmetric encryption algorithm. When each data block length is l bits, then $D_t = (2l)^n$, n is the number of data blocks. That is $p \approx l \cdot n$. Supposed, the length of encryption key is equal to the length of data block, that is. $p = n$.

$$\frac{D_t}{D} = l^n \tag{10}$$

As shown by (10), AD-IHSE has at least a higher security than a traditional attacker having encrypted data without encryption keys. The sequences generated by the chaotic system is irregular, the more the number of iterations, the more intense the chaos will be. However, the pseudorandom sequences have inherent regularity because of the system seeds. The last but not the least, the encryption key and the encrypted data are stolen by an attacker, as long as the data owner has deleted the chaotic sequence, the

Table 1. Comparison of security with other schemes

Comparison item	Deletion content	Whether the cipher text is destructed	
FADE in [2]	Decryption key	NO	
Vanish in [3]	Decryption key	NO	
SSDD in [4]	Decryption key and extracted cipher text	YES	
Safe Vanish in [5]	Decryption key	NO	
Two-party in [7]	Decryption key	NO	
ADCSS in [6]	Extracted cipher text	YES	
Our scheme	Information of extracted cipher text	YES	
Comparison item	Whether introduces a third party	Whether can resist brute force attack	Whether can resist cryptography analysis
FADE in [2]	YES	NO	NO
Vanish in [3]	NO	NO	NO
SSDD in [4]	NO	YES	YES
Safe Vanish in [5]	NO	NO	NO
Two-party in [7]	NO	NO	NO
ADCSS in [6]	YES	YES	YES
Our scheme	NO	YES	YES

attacker will only affect the current file. Because of the chaotic characteristics of the logical chaotic mapping, the attacker can not obtain any useful information about the former or the post files. Therefore, the AD-IHSE guarantees the forward secrecy and the backward secrecy.

Table 1 compares our scheme with other security related guaranteed deleting schemes.

5 Implementation and Experimental Analysis

We have done experiments as follows to show both the security of our scheme and the time costs of this process. We have implemented a prototype system of our scheme using JAVA and the experiments are conducted on a PC with Mac OS, Intel Core i5 2.7 GHz Processor and 8 GB Memory.

5.1 Performance of Information Hiding

The Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) are two objective evaluation method. The MSE represents the cumulative squared error between the original image and dense image whereas PSNR represents a measure of the peak error. Suppose $r = n/L$ is the ratio of the length of the secret information to the total number of carrier pixels.

When r is becoming higher, the image quality will be lower. However, the greater the amount of information embedded in the image, the more difficult for the attacker to recover the original data. The PSNR is often expressed on a logarithmic scale in decibels (dB). PSNR values falling below 30 dB indicate a fairly low quality. A high quality Stego should strive for 40 dB and above [11]. Considering the above two factors, this paper chose that 40 dB is more appropriate. As is shown in Fig. 3, the Guides line means that PSNR is 35 dB. r = 2.5.

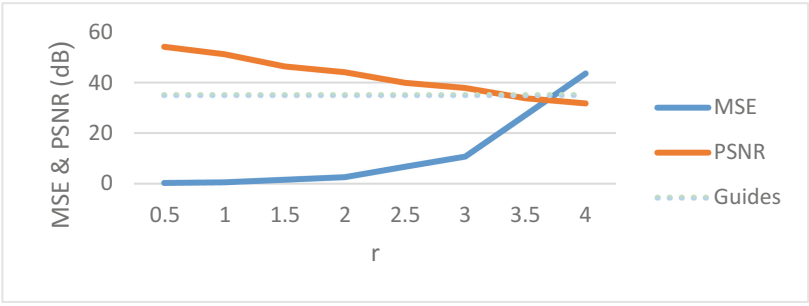

Fig. 3. PSNR changes with the embedding rate

When the carrier image is selected, the number of embedded bits can be determined according to the r calculation formula.

5.2 Time Costs for Data Extraction

This section analyzes the time performance of the cipher text extraction, according to the relationship between the size of the encrypted file and the time consumption. The trend is shown in Table 2.

Table 2. Time costs

File size (M)	1	8	16	32	64	128	256	512	1024
Time for encryption (ms)	24	102	195	380	734	1505	2912	5735	11445
Time for extraction and hiding (ms)	8534	9019	10010	9716	10514	11218	17325	23542	39439
Time for decryption (ms)	16	96	254	366	722	1416	2805	5919	12533
Time for recovery (ms)	8620	9090	10102	11542	11403	11753	17924	24012	40137
Total time (ms)	17194	18307	20561	22004	23373	25892	40966	59208	103554

From Table 2, we can see that the time of extraction and hiding for 1024 M file is less than one minute. The time it takes for the steps we introduce is consumed within a reasonable range. This scheme has a little more time costs, but it gains more security as we had proved in security analysis.

6 Conclusion and Future Works

In this article, we propose a new solution called AD-IHSE to solve the issue of cloud data assured deletion. In this method, we do not bring other third parties, nor do we store the complete encrypted data in the cloud service provider. If the data owner has deleted the data, even if the encryption key has been appropriated by some attackers in the worst condition, we can also achieve the intention that the data will not be accessed. The scheme is based on the procedure of information hiding and two encryption chaotic sequences. After encryption process, the data owner must use the Logistic chaotic mapping for data extraction so that the encrypted data are divided into two parts. By using this method, we can ensure that the encrypted data in the cloud are not complete. Therefore, even if the attacker has acquired the encryption key and encrypted data in the cloud, the attacker can not scramble the location information of the bit stream of the raw data and encrypted data, but can not decrypt the encrypted data in the cloud. So, if the data is determined to be deleted, what we really delete is information about the change location. Finally, security analysis and experimental results show that

our method can satisfy the desire of data owners to make sure that deleted data will not be accessed by illegal users. The next work is to consider the optimization of vectors in information hiding.

Acknowledgment. This work was partly supported by the NSFC-Guangdong Joint Found (U1501254) and the Co-construction Program with the Beijing Municipal Commission of Education and the Ministry of Science and Technology of China (2012BAH45B01) and the Fundamental Research Funds for the Central Universities (BUPT2011RCZJ16, 2014ZD03-03) and China Information Security Special Fund (NDRC).

References

1. Perlman, R.: File system design with assured deletion. In: The 14th Annual Network & Distributed System Security, pp. 1–7. IEEE (2007)
2. Tang, Y., Lee, P.P.C., Lui, J.C.S., Perlman, R.: FADE: secure overlay cloud storage with file assured deletion. In: Jajodia, S., Zhou, J. (eds.) SecureComm 2010. LNICST, vol. 50, pp. 380–397. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16161-2_22
3. Xiong, J., Yao, Z., Ma, J.: A secure document self-destruction scheme with identity based encryption. In: The 5th International Conference on Intelligent Networking and Collaborative Systems, pp. 239–243. IEEE, Xi'an (2013)
4. Wang, G., Yue, F., Liu, Q.: A secure self-destructing scheme for electronic data. J. Comput. Syst. Sci. **79**(2), 279–290 (2013)
5. Zeng, L., Shi, Z., Xu, S., Feng, D.: Safevanish: an improved data self-destruction for protecting data privacy. In: Proceedings of CloudCom, pp. 521–528. IEEE (2010)
6. Zhang, K., Yang, C., Ma, J.F., Zhang, J.W.: Novel cloud data assured deletion approach based on cipher text sample slice. J. Commun. **36**(11), 108–111 (2015)
7. Mo, Z., Qiao, Y., Chen, S.: Two-party fine-grained assured deletion of outsourced data in cloud systems. In: International Conference on Distributed Computing Systems, pp. 308–317. IEEE (2014)
8. Li, C., Chen, Y., Zhou, Y.: A data assured deletion scheme in cloud storage in China. Communications **11**(4), 98–110 (2014)
9. Shahzad, F.: Safe haven in the cloud: secure access controlled file encryption (SAFE) system. In: Science and Information Conference, pp. 1329–1334. IEEE (2015)
10. Patidar, V., Pareek, N.K., Purohit, G., et al.: A robust and secure chaotic standard map based pseudorandom permutation-substitution scheme for image encryption. Opt. Commun. 4331–4339. IEEE (2011)
11. Deshmukh, P.U., Pattewar, T.M.: A novel approach for edge adaptive steganography on LSB insertion technique. In: International Conference on Information Communication and Embedded Systems, pp. 1–5. IEEE (2015)