Journal of Autonomous Intelligence (2024) Volume 7 Issue 1
doi: 10.32629/jai.v7i1.567

ORIGINAL RESEARCH ARTICLE

Intelligent encryption with improved zealous method to enhance the anonymization of public health records in cloud

P. Malathi[1,*], Devi S. Suganthi[2], J. Jospin Jeya[3]

[1] *Department of Computer Science and Engineering, Annamalai University, Chidambaram 608002, Tamil Nadu, India*

[2] *Department of Computer Science and Engineering, Srinivasa Subbaraya Polytechnic College, Puttur 609108, Tamil Nadu, India*

[3] *Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram Campus, Chennai 600089, Tamil Nadu, India*

*** Corresponding author:** P. Malathi, malathi02it@gmail.com

ABSTRACT

In order to keep the cost of installing advanced encryption scheme (AES) hardware to a minimum and to hide the protected key from hackers, existing networks' tactics are employed in this study. This is the biggest drawback of the present methods since there is no security while keeping the concealed key of the AES encryption method. The user is unable to remember all the AES keys since it is necessary to connect with several people using various AES keys. The suggested approach successfully counters both facet and brute force assaults. The advanced encryption method is the safest algorithm to utilize for trustworthy encryption, according to the findings. But the issue is that the advanced encryption scheme makes brute force attack less effective. Honey encryption is thus used. The recommended method encrypts the dataset after dataset anonymization to ensure privacy. An enhanced zealous technique is used to do anonymization when a middle dataset is created using the supplied key. After the dataset has been sorted, the dataset with the higher rank is the one that gets encrypted first. The user receives these datasets together with a decryption key for the encrypted records, enabling them to swiftly obtain the data they need.

Keywords: honey encryption; advanced encryption scheme; security; brute force attack; public health record; improved zealous method

ARTICLE INFO

Received: 26 March 2023
Accepted: 6 August 2023
Available online: 17 October 2023

1 Introduction

The cloud data protection issues have been raised by recent security events concerning public cloud data storage. In the past few years, growth in the hacking area has increased. As a consequence, cyber protection which acted as essential part in the covering of sensitive information is needed. At present, the difficult channel attack is both the symmetric and the asymmetric algorithm of encryption. The hidden key has to be communicated via a security demanding medium in both parameters. The focus is on the private key limitations for easy use of certain round key in different locations of the AES algorithm. In hackers' opinion, having the straight-forward text for the encryption key is virtually the same. The best encryption technique is then used to point the intruder by producing a wrong key at any attempt. If the intruders try to do so, the storage manager immediately produces an indicator. Honey encryption is the easiest algorithm to solve the shortcomings of the AES algorithm but it still has certain time

restrictions. This prevents the brute force attack and provides a secure system to keep the hidden key.

The suggested scheme overcomes the complexities that occur using the methods below. In order to address different problems that impact the security of the AES algorithm, the techniques use different approaches. An algorithm proposed by Juels provided a stable encryption method is in 2014. Whenever a false key is used to generate the original messages, it needs to be figured out whether a hacker is attempting to hack or not by generating the warning. With the honey algorithm, the message is encrypted. The distinguished capacity and confidentiality are the two primary notions of the honey. Majeed et al.[1] used the notion of honey in his research paper to make the password safer. Within the password bank, several passwords can be made secure. A method to secure passwords in the password vault was introduced by Juels et al.[2]. The files are the data in the password vaults.

The main drawback to this automated symmetric encryption being the brute force intrusion, which is the first assault by deleting the final two bits from the AES key. The remaining AES assaults are dictionaries. The main value of the strategy against the AES is the absolute absence of the brute force attack. So, we can solve the issue with the honey concept of AES key storage systems[3]. The main feature of the honey algorithm is when the assailant first tries to generate the false, believable keys with the wrong private key. When choosing the real keys from the massive plausible buttons, the attacker will be distracted by these fake keys. Honey encryption is the way an assailant tricks the wrong key that appears to be a genuine key, so that it has been misleading by picking the right key[4].

These two algorithms need to be merged since the honey technique is free of the BFA. The flaw with the honey technique is that the attacker produces a false key on any wrong attempt[5]. So, we have to build every time an immense number of false keys that should really look like an initial key. It just takes 128-bit keys to encrypt honey and so it doesn't take a long time and as it's a simultaneous process to make sure that the issue is still effectively solved.

2. Related work

This research personal health record (PHR) is outsourced to a semi-trusted third-party server. This leads to problems with patient-sensitive health details about protection and privacy[6]. This article proposed on security protection mechanisms to obtain the best solution in the cloud environment in order to prevent from attacks and sharing the data safely in the cloud.

2.1. Related works for analysis of criticality of public health record (PHR) in cloud

The diagnosis and treatment of diseases in health care systems can have a better quality of service and assist various stakeholders like doctors, patients and scientists by performing artificial intelligence-based data analytics on health records[7,8]. The attributes of the input training set form the basis of statistical classification of PHRs into groups or classes[9]. Support vector machines, decision trees and Bayesian networks are some of the state-of-the-art approaches for statistical classification. To overcome the issue of uncertainty in datasets, there is a need for a probability-based approach and hence the Bayesian classification is the most widely used technique for classification. On the other side while hosting the deep learning applications on server is an easiest one if it is hosted on any public central service cloud providers such as Amazon EC2, Google app engine etc. these public cloud servers can be used for a highly complicated building application to get an efficient and accurate result for lack of mistakes should not occur on it[10]. Deadline based resource is provided by the algorithm to process millions of instructions. As these instructions are to be processed in balancing the load for which it is used to identify the priority vector for dynamic list of preserving cloudlets. Scalability can be achieved by applying encryption techniques for producing in an efficient manner[11]. A successful verification between the patient and physician should be made if a pair or assertion happens in the cloud

processing by generating a set of public keys and private keys and sending the key information to both the physician and client's device[12].

Long short-term memory (LSTM) is applied for predicting and capturing the spatial changes occurring on the network traffic. Latency may vary due to nature of the guaranteed network with tolerated to the depending area[13]. This is vital for decision making in the significant volume of generating the healthcare data which has to be generated day to day[14]. As the data has to be transmitted continuously edge-fog-cloud computing process for coexistent of data which needs to be extracted whenever needed. The pushing and pulling methods are used: push is used to initiate the first block for sending the data and pulling is used to execute the output of initiating block. The continuous security scheme is made to establish the data integrity, accessibility and verification. There are many deep learning algorithms but a suitable algorithm is to be implemented in order to gain the result[15]. Diagnostic reports are highly sensible as patient details are to be maintained confidentially and not be stolen by any physical or cybercrime thefts. Such encryption of these sensible data is to be made before it has to be outsourced[16]. Cryptographic algorithms can be used for making more confidentiality on increasing the security purposes. In case of any emergency the data should be accessed as soon as possible by the doctor or patient by making a special search which can be made by adding a remark for improving the integrity[17]. The authorized set of attributes is to be associated with set of rows and columns by sharing the generated matrix key within a locality by cipher text which is attached by the cloud service providers.

A simple Bayesian network makes use of the conditional independence among attributes for the given class which is impractical. However, crisp partitioning of the domain in Bayesian classification results in loss of information[18]. To overcome this issue, fuzzy logic is used for the fuzzification of health attributes in health care domain[19]. Since healthcare data comes under the category of big data, proper attention should be provided for the term speed.

2.2. Related works for secured PHR management in cloud

The public key infrastructure is involved in most grid-based implementations as it is which is evenly supported and can be easily combined with various applications on different platforms. An identity-based encryption act as a public key encryption process in which a public key is used as arbitrary string such as a telephone number or email address[20]. Usually, the private key generated by a private key generator, is the powerful of a master secret. By this interference, everyone can verify signatures and encrypt messages beyond the dissemination of public features and the public key "strings" without prior key distribution.

The multi-group-based services can be co-existed on a single network for proliferating on owing to convergence of upcoming mobile technology and wireless technologies which is an emerging communication key on mobile-multicast keys management[21]. Group key management (GKM) keys are inefficient as it overhead rekeying to enhance GKM keys and so, it can multiple in multicast group environments. Secure group communication can be made on a just coughing on single group service for existing GKM. The overhead occurs on low rekey transmission on servicing in a multiple group network across a homogeneous or heterogeneous network. If a wireless network participates on a group network, it can support a single and multiple movement member. A multicast group namely slot basis multiple GKM scheme for a multiple multicast group. The various kinds of problem on key management for multi cast communication sessions are discussed.

The investment pressure and single point of failure rekeying at a network causes signal loading if it mitigates one affect n-phenomenon. Symmetric polynomial based dynamic conference scheme (SPDCS) is a special application for approximation of SGC application a well-known technique[22]. The formation of privileged subgroups to allow arbitrary subsets of users to make conference schemes in resulting to the extension of SPDCS. An important problem raising is dynamic and secure multicast communication which

has to maintain a lower computation and storage complexity is to construct a centralized group to concurrent the distribution of protocol. It helps to propose an efficiency of centralized group by minimizing the computation key cost for key server (KS) on updated keys. It can dynamically perform busty operation effectively when a user joins or leaves in a group.

3. Proposed methodology for enhancing the security in cloud

Most of the cloud carriers offer primary key encryption schemes for protecting data or may additionally go with the user to encrypt their own records. Either way, there is a want to encrypt facts that is involved inside the cloud. But the strategies for coping with the keys which are used for encryption, garage places of the keys, accessing authority of the keys and approach for recovering information at some point of the loss of keys are the most important tasks within the key management process.

The intelligent key management is a key component of the overall record management process within the cloud. There are actually three types of cloud protection examples. First, the secret to free encrypted statistics is kept in the same cloud. In the second example, companies hire vendor responses that house the important item in a hidden venue. The third state of affairs is to protect the valuable stuff on-site inside the company. This is a very high-priced job due to the need for an unnecessary degree of protection. But none of these options are feasible or ideally tailored to existing employer needs. When saving data in the cloud, more than one key can be used for the protection of various types of information. Owing to key security issues, a substantial number of company statistics are routinely encrypted with encryption keys. Encryption is a powerful method used to encrypt data in the cloud. Encryption guarantees the security of the information at the same time as the key control allows the restricted information to be entered. Key controls take additional time, but they could be streamlined.

The portability of the numerous encryption tools for outstanding cloud environments is getting extra complicated. Encrypting statistics on one cloud issuer and decrypting on another cloud business would not work if each platform uses a special proprietary key management system. Without a third-party key control scheme, transferring data between cloud services involves a complete decryption of all the information. This often ends in the security hazard at some stage in the information sharing process. The business agency must select the required main control status depending on the form of data and the degree of risk tolerance.

3.1. Securing the key store

The key store itself should be protected from the malicious users. If the malicious user has access to the passwords, they will be able to access any encrypted data related to the corresponding key. Key stores themselves must then be protected by packaging, transport and backup media.

3.2. Accessing the key store

The access to the main store can be limited to users who have the competent authority to enter the data. The classification of consumer positions must be used to promote the control of access to the facts. The entity that uses the key does not be the entity that stores the key.

3.3. Backup and recovery of keys

There is a need for secured backup and recovery methods for cryptographic keys, in the course of critical circumstances like the loss of main results in the destruction of private business enterprises. The cloud providers should avoid key loss by means of backup and restore processes.

4. End to end intelligent encryption methodology

The current focus is just on side-channel evaluation and implementation efficiency. However, during access control, there is indeed a lack of security. In different locations, the AES key is stored. If an attacker

can get there, he can definitely go somewhere to get the whole AES key. Human beings cannot store all of the keys so that they have a lot of other methods to link. So, in one position he must store the AES keys. Now a day, the attacker would actually go to a private key spot to get the clear text out of the chip text rather than the other attacking ways. So, recommend a system in place to ensure the secure storing of AES in an effective manner, with a view to avoiding attacks of this kind. The **Figure 1** illustrates the entire technology process under which the honey retains the AES key. For a random generator, honey requires two different keys.

The primary aim of the encryption method is to provide necessary protection, monitor entry, and confidentiality of data to cloud-based facts. The plain textual material in cipher text (encrypted form) is a conversion mechanism that occurs in the encryption system. The encryption solution provides high security, access to management, and protection of cloud data. It also offers authentication, assurance of the sender of the message, credibility after sending, evidence that the substance of the message has not been modified, non-repudiation cannot refuse the sending of the message. The most efficient way to ensure information confidentiality is to encrypt it by transmission. Standard information (plaintext) is translated to cipher text (encrypted form) using the encryption algorithm and the encryption key.

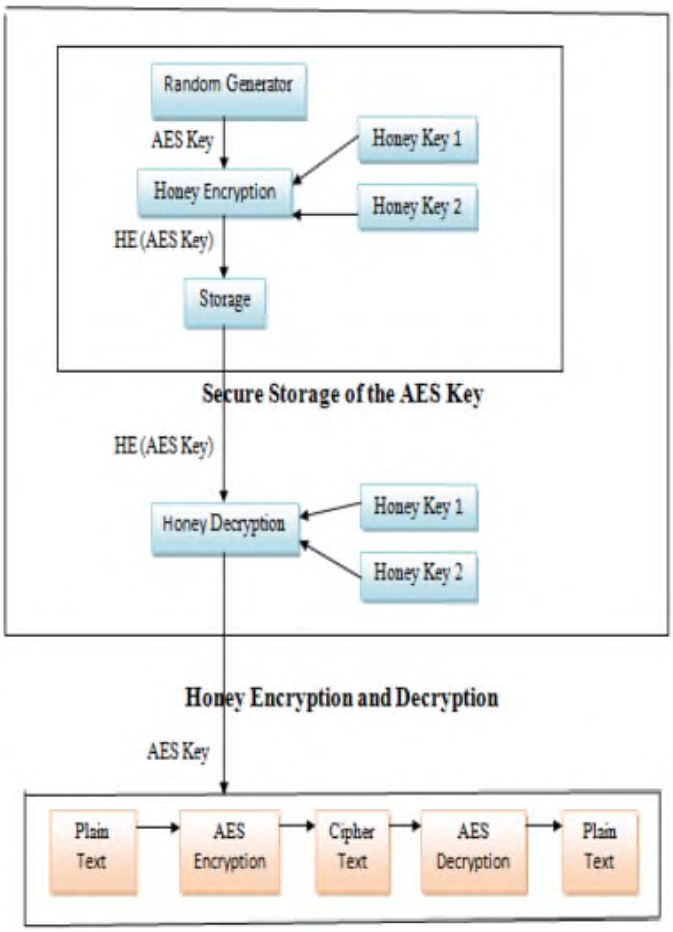

Figure 1. Honey-encryption using round key AES algorithm.

The asymmetric encryption is referred to as public-key cryptography, which uses two keys on both public and private clouds. Public key statistics are exchanged with others, but they need to be kept secure in the private key. Both public and private keys are used to encrypt an RSA algorithm message. This characteristic can no longer provide handy anonymity, but can also provide honesty, credibility and non-reputability. Secret is used in symmetric encryption to encrypt and decode a letter or file. AES is not like the methodology of DES in several respects. Asymmetric encryption is not easier than inconsistent encryption. The most commonly used current algorithm is the advanced encryption standard (AES), which is necessary to protect the data.

4.1. Generation of random keys

The random key wanted for the 128-bit AES key's 44 phrases and for this it takes four words as input and for the 192-bit 6-word enter and for the 256-bit 8-phrase input. Two keys are wished for honey technology, which is likewise generated from the key growth algorithm.

4.2. Key storage by honey

Honey encryption has been used to store what is essential. It needs to be encrypted and preserves the AES key on one location. The sweet set of rules needs key, which is generated and stored by a random generator algorithm. Sweet encryption is done more efficiently until the authentication procedure needs to be carried out for each letter, thereby reducing the issue of the sweet solution.

4.3. Process of honey-encryption algorithm (HEA) and honey-decryption algorithm (HAD)

Take, for example, the 128-bit AES key. This has to be encrypted with the aid of honey. One of the keys is to set the AES 128-bit key. The remaining keys are the fake key, which is also the 128-bit length key. The seed area is a kind of index price for the hash function. So, by using DTE, we can map the AES 128-bit key to the seed field.

The **Table 1** displays the honey encryption algorithm. Honey coding can be used once. Honey decryption is done for each and every AES encryption or decryption device. The honey algorithm calls for the encryption and decryption keys. The **Figure 2** explains the honey encryption and decryption process in corresponding sender and receiver nodes.

Table 1. Tabular representation of honey encryption algorithm.

Message	CDH	Seed space	Secret key	Result
Message 1	3/8	111	Addition of key	Cipher text
		110		
		101		
Message 2	2/8	100		
		011		
Message 3	1/8	010		
Message 4	2/8	001		
		000		

Figure 3 is explaining just the method of mapping kernel space and main sets. The phases of the honey method are seen in **Table 2**. Honey encryption is provided with the username and the key AES (HE). The key is just a sweet-screen key. The seed space shall substitute the AES key to give the seed value (SV). R is an alleged key provided by a key algorithm for expansion. After it is combined, the hash value (H) can be obtained from R and key. Then XOR can be done to build a cipher key between S' and SV (C). The opposite form of sweet encryption, honey decryption (HD). The decode process for honey decryption will take place.

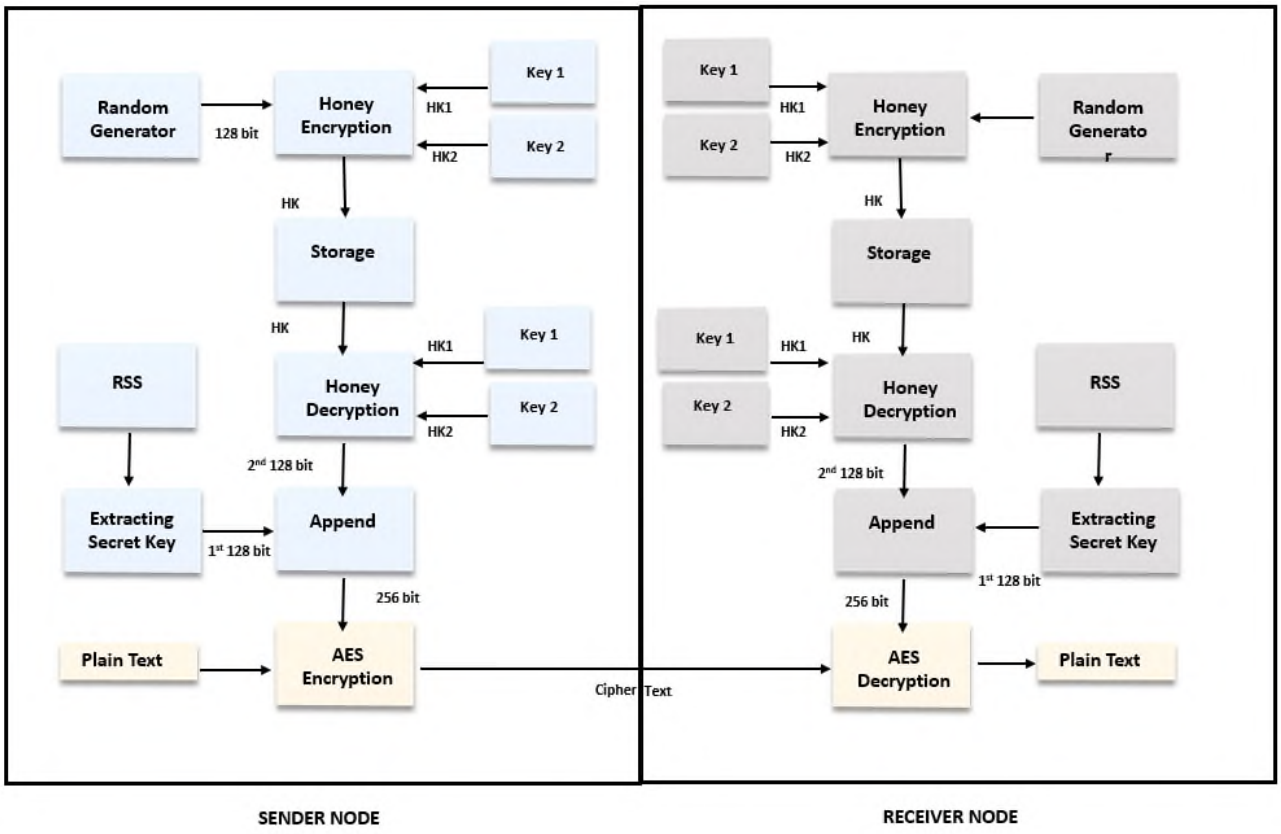

Figure 2. Honey encryption and decryption process.

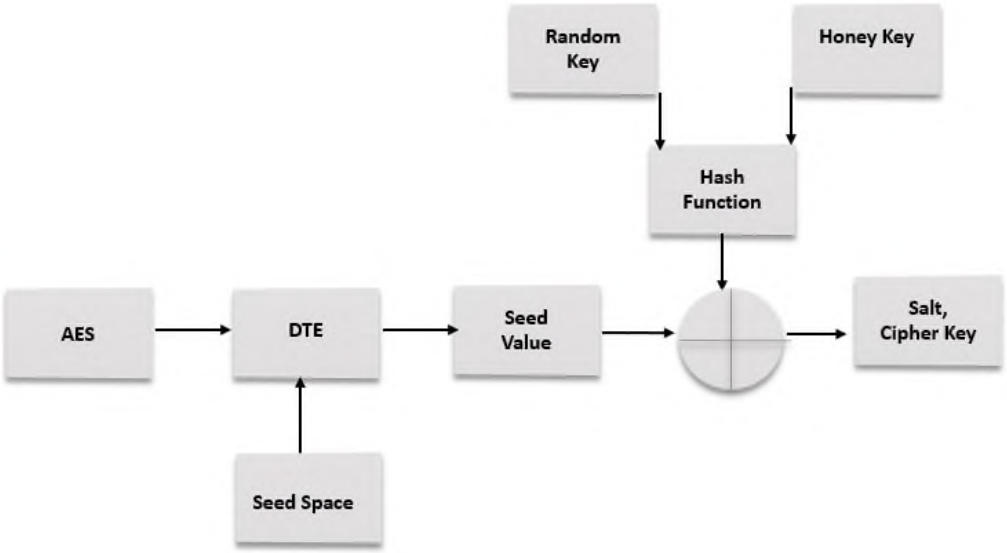

Figure 3. Process of seed space and key set mapping.

Table 2. Algorithm for honey encryption and decryption.

Honey encryption algorithm	**Honey decryption algorithm**
Henc (K,M)	HDec (K, (R,C))
S← $encode(M)	S'← H(R,K)
R←${0,1}^{n}	S← C ⊕ S'
S'←H (R, K)	M← decode(S)
C← S'⊕S	return M
return (R,C)	-

5. Improved zealous method with anonymization and encryption for confidentiality

The suggested architecture offers a solution by combining dataset anonymization and encryption to protect sensitive personal data privacy. The first user will get in touch with the server to submit the information necessary for assigning a key to the dataset record. Multiple intermediate datasets are produced in this manner. Every ID is then put through the anonymization procedure after that. For easier identification among the datasets, each ID is given a special identifier. They are prioritized depending on the user's use level, such as which dataset is visited more or less, and are then followed by IDs. The dataset that was anonymously ranked one is encrypted. The encrypted file and key are given to the user, who then uses the key to decode the original file. The suggested system maintains dataset privacy. The following are the elements of the suggested architecture.

User interface-it gives the user a view of the fields that are present in a data collection. The user then chooses the necessary fields to produce an intermediate data collection.

Anonymization-for each intermediate dataset, the server anonymizes the data. The dataset's fields are then each given an ID.

Ranking datasets-based on how often a user accesses a certain intermediate data collection, a rank is given to it.

Dataset encryption-this occurs after anonymization, with the highest-ranking intermediate dataset being encrypted before generating the cypher dataset and being delivered to the user.

Dataset decryption-before evaluating the dataset, decryption is carried out on the user's end.

Improved zealous method

MongoDB is used to store and process the datasets. The dataset is accessible to administrators and analytics users as well as service providers. Such datasets are processed with the use of sudo commands. The dataset's accessible fields will be presented throughout the user login process, allowing the user to choose the necessary fields from that list. The outcome is the generation of the intermediate dataset. As no one will be able to identify the user participating in the processing, anonymization conceals information about a person's identity. The modified zealous technique is used to assign IDs to intermediate datasets for each field. The algorithm's steps are listed below.

1) Dataset H, distinguishing elements m, Laplace distribution, threshold TC, and threshold TN
2) Set Hu contains m unique objects for each user.
3) Create a pair (K, CK) using the items you choose, where k stands for an item and CK is the number of people that have item k in their set Hu, which is the original set.
4) Remove the pairings (K, CK) such that count CK is smaller than count TC from this collection.
5) Find a number k at random for each pair (K, CK) in the set. This is selected using the Laplace distribution Lap( ) and CK with k added. As a consequence, the count is noisy: CNK = CK + k.
6) Remove the value (K, CK) from the set using the noisy count formula CNK = TN.
7) The noisy counts are provided together with the remaining items.
8) By fusing a count with a broad term, create ID.

6. Performance evaluation metrics

The performances of proposed techniques are evaluated using various metrics in the section. The metrics used for evaluating the proposed approach is as, Time average and query processing time. The PHR dataset used in this research, the dataset is originally from the national institute of diabetes and digestive and kidney

diseases. The results are obtained for the proposed method along with the existing methods of dynamic searchable symmetric encryption (DSSE), security Hadoop distributed file system sec (HDFS) and tree-based access control (TBAC).

6.1. Query processing time

The query processing time is defined as the total time required for the user requests to be resolved and the individual attributes to be executed. The processing time of the question is dependent on three steps.

- Parsing and translation
- Evaluation
- Optimization

The query-execution time requires three major stages, including the query-evaluation plan, the query execution plan, and the query response plan. In accessing the PHR data, the above steps are involved. **Table 3** represents the evaluation of query processing time. **Figure 4** illustrate the query processing time for varying count of users. **Figure 5** illustrate the overall query processing time for varying methods.

Table 3. Evaluation of query processing time.

Number of users (in count)	**Query processing time (in milli-seconds)**			
	DSSE	**Sec (HDFS)**	**TBAC**	**Proposed**
10 users	3	2.4	2	1.7
20 users	4.1	2.8	2.4	2
30 users	6	5	4.3	3.8
40 users	7.5	6.7	5	4.3
50 users	8.9	7.9	6.9	5.7
60 users	15	13	11	9.8
70 users	17	15	12.8	10.5
80 users	17.9	16.8	14	11
90 users	19.8	18	16	11.9
100 users	23	21.2	18	14
Overall query processing time	122.2	108.8	92.4	74.7

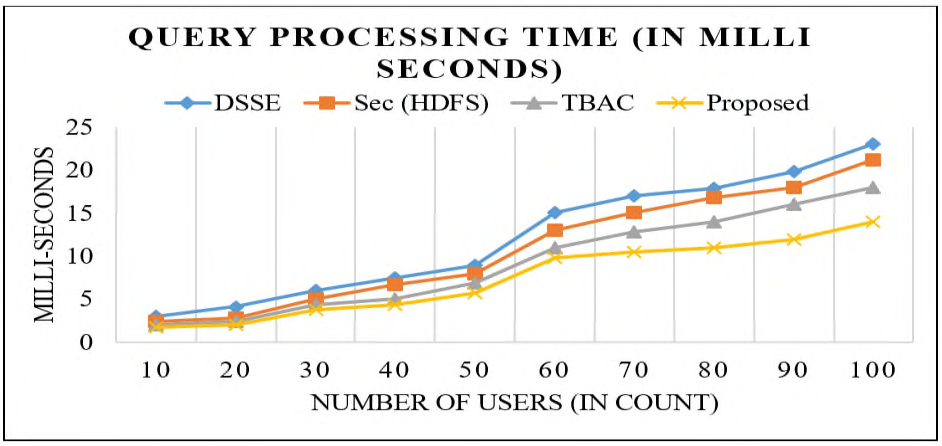

Figure 4. Query processing time for varying count of users.

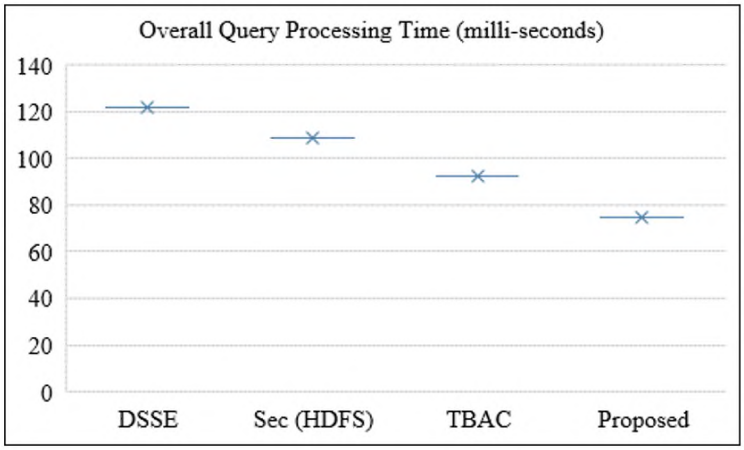

Figure 5. Overall query processing time for varying methods.

6.2. Time average

The time average (AT) is defined as the sum of the total time taken for generating the keys (GKT) in the group which is divided by the number of users (NU). The statistical representation of average time is calculated by Equation (1). **Table 4** represents the evaluation of time average. **Figure 6** illustrate the time average in milli-seconds for varying users in groups. **Figure 7** illustrate the overall time average for varying methods.

$$Time\ average\ (AT) = \frac{GKT}{NU} \tag{1}$$

Table 4. Evaluation of time average.

User group (in count)	**Time average (in milli-seconds)**			
	DSSE	**Sec (HDFS)**	**TBAC**	**Proposed**
100 users	9	5	4	3
200 users	19	10	8	6
300 users	28	20	10	8
400 users	35	25	13	12
500 users	43	29	19	14
600 users	51	30	22	18
700 users	59	33	24	22
800 users	65	37	31	25
900 users	71	42	36	31
1000 users	78	46	40	33
Overall time average	458	277	207	162

As a result, the round key for AES encryption can be screened and all facet channel attacks can be dominated. It should absolutely apprehend the patient-pushed knowledge; patients will have full control of their very own privacy by encrypting their doctor's documents to allow for fine-grained access. The shape tends to have unique difficulties added by diverse physicians and clients; ABE should be used to encrypt patient data so that patients can access male or female clients, as well as different clients, from open spaces with numerous expert parts, capabilities and affiliations. The challenges of accessing the cloud personal health record (PHR) are implemented earlier research. It takes more time for the current works to encrypt and decode the material. To resolve these problems, an approach to access management policy is implemented in the cloud for safe PHR entry.

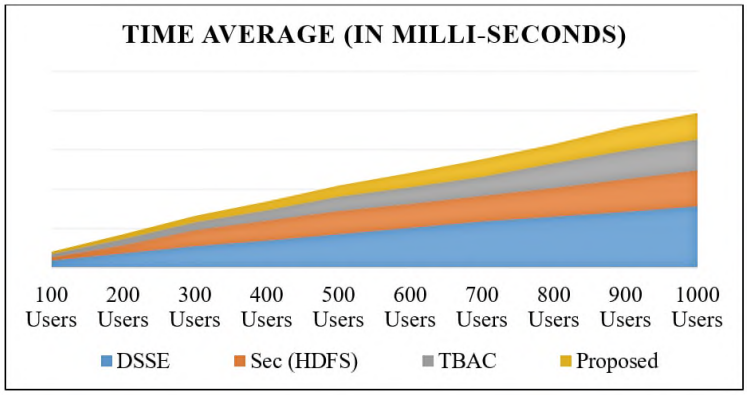

Figure 6. Time average in milli-seconds for varying users in groups.

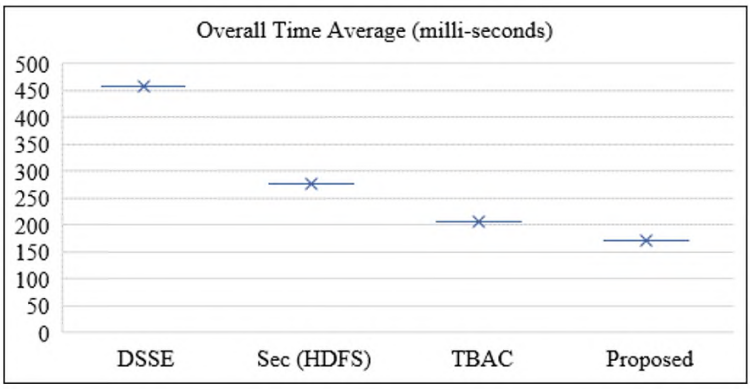

Figure 7. Overall time average for varying methods.

6.3. Result analysis of the anonymization using improved zealous method

The dataset is kept in MongoDB, while Hadoop HDFS and map-reduce are used to accomplish the anonymization procedure. The tasks are written in parallel using the mapper and reducer routines. For a certain dataset, the Hadoop HDFS system distributes the dataset across data nodes. An interface that is linked to the primary database connects to the generated dataset. An anonymization-based confidentiality strategy is evaluated in a map-reduce setting using Hadoop. Hadoop is used to handle a big data collection with 70,000 entries and a size of 2.2 GB, and the times taken to complete anonymization, encryption, and decryption are recorded. The suggested solution, which is based on anonymization, functions well as data volume rises and is processed in the Hadoop environment. **Figure 8** illustrates the execution time with encryption and decryption. The findings show that despite the enormous amount of data, the execution time is relatively short (in milliseconds). The existence of the map-reduce function is the primary cause of the improved performance. When there is a large amount of data, systems without map-reduce functionalities cannot achieve such performance.

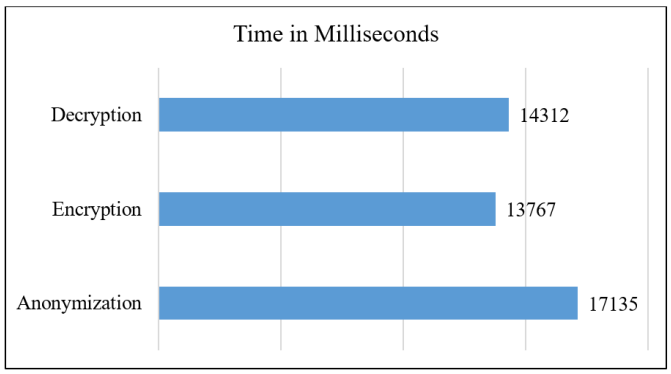

Figure 8. Time taken for decryption, encryption and anonymization.

The previous works rely on data encryption to maintain user privacy, which is a significant drawback from the perspective of the client. In the case of such techniques, the whole dataset is taken into account for encryption, which results in time loss and poor performance. The aforementioned strategy makes use of encryption and anonymization in addition to the ranking technique by using characteristics that show the least and most often viewed files. The administration and processing of the intermediate dataset rank as two of the several types of study that have been done. Sensitive personal information may be masked by anonymization using the modified zealous algorithm. Sensitive personal data will not be revealed even after decryption since it has been anonymized. As a result, the privacy is protected. Only the chosen intermediate dataset is subjected to the full procedure, not the complete dataset. Additionally, the serial feedback technique may be protected from outside threats like brute force. The performance is not affected here, therefore the time gain occurs since only the intermediate data set is handled by Hadoop and NoSQL technologies. While maintaining privacy, the anonymization process may lose its usefulness. The zealous algorithm is selected to increase usefulness while maintaining a high degree of anonymity. After anonymization, the data item is often hard to recover. The suggested solution ensures that data owners always have a backup of their data, making it easy to get the original data when necessary. The suggested approach is thus safe against the deanonymization attack since Laplacian noise with threshold is injected throughout the anonymization process using the zealous technique.

7. Conclusion

By keeping the AES non-public key after encrypting it using the honey encryption, the built-in system has effectively been removed from the private key safe storage of the AES series of rules, which is lacking from the existing system. It is necessary to provide a complete, robust AES algorithm. As can be observed from the contrast graph, the inclusion of the honey approach has no effect on the AES algorithm's overall effectiveness and cost. These solutions were used to successfully address the issues of key control, channel attack facet, and brute force assault. The suggested technique offers a solution by combining dataset anonymization and encryption to protect sensitive personal data privacy. An anonymous ID is given after the sensitive data is anonymized using a modified aggressive algorithm. The data owner encrypts the anonymized intermediate dataset using a serial feedback generation process. Combining a generic keyword with a Laplacian noise count results in the anonymous ID. Sensitive personal data will not be revealed even after decryption since it has been anonymised. As a result, the privacy is protected. Laplacian noise and noise threshold are introduced throughout the zealous algorithm's anonymization phase, making the solution suggested safe against de-anonymization attacks. In future, by using sophisticated encryption algorithms to any deep learning approaches based on safe and protected anonymity better than the original algorithm, this work may be improved in the following ways. Using apache spark in big data for public health records may remove the repeated settings for privacy and security and increase accuracy.

Author contributions

Conceptualization, PM; methodology, PM; software, PM; validation, DSS; formal analysis, DSS; investigation, DSS; resources, JJJ; data curation, PM; writing—original draft preparation, PM; writing—reviewing and editing, PM; visualization, PM; supervision, DSS and JJJ; project administration, DSS. All authors have read and agreed to the published version of the manuscript.

Conflict of interest

The authors declare no conflict of interest.

References

1. Majeed A. Attribute-centric anonymization scheme for improving user privacy and utility of publishing e-health data. *Journal of King Saud University—Computer and Information Sciences* 2019; 31(4): 426–435. doi:

10.1016/j.jksuci.2018.03.014
2. Juels A, Rivest RL. Honeywords: Making password-cracking detectable. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (CCS'13); 4–8 November 2013; Berlin, Germany.
3. Alvarez R. The electronic health record: A leap forward in patient safety. *Healthcare Papers* 2004; 5(3): 33–36. doi: 10.12927/hcpap.2004.16862
4. Mewada A, Gedam P, Khan S, Reddy MU. Network intrusion detection using multiclass support vector machine. *International Journal of Computer and Communication Technology* 2010; 1(4): 7. doi: 10.47893/IJCCT.2010.1054
5. Cao B, Xia S, Han J, Li Y. A distributed game methodology for crowdsensing in uncertain wireless scenario. *IEEE Transactions on Mobile Computing* 2020; 19(1): 15–28. doi: 10.1109/TMC.2019.2892953
6. Boneh D, Boyen X, Shacham H. Short group signatures. In: *Lecture Notes in Computer Science*, Proceedings of the 24rd Annual International Cryptology Conference; 15–19 August 2004; California, USA. Springer; 2004. Volume 3152, pp. 41–55.
7. Bustamante C, Garrido L, Soto R. Fuzzy naive bayesian classification in robosoccer 3D: A hybrid approach to decision making. In: *Lecture Notes in Computer Science*, Proceedings of the RoboCup 2006: Robot Soccer World Cup X; Berlin, Germany. Springer; 2007. Volume 4434, pp. 507–515.
8. Chase M, Chow SSM. Improving privacy and security in multi-authority attribute-based encryption. In: Proceedings of the 16th ACM Conference on Computer and Communications Security; 9–13 November 2009; Chicago, USA. pp. 121–130.
9. Li Y, Liu J, Cao B, Wang C. Joint optimization of radio and virtual machine resources with uncertain user demands in mobile cloud computing. *IEEE Transactions on Multimedia* 2018; 20(9): 2427–2438. doi: 10.1109/TMM.2018.2796246
10. Chatterjee R, Bonneau J, Juels A, Ristenpart T. Cracking-resistant password vaults using natural language encoders. In: Proceedings of the 2015 IEEE Symposium on Security and Privacy; 17–21 May 2015; Washington, USA.
11. Chen YY, Lu JC, Jan JK. A secure EHR system based on hybrid clouds. *Journal of Medical Systems* 2012; 36(5): 3375–3384. doi: 10.1007/s10916-012-9830-6
12. Danwei C, Linling C, Xiaowei F, et al. Securing patient-centric personal health records sharing system in cloud computing. *China Communications* 2014; 11(13): 121–127. doi: 10.1109/CC.2014.7022535
13. Kwon D, Natarajan K, Suh SC, et al. An empirical study on network anomaly detection using convolutional neural networks. In: Proceedings of the 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS); 2–6 July 2018; Vienna, Austria.
14. Goyal V, Pandey O, Sahai A, Waters B. Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM Conference on Computer and Communications Security; 30 October–3 November 2006; Alexandria, USA. pp. 89–98.
15. Hababeh I, Gharaibeh A, Nofal S, Khalil I. An integrated methodology for big data classification and security for improving cloud systems data mobility. *IEEE Access* 2019; 7: 9153–9163.
16. Leng C, Yu H, Wang J, Huang J. Securing personal health records in the cloud by enforcing sticky policies. *Indonesian Journal of Electrical Engineering and Computer Science* 2013; 11(4): 2200–2208. doi: 10.11591/telkomnika.v11i4.2406
17. Li M, Yu S, Zheng Y, et al. Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. *IEEE Transactions on Parallel and Distributed Systems* 2013; 24(1): 131–143. doi: 10.1109/TPDS.2012.97
18. Hemant P, Chawande NP, Sonule A, Wani H. Development of server in cloud computing to solve issues related to security and backup. In: Proceedings of the 2011 IEEE International Conference on Cloud Computing and Intelligence Systems; 15–17 September 2011; Beijing, China. pp. 158–163.
19. Razak SA, Nazari NHM, Al-Dhaqm A. Data anonymization using pseudonym system to preserve data privacy. *IEEE Access* 2020; 8: 43256–43264. doi: 10.1109/ACCESS.2020.2977117
20. Suh SC, Kim H, Kim J, et al. An encoding technique for CNN-based network anomaly detection. In: Proceedings of the 2018 IEEE International Conference on Big Data; 10–13 December 2018; Seattle, USA.
21. Shekelle PG, Morton SC, Keeler EB. Costs and benefits of health information technology. *Evidence Report Technology Assessment* 2006; 132: 1–71. doi: 10.23970/ahrqepcerta132
22. Zhao G, Rong C, Li J, et al. Trusted data sharing over un-trusted cloud storage providers. In: Proceedings of the 2010 IEEE Second International Conference on Cloud Computing Technology and Science; 30 November–3 December 2010; Indianapolis, USA. pp. 97–103.