TEST Engineering & Management

May-June 2020
ISSN: 0193-4120 Page No. 4653-4656

Sensitive Data Sanitization along with Encryption and Third-party Auditing

Pavan Kumar S[1], Nimmanapalli Sai Abhijith Reddy[2], Nithesh G[3], Mrinal Mohan[4] Shruthi G[5*]
[5]Professor
[1,2,3,4,5]School of Computing & Information Technology, REVA University, Bangalore, India
[1]pavansrinivas67@gmail.com, [2]abhijithreddy888555@gmail.com, [3]nitheshgowdru49@gmail.com, [4]mrinalmohan28@gmail.com, [5]shruthig@reva.edu.in

Article Info
Volume 83
Page Number: 4653-4656
Publication Issue:
May-June 2020

Article History
Article Received:*19 November 2019*
Revised: *27 January 2020*
Accepted: *24 February 2020*
Publication: *12 May 2020*

Abstract

Cybercriminals will be in hunger, looking into the easy links to break the algorithm. Security being the primary essential of anything and everything in this world, it's been more challenging to safeguard the sensitive data of every individual. In the recent trends avoiding data corruption and providing security for our data is of very difficult task. When it comes to the data security in banks, Hospitals, Pharmacy, Organization and so forth is completely crucial. Encryption is on high step to be chosen for hiding the sensitive data. There are many of such encryption strategies that would secure your sensitive data. Sanitization has to be done to recognize the sensitive data either manually or by any automation methods. Considering the drawbacks of many encryption techniques, this paper proposes the AES-256 algorithm. Existing methods only provide private auditing whereas, our proposed system eliminates this drawback too. This paper also stresses on how fast the data is encrypted and how easy it is to control and coordinate the process of encryption.

Keywords: *Security, Sanitization, Encryption*

1. Introduction

In the traditional cloud storage system, entire file was been encrypted before uploading or residing it on cloud. This makes it difficult to encrypt the entire file if the file is very large, that results in time consumption, Data corruption and large space utilization. Thus our proposed system understand each and every drawback, it provides only a chosen data (sensitive), Encrypts it thus providing security. The selection of such sensitive data is termed as Sanitization. The AES-256 algorithm having keysize of bit 256 which helps the largest bit size and which is impossible to break using brute force using standard computational power and it is the strongest encryption algorithm. Thus our paper revolves under the usage of only AES-256 algorithm due its idiosyncratic features as compared to other algorithms. The DES Algorithm has lot more limitations compared to AES-256 Algorithm. DES has a weak key compared with AES-256.DES algorithm will be expecting same result from S-boxes on permutation on different data inputs given, which are called as semi-weaker keys and 1's compliment of the encrypted data will be as same as complimented messages by encryption of the and compliment keys. So we made a conclusion to take AES_256 bit algorithm.

2. Related Work

In [1] it introduces a secured cloud storage which supports Publicauditing. The approach also provides third party auditing. In [5] discussed on third-party auditing, wherein the data sent by the client to the server is checked and maintained by the auditor who ensures that data will be secure and remain unharmed in the cloud. In [2] describes the technique that allows the client to protect and store their data in cloud so the auditing protocol is efficiently executed in order to check if the cloud has copies all of his data for the future use. In [8] discussed on proofs of retrievability which provides light-weight storing and proving but the verification time is longer because the entire block of data needs to be encrypted and must be maintained by the server while in our approach it provides third party auditing along with data sharing with sensitive info hiding. In [3] it uses an

May-June 2020
ISSN: 0193-4120 Page No. 4653-4656

approach which is mainly based on untrusted server the data integrity will be stored. This technique presents a definite frameworks also with constructions for dynamically provable data integrity possessions wherein integrity of data is maintained. In [7] the security of data is maintained with a technique but the entire data sent by the client needs to be encrypted whereas in our paper, along with maintaining integrity of the data, only specific or sensitive data blocks are encrypted which requiresless storage so taken the idea of data integrity from this technique. In [4] it is totally made clear of the fact how strong is AES encryption techniques. In [11] it has allowed us to make difference between various AES algorithms and we had come to conclusion that AES-256 Algorithm is unbreakable with our current computing power due its very huge amount of possible key combinations.

Below were the different algorithm techniques among which AES-256 Algorithm was selected in this paper.

- Cipher text-policy Attribute-Based Encryption
- De centralizing attribute-based encryption
- AES (128 bit Advanced Encryption Standards)
- **AES (256 bit Advanced Encryption Standards)**

3. Methodology

AES-256 Algorithm:

Encryptions is common way for sensitive hiding of data (sensitive encryption). AES-256 Algorithm's key length is 256 bits, allowing it very difficult to break it. As the key size increases the possible keys combination also increases, which is why this algorithm is unbreakable by using any of the methods like brute-force.

Table 1: Key size and respective possible key combinations

Key Size	Possible combinations
1 bit	2
2 bits	4
4 bits	16
8 bits	256
16 bits	65536
32 bits	4.2×10^9
56 bits (DES)	7.2×10^{16}
64 bits	1.8×10^{19}
128 bits (AES-126)	3.4×10^{38}
192 bits (AES-192)	6.2×10^{57}
256 bits (AES-256)	**1.1×10^{77}**

The AES-256 Algorithm was formerly called as Rijndael as its primary designers were Vincent Rijmen and Joan Daemon.

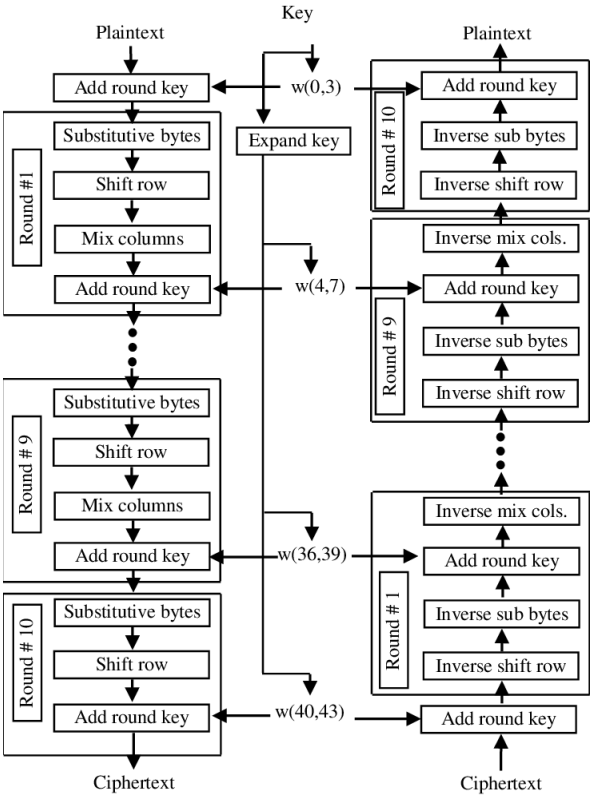

Figure 1: AES-256 Algorithm work flow

Figure 1 describes the work flow of AES-256 Algorithm.

- **Sub bytes**: The sixteen bytes of input is to be exchanged by a look into a fixed table that is S-box given in model. The output matrix is a 4 *4 matrices.
- **Shift rows**: Every row of the matrices is shifted to left Any entries that doesn't seems to fit there will be inserted at extreme right of that row.
- **Mix columns**: Now each column consisting of thirty-two bits is converted using a mathematical algorithm. This algorithm generally will take 32 bites as an input for one of the column and will return four new bytes of data, which replaces the original data column. The result of this is another new matrix consisting with 16 new bytes of data.
- **Add round key**: The sixteen bytes that we have obtained from the previous Mix columns are taken as 128 bits and functionally XOR'ed to 128 bits of the add round key. The output cipher text is obtained if this was the last round. Else, a resulting 128 bits are again considered as 16bytes and thus, we repeat the process till we get the desired output.

May-June 2020
ISSN: 0193-4120 Page No. 4653-4656

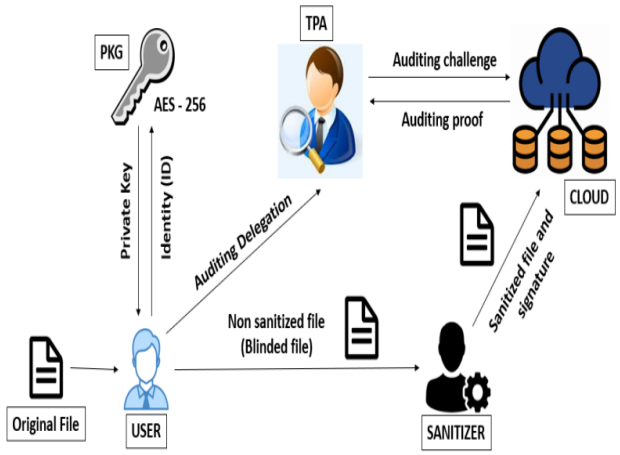

Figure 2: Work flow of proposed system

The above model which consists five different kind of entities, they are:

- **Cloud:** It provides enormous data storage to the user space.
- **User:** member of organization who provides the original file.
- **Sanitizer:** sanitizing the data blocks inside the original file.
- **PKG:** Private Key Generator. Secret key which is another name for private key, which will a variable in cryptography which isused to encrypt and decrypt using the algorithms. Key's generator are only shared by secret keys which will be highly secured.
- **TPA:** Third Party Auditor. The user's data can be stolen or modified by outside attacker, causing the user to be concerned about the integrity of data stored in cloud. Hence, we make use of data auditing, data auditing with the help Third Party Auditor (TPA) entity then check the integrity of data in the cloud.

Figure 2 depicts in flow the proposed system. Here original file is taken from the user. This file is not sanitized i.e. sensitive information is not identified. Thus, the file is sent towards the sanitizer who takes out the sensitive information or data from the original file of the user. Meanwhile a (PKG) Private key Generator generates the key using an AES-256 algorithm and sensitive data is only encrypted. This encrypted file is kept in the cloud. The TPA will always make sure that our data will be safe.

Advantages of AES-256 Algorithm compared to other algorithms are listed below

- 256 bit AES which supports biggest bit which will have the length 256(bits) and which is impossible to break by brute force based computational power and it is strongest encryption algorithm.
- Programming languages like C, C++, JAVA, and PYTHON was developed from AES LIBRARIES.
- For encryption and as well as decryption the secret key will be same and AES will be a symmetric key cipher.
- The copy of the key will be with both sender and as well as receiver.
- The asymmetric key has different keys for different processes and for the file transfers Asymmetric key will be best and for Symmetric key used for internal encryption.
- The plus point for AES is speed why because it requires very less power, so it faster and efficient.
- The AES-256 is used in several places such as SSL or TLS . It is one of the best cipher techniques. It's very difficult to crack the file unless they get proper combination of keys.
- AES-256 is one of the most secure and protected encryption protocol.
- 1.1 x 10^{77} attempts are required for the hacker to break the encryption which is impossible for one as per the current computing power that we have.
- This algorithm is faster in both software and hardware which makes it more useful than any other algorithms.
- The algorithm is unbreakable by using any of the methods like brute-force.
- The DES 56-bit is not secured and it can br cracked very easily within a day, but thus the AES is not the easy to break and they need n number of years to breaks it.
- Hackers will not even look into it ,if it is a AES encrypted

Applications

Our proposed systems deals with application in Banking sectors, Pharmacy, Hospitals, University databases, Employee database and many more.

- **Banking Sector:** Information such as password, account number, phone number, address and many confidential information has to be encrypted or else it may be very difficult to secure these confidential data.
- **Pharmacy:** Lab tests in pharmaceutical research are hidden safely as someone else may take it. Therefore, it is very necessary to provide encryption.
- **Hospitals:** Patients name, age, drugs or medicines prescribed by the doctor may come as sensitive for some or the other patients, hence it may be very much in use for the patient as well as the doctor to safeguard their prescription. Thus, providing proposed system to them may safeguard their concern.

4. Conclusion

We proposed a user-based sensitive data encryption along with auditing scheme to ensure data integrity for secure cloud storage systems, also to ensure less data corruption. In this scheme original file is taken from the user. This file is not sanitized i.e. sensitive information is not identified. Thus, the file is sent towards the sanitizer who takes out the sensitive information or data from the original file of the user. Meanwhile a key will be given by PKG using a 256 bit AES algorithm, the sensitive data is only encrypted. This encrypted file will be accumulated in cloud. The Third-party Auditor always ensures that our

May-June 2020
ISSN: 0193-4120 Page No. 4653-4656

data is safe. Besides that, most importantly our system ensures encryption of only sensitive information as discussed resulting in less time consumption and less storage. Earlier the whole file was been encrypted which would result in data corruption, more storage and much time consumption. All over to be specific our proposed system combines all the essentials that a user thinks of his data to be secure requires as our system provides less data corruption, less storage and importantly less time consumption.

References

[1] K. Ren, C. Wang, and Q. Wang, *"Security challenges for the public cloud"*, IEEE Internet Computing, vol. 16, no. 1, pp. 69–73, Jan 2012.

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, *"Provable data possession at untrusted stores"* ,in Proceedings of the 14th ACM Conference on Computer and Communications Security, ser. CCS '07, 2007, pp. 598–609.

[3] Juels and B. S. Kaliski, *"Pors: Proofs of retrievability for large files",* in Proceedings of the 14th ACM Conference on Computer and Communications Security, ser. CCS '07, 2007, pp. 584–597.

[4] H. Shacham and B. Waters, *"Compact proofs of retrievability*", J. Cryptology, vol. 26, no. 3, pp. 442–483, Jul. 2013.

[5] Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "*Privacy-preserving public auditing for secure cloud storage*", IEEE Transactions on Computers, vol. 62, no. 2, pp. 362–375, 2013.

[6] S. G. Worku, C. Xu, J. Zhao, and X. He, *"Secure and efficient privacy preserving public auditing scheme for cloud storage*",Comput. Electr. Eng., vol. 40, no. 5, pp. 1703–1713, Jul. 2014.

[7] Chris Erway, Alptekin Küpçü, Charalampos Papamanthou, Roberto Tamassia, *"Dynamic Provable Data Possession*", 2015.

[8] Guan, K. Ren, F. Zhang, F. Kerschbaum, and J. Yu, *"Symmetric-key based proofs of retrievability supporting public verification*", in Computer Security – ESORICS 2015. Cham: Springer International Publishing, 2015, pp. 203–223.

[9] Chaowen Guan, Kaili Ren, Fangguo Zhang, Florian Kerschbaum, "*Symmetric-Key Based Proofs of Retrievability Supporting Public Verification*", 2015.

[10] Cong Wang, Sherman S. M. Chow, Qian Wang, Kui Ren, Wenjing Lou, "*Privacy-Preserving Public Auditing for Secure Cloud Storage*", 2013.

[11] Joan Daemen and Vincent Rijmen, *"The Design of Rijndael: AES- The Advanced Encryption Standard*," Springer-Verlag, 2002.

[12] Wu, D. and Haven, J., 2012. "*Using homomorphic encryption for large scale statistical analysis*". Technical Report:cs. stanford.edu/people/dwu4/papers/FHESIReport, pp.1-20.