Privacy Preservation in Healthcare Monitoring System

Mangore Anirudh K, M Roberts Masillamani

Abstract: Indexing might make a technique to process of Information confidentiality; acceptability and integrity with healthcare which is necessary to get the thought for engaging the healthcare which needs to get the thought for enabling and empowering the healthcare system progressively reliable, increasingly successful as far as improvement of medical science and relieving large number of patients at once forecasting the possible medical problem and diseases. There square measure numerous challenges to implement indexing algorithmic rule like execution time, memory demand, and computation power. So as to make sure the consistency and nobility of big data in Health monitoring system. We are using the HMAC-SHA256 and Selective Encryption Algorithm to develop security on healthcare dataset and authentication of the entire Healthcare dataset. In the paper, the experimental demonstrated that a values of the HASH will change if the dataset turned into a distinct value. Therefore, it can firmly demonstrate that our proposed algorithm can help to check the consistency and integrity of dataset before access.

Index Terms*: *Cloud Computing, Healthcare, Medical, Privacy and Security, SS-Tree.

I. INTRODUCTION

To maintain the patient's treatments information, health monitoring dataset uses the way of electronic. Hence, the integrity and security of the patient healthcare records is essential, to guarantee that the information isn't obtain by unauthorized institutions throughout the network transmission and its consistency and fulfillment when transmission, it is important to embrace the authentication and encryption measure. The access to this sort of a worldwide data and correspondence foundation by with an advance in storage space and digital sensors had been created very big quantity of data, for example sensor, Internet, mobile device data or streaming. Moreover, in numerous fields of knowledge the reason for investigation is data analysis, for example engineering, science and also management. In the contrast of web-based big data, User Enrolment Id data is all-important segment of portable enormous information, which are outfit to customize and optimize mobile services

Revised Manuscript Received on December 22, 2018.

Mangore Anirudh K, Department of Computer science and Engineering, Bharath Institute of Higher Education and Research, Chennai (Tamil Nadu), India.

M. Roberts Masillamani, Department of Computer science and Engineering, Bharath Institute of Higher Education and Research, Chennai, (Tamil Nadu), India.

II. RELATED WORK

In the previous systems database review had concentrated on indexing lower dimensional data and on various sorts of questions other than closeness inquiries. For indexing of multidimensional data for nearest neighbor queries, the Ic-d structure was developed [17]. As of late, this structure had been utilized in geographic data frameworks for questions like similitude inquiries [18], and may be valuable for closeness ordering. Different techniques, for example, linear quad trees, grid files [19], and space filling curves [20], as it failed to scale with parameter of high measurement, however might be valuable for medium dimensional data.

To indexing high dimensional information in the literature of database, the [21] R-tree and its variety such as R*-tree [22], have been used frequently. In any case, since ranges are put away on every measurement, the index needs maximum existence to look in greater dimensionality. Therefore, before indexing in R-trees excessive dimensional information ordinarily tying strings to a lower dimensional space [4, 23].

The TV-tree[25] is the main strategy for recording high - dimensional information in the database writing up to this point. Correlations of execution clearly show that the TV-tree can be much more productive than the R*-tree. Nevertheless, the improved execution is based on two hypotheses. The main supposition is that measurements and the component vectors are requested by "importance". This second suspicion is that arrangements of highlight vectors in the dataset will generally coordinate precisely on dimensions, especially on the initial few "critical" dimensions.

Since appropriate transformation may be used, the main assumption is reasonable.The second suspicion has not been expressed explicitly, In the paper, but a cautious examinatio n of their calculations reveals that their improvement in exec ution is based on it.The first component vectors contain a sm all arrangement of discrete amounts in certain applications, s o the second suspicion holds.

Unfortunately, this second presumption in visual data fra meworks and numerous different applications will typically not be valid.Highlights in these applications are generally va lued genuinely, so there are insignificant chances of precisel y coordinating measurements. For this situation, the TV-tree lessens to a record on just initial couple of measurements. Small changes in the proposed calculations should allow the TV-tree in these applications to be a humble improvement over the R*-tree.

In this paper, however, we will allude to the R-tree (and variations) as the most recently known structure for comparability ordering since it has proven itself in larger applications, unfortunately.

There is also related work outside the literature of the dat abase.

Work has been done on group files in the information retriev al literature [10, 11] proposing structures [25] like the SS - tree.A static indexing structure dependent on Kohonen net works has been proposed in the image database network[26] In the literature on computational geometry and vector quant ization, there is additionally related work[12].

To ensure consistency and integrity in network transmissi on of XML Electronic Medical Records[29].This system us ed the HMACSHA256 algorithm to consider "computerized unique fingerprinting" and to confirm the entire XML Electr onic Medical Records. The result in the system compared th e three experimental data groups showing that the HASH val ues will change if the PIN or XML Electronic Medical Reco rds are transformed into a replacement esteem.

In [30], creator talk is considered about the routing approach in specially designated portable systems from the security perspective, breaking down the dangers against ad hoc routing protocols and introducing the necessities that should be targeted for secure routing. Existing secure routing protocol is either proactive or responsive in nature for mobile ad - hoc networks. In this system, two procedures are used, in particular HMAC-SHA256 for furnishing upright information alongside confirmation and trust based framework to gradually secure the system by anticipating Denial of Service attack in the network. In particular HMAC-SHA256 for furnishing upright information alongside confirmation and trust based framework to gradually secure the system by anticipating Denial of Service attack in the network.

III. PROPOSED SYSTEM

From literature survey, it is observed that it is impossible to every patient to carry their medical report and all medical related documents along with them every time to solve this problem we proposed new system which consists cloud based health monitoring privacy preservation for protect the patient confidential data and medical reports and all medical related data are store on cloud which is accessible in everywhere in world.

To provide patients with faster relief by providing evidence-based medicine that detects diseases in the preceding stages depending on accessible clinical information, limiting medication doses to evade reaction and giving effective drug dependent on genetic structure. The health records have to be private and secure. The Health cover Portability and responsibility is a set of rules on who has access to the patient's health report. In order to comply with the rules and regulations whenever there is an emergency by using these system appropriate medical professionals who can access that patient's health data. This aides in diminishing readmission rates in this manner reducing cost for the patients. Predicting before spreading the viral diseases depending on the live examination. This can be distinguished by analyzing the social logs in a specific geo - user enrollment I d of patients suffering from a disease. This encourages healthcare professionals to take the necessary preventive measures to encourage the victims.

Observe whether the hospitals are set up in accordance with the Indian medical council's standards. This periodic medical helps government to take essential methods against hospitals being disqualified. Customized patient treatment continuously observed the effect of medication and can be changed for faster relief depending on the analytical dosage of medication. Observing vital patient signs to provide patients with proactive care. Examining the data produced by patients who had previously experienced similar symptoms helps doctors to provide new patients with effective drugs.

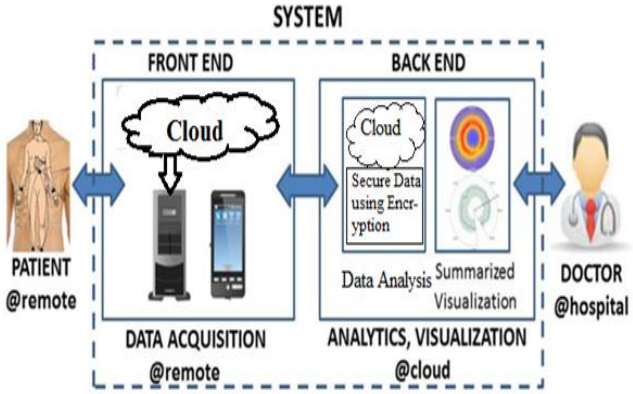


Fig.1 System Architecture in healthcare System

In healthcare system increases the concern of its security and privacy because of big data. In cloud based healthcare, the data of the patient is stored remotely on cloud big data infrastructure. Along these lines, it is required to guarantee the privacy of the data which is stored in cloud. Hence, big data governance is fundamental begin to revealing data to analytics.

Following figure shows the flow of the Healthcare Monitoring system and in this used each algorithm describes in below.

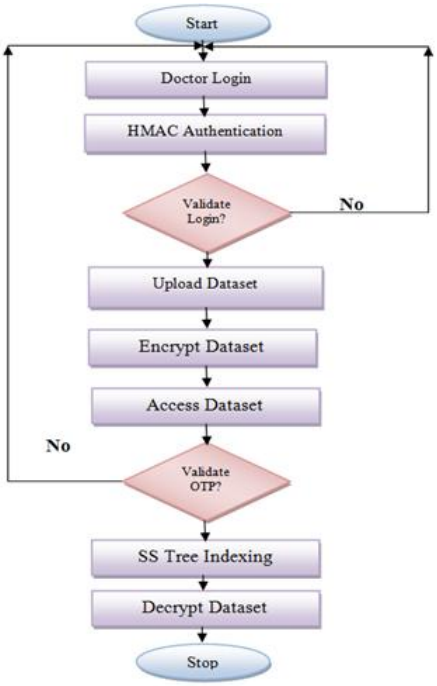


Fig 2: Flow Diagram of Proposed System

In healthcare monitoring system, security purpose we used HMAC-SHA 2 Authentication in this HMAC-SHA 256 generate hash key if any user try to access the system that time hash key get automatically changed and failed to login.
Doctor can upload data of patients this is very confidential data that's why this is necessary to encrypt dataset for privacy and security purpose. This data without decryption not possible read or access data. Healthcare monitoring system generate huge amount of data that data retrieve from dataset is time consuming task that's why we implement SS-Tree algorithm which helps to get data in minimum time of process.

A. HMAC-SHA 2

HMAC-SHA256 is a kind of keyed hash calculation created from the hash capacity of SHA-256 and utilized as a Hash - based Message Authentication Code (HMAC). The HMAC system mixes a mystery key with the message information, hashes the outcome with the hash work, again mixes the hash an incentive with the mystery key, and after that the hash work is connected a second time thereafter. The length of the yield hash is 256 bits[29]. A HMAC can be utilized to decide if a message sent over a dangerous channel has been modified, as the sender and beneficiary offer a mystery key. The sender figures the hash an incentive for the first information and sends as a lone message the hash an incentive just as the first information. The beneficiary recalculates the hash an incentive on the got message and watches that the determined HMAC compares to the transmitted HMAC[29].

Any adjustment in the information or hash esteem results in a misalignment, since learning of the mystery key must change the message and reproduce the right hash esteem. The message is authenticated[30] if the genuine and registered hash esteems coordinate.

HMACSHA256 acknowledges keys. It very well may be of any size this key. It creates a 256-piece hash arrangement long.
HMAC utilizes two goes for hash computation. The mystery key is utilized to decide within two keys first and the external of the second. The calculation's first pass produces an interior hash from the message and the inward key. The second pass makes the last HMAC code from the result of the inward hash and the external key. The calculation gives better insusceptibility to longitudinal augmentation assaults along these lines [30].
The message isn't encoded by HMAC. Rather, the message must be sent close by the HMAC hash. The mystery key gatherings will hash the message themselves once more, and in the event that it is credible, the hashes got and determined will coordinate [29].

$$HMAC(K,m) = H((K' \oplus opad)||H((K' \oplus ipad)||m))$$

Where,
H is a cryptographic hash function,
K is the secret key,
m is the message to be authenticated,
K' is secret key, derived from the original key K (Padding K to the right with additional zeroes to the h ash function's input block size or hashing K if it is longer tha n that block size)
|| denotes concatenation,
⊕ denotes exclusive or (XOR),
opad is an external padding (0x5c5c5c ... 5c5c, hexadecimal constant one block long), and ipad is an internal padding (0x363636 ... 3636, hexadecimal constant one block long).

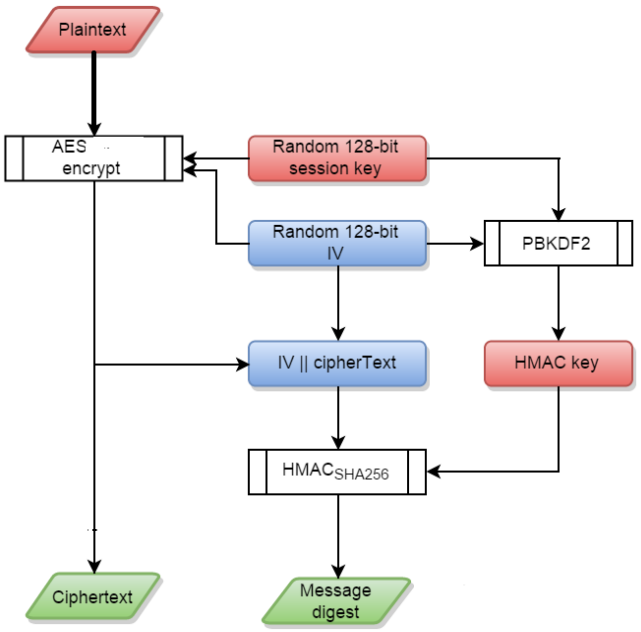


Fig 3: HMAC-SHA 256 Algorithm

Algorithm:

Function hmac
Inputs:
key: Bytes //array of bytes
message: Bytes //array of bytes to be hashed
hash: Function //the hash function to use
block Size: Integer // underlying hash function block size
output Size: Integer //the output size of the underlying hash function
//Keys longer than block Size are shortened by hashing them
if (length(key) > block Size) then
Key ← hash (key) //Key becomes output Size bytes long

//Keys shorter than block Size are padded to block Size by padding with zeros on the right
if (length(key) < block Size) then
Key ← Pad(key, block Size) //pad key with zeros to make it block Size bytes long
o_key_pad = key xor [0x5c * block Size] //Outer padded key
i_key_pad = key xor [0x36 * block Size] //Inner padded key
return hash (o_key_pad ‖ hash (i_key_pad ‖ message)) //Where ‖ is concatenation

B. Selective Encryption Algorithm

AES - Rijndael handles data in 4 social occasions of 4 bytes with 128/192/256 piece keys and 16 byte data, working a whole square in each round. Around at that point, AES is seen as wrong for visual data, for instance, propelled picture on account of the long system of figuring. Late advances in programming and gear improvement have achieved achieving the perfect execution rate once we

comprehend the data status scale by executing our SEA count structure. The result shows that the degree of the data state between 20 × 20 to 30 × 30 may take as pitiful execution time as could be normal in light of the current situation. We suggested a novel encryption count considered SEA in this paper is explicit and improves the AES computation. Sea's Architecture is showed up in Figure 4. The Architecture empowers one to perform center reasoning about our count is an optional course realized by the fragment Selector showed up in Figure 4. Since the present example of framework - wide remedial picture transmission is developing. There are some extraordinary sorts of cutting edge visual data, for instance, picture, sound, content record, video, and so forth. As we likely am mindful, there are different sorts of stages over the wire/remote framework from different contraptions. Thusly the selector part plays out the selector work where it is optional to pack the plain substance or unrefined picture noted as Cyn, the degree of the information state noted as InpS, the proportion of the key noted as KeS and the amount of rounds noted as Rn [31].

Study that assurance from cryptanalysis strikes from AES-set up together encryption depends inside and out concerning the Rn used. The weight part that we proposed in our figuring using Huffman coding to thoroughly diminish the Rn used just like keeping less execution time. Meanwhile, we use compacted data as data state to improve AES's assurance from breaking strikes. The piece of the Huffman blower is given in our arrangement count. The state - transformation work, an immediate limit, may be optional to consolidate into the proposed figuring a negative rotate by 90 degrees in the data state. Since the unrefined AES estimation contains numerous twofold stream codes, it takes a lot of time in the midst of its utilization state. In like manner, our proposed figuring performs methods for unrolling and solidifying that supplant the twofold course codes in order to keep up its base execution time [31].

Algorithm:

```
int i;
for (i = 1; i < BS; i++)
{
newData[0]= holder[i][(0+i) % BS];
newData[1]=holder[i] [(1+i) % BS];
newData [2]=holder[i] [(2+i) % BS];
newData [3]=holder[i] [(3+i) % BS];
holder[i][0] = newData[0];
holder[i][1] = newData[1];
holder[i][2] = newData[2];
holder[i][3] = newData[3];
}
```

Following Figure shows the working of selective encryption algorithm.

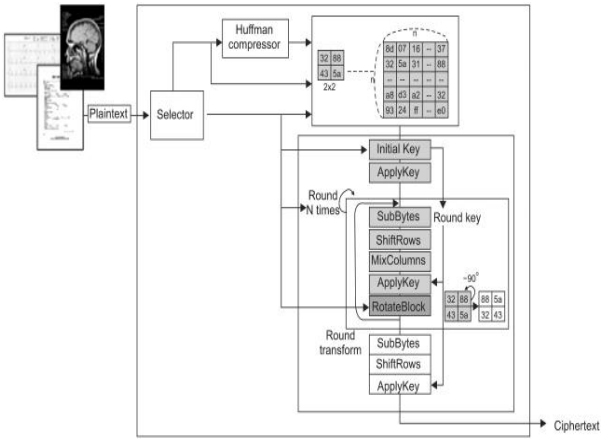


Fig 4: Architecture of selective encryption algorithm.

C. SS-Tree Algorithm

Our essential application territory for indexing is secure and fastest data retrieval from huge amount of databases, this database contains personal information of patients. Hospital database includes personal information, Heath data, Medical Report, Images, X-rays, etc.

We rely on a domain expert to help in the indexing process t o use the SS - tree, as already mentioned in figure 3.

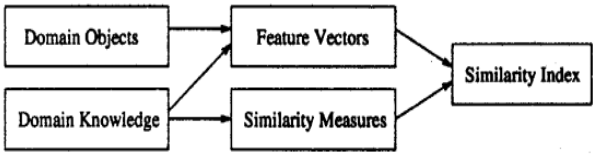


Fig 5: Illustration of dependencies in the creation of similarity index

Feature vectors must be provided in a format to approximate the desired measure of dissimilarity by a weighted Euclidean distance metric on those features.

Learning the space domain to be used to restrict the kinds of measurements of similarity between feature vectors that are normally used in queries. This learning could then be used to tune the SS - tree (or different SS - trees) execution.

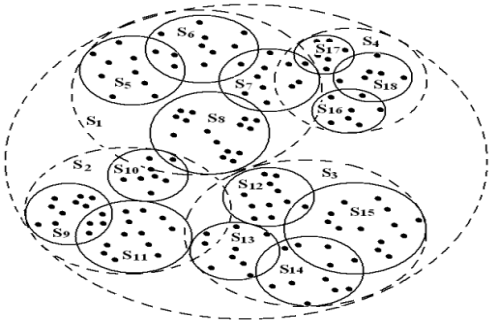
The essential objective of comparability indexing is equivalent to other indexing techniques: to limit normal and most pessimistic scenario time required for queries tasks. In addition, structures that help dynamic updates and have effective disk - based usage are favored as many applications require these features to scale up to huge databases.

SS – tree [5] is an R-tree variation. It uses the super spheres rather than the super rectangles to deal with the nodes and is a totally powerful and dynamic index structure as a R-tree.

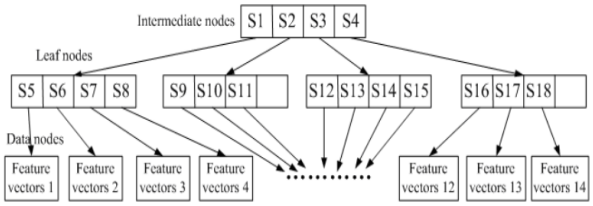
1. The structure of SS-tree

The SS - tree has similar structure and characteristics as the R - tree. It mainly consists of the intermediate node, the leaf node and the data node in our structure. The data objects are placed in the data node and each intermediate node and leaf node is created by the super spheres that completely enclose each of the super spheres of their lower level node, as shown in Figure 1. As shown in Figure 4, each node in this structure consists of a variety of SSTreeElem structure. Each structure of SSTreeElem speaks to a node entry. In Figure 4, childptr speaks to its child node to the pointer of this entry ; update count is used to recalculate the section occasionally when its child node is

changed ; range is the span of its child node's encasing sphere ; and centroid [Dim] is the mean estimate of all the centroids of its child entry.



(a): The data set of feature vectors



(b): The R-Tree structure of (a)

Fig 6: Example of SS-Tree structure

Structure SSTreeElem
{
BYTE *childptr; // Child Pointer
int update count; //Refresh Value
float radius; //The Radius enclosing sphere
float centroid[Dim]; //The mean value of its child entrie's centroid
}

Figure 6 shows one case of development of the SS - tree. Each point in Figure 6(a) represents one vector element and each super sphere represents one node section. The list development shown in Figure 6(b) is the related structure of the SS - tree for Figure 6(a). Give M a chance to be the maximum number of entries in a single node, and the minimum number of entries in a single node. In Figure 6(b), m=2 and M=4 are present here. We can find it in paper [1] and paper [5] for a more detailed description of the properties, update the calculation of SS - tree. The split algorithm used here finds the dimension with the highest variance and selects the split location to minimize the sum of the variances on each side of the split. [5].

2. Nearest Neighbour Queries

Give a feature estimate area D—indexed element vectors, a query vector Q ⊕ D, and a whole number k⩾1, the KNN query NN(Q, k) select the component vectors k ordered with the shortest distance from Q[9].

MINDIST algorithm [9] is one of the most popular neighbor search algorithms in the neighborhood. The MINDIST (Minimum Distance) of a point Q(q1, q2, ..., qd) in Euclidean space E(d) from a sphere S(O, r) (where O(o1, o2, ..., od) is the center of the circle, and r is the span of the circle) in a similar space is:

$$MINDIST(Q,S) = \begin{cases} 0, if\ d(Q,O) \le r \\ d(Q,O) - r,\ others \end{cases}$$

Where,

$$\sum_{i=1}^{d} (|\, q_i - o_i \,|)^{2}{}^{1/2}$$

- Instead of bounding rectangles, SS-tree uses bounding spheres
- Motivated by range and kNN queries which are hyper-spheres
- Center of sphere is centroid of points
- Maintains total number of points in the sub tree
- Higher fan-out due to smaller storage requirements
- Considerable volume overlap
- Split axis is selected depend on variance
- Better than R*-trees.

Bounding Spheres and Bounding Rectangles:

- SS - tree has shorter diameter regions
- R*-tree has smaller volume regions
- How?
- Rectangles in d-dimensions
- Diameter: Between 0 and $\sqrt{d}$
- Volume: Between 0 and 1
- Spheres in d-dimensions
- Diameter: Between 0 and $\sqrt{d}$
- Volume: Between 0 and

$$(\sqrt{\frac{d}{2}})^{d}(\pi^{\frac{d}{2}}/T(\frac{d}{2}+1)) \gg 1$$

Algorithm

Search_ss-tree (node nd, query tokens Ks, node_list ndl)
1: \\ nd: the node to be searched
2: \\ Ks: Two tokens array associated with queries predicate vectors. Ks [0] is the token for POI matching Detection, whereas Ks[1] detects intersection of circular areas.
3: \\ ndl: the list to store matched leaf nodes
4: Read all nodes
5: C ← nd.encrypted_attribute_vector
6: if nd is a leaf node then
7: if Check (Ks [0], C) == 1 then
8: \\ nd's record matches the q's area
9: Add nd to node_list ndl.
10: end if
11: else
12: if Check (Ks [0], C) == 1 then
13: \\ nd's area intersects with the q's area
14: for each child node cld_i of nd do
15: Search_ ˆ ss-tree (cld_i, Ks, ndl)
16: end for
17: end if
18: end if

IV. RESULTS

We provide security to the healthcare system for that purpose implement HMAC-SHA256 algorithm. In HMAC-SHA256 algorithm encrypt all healthcare related data, i.e. any unauthorized person can't be access any data from healthcare system.

For provide security to healthcare system's digital images like X-Rays, MRI, etc. We implement Selective Image Encryption algorithm. This algorithm generate encrypted image it cannot possible view by human without decryption.

Below figure shows the original sample image from healthcare system database

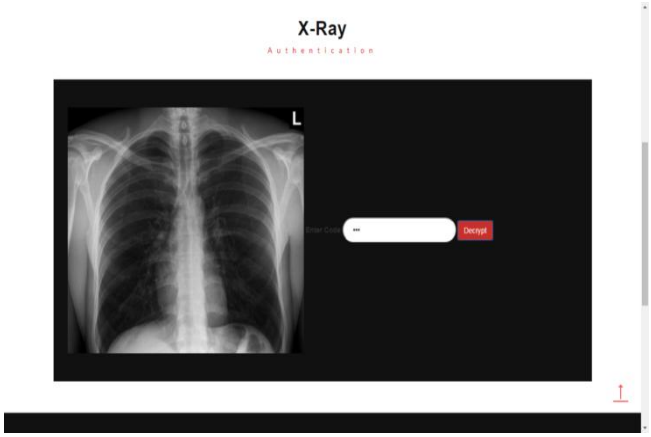


Fig 7: Sample original digital image in healthcare system

Following figure shows Encrypted image from healthcare system this image is secure using image encryption algorithms. Without doctors permission this image cannot be decrypted.

We quantified the performances of the SS-tree under a similar condition with tile test in figure 9. The most extreme number of sections in a node and in a leaf is appeared in figure 6.

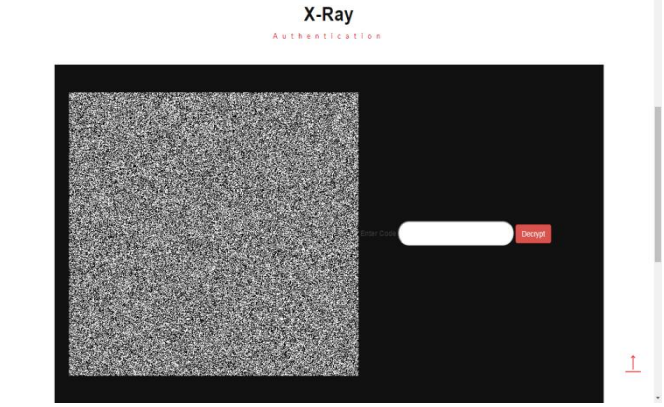
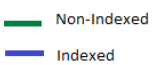


Fig 8: Encrypted image in healthcare system

For tile SS-trees, the exact volume and precise diameter are not estimated, in light of the fact that it is very hard to process them for the intersection of a sphere. Rather, we measured the volumes and the diameters of their bounding spheres. These estimations demonstrate the upper limit of real volume arid the real diameter. Since a region. Which is intersection of its bounding sphere, has a smaller volume help a shorter diameter than its bounding sphere and its bounding sphere.

1. Space Complexity



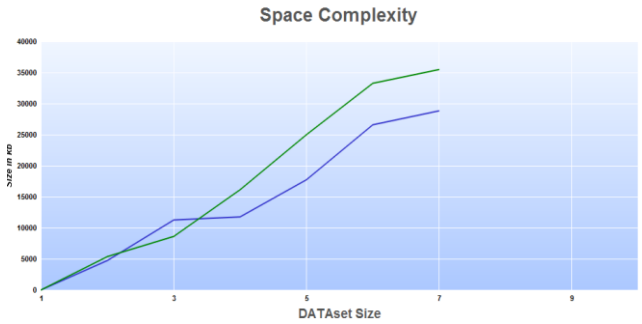


Fig 9: Space Complexity

Above graph shows the result for memory or Space required for retrieve data from huge amount of data .This result get in two different way first green line means memory required for execution or fetching data in normal mode or without indexing fetching data and secondly blue line shows the result fetching confidential data with indexing mode.

2. Index Efficiency

The following result shows the result of Index efficiency means get output in within minimum time, there are two different lines in which first green line indicates normal method has time consuming techniques where as the below line means blue line is shows time required using indexing techniques.

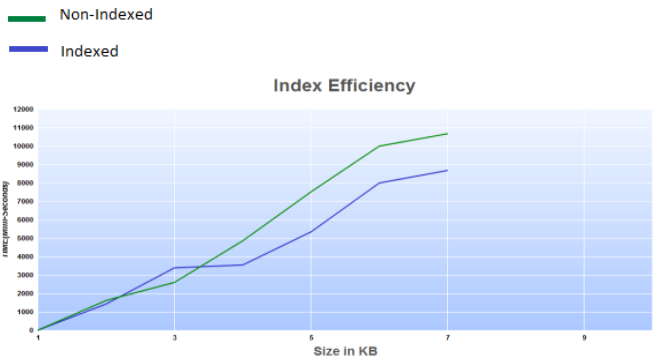


Fig 10: Index Efficiency result

The paper is ready for the template after the text edit has been completed. Duplicate the template file by using the Save As command and use the naming convention recommended for the paper name by your conference. Highlight all the content in the recently produced document and import your prepared text file. Now you're ready to style your paper ; just scroll down the window on the left of the toolbar for MS word formatting.

V. CONCLUSION

We implement the indexing algorithms that were necessary to secure data retrieve in a privacy of big data in healthcare monitoring system. We studied and implement the indexing algorithms that could work properly in our system with limited computation power and small memory unit. We use one kind of indexing in this paper, which we call "similarity indexing,"

and give a solution as another indexing structure called the SS - tree. Included in the definition are vital ideas, such as the task of sampling similarity. We use SS-Tree indexing algorithm for secure data retrieved. The after - effects of our tests suggest that for similarity indexing applications the SS - tree is prevalent. Using SS-Tree Index similarity algorithm we reduce query time for data retrieved.

We also implement HMAC-SHA256 for security purpose to healthcare database authentication, prevent unauthorized access of system. And Selective encryption algorithm for healthcare related images like X-Rays, MRI, Angiography, etc.

REFERENCES

1. R. Jain, "InfoScopes: Multimedia Information Systems," in Multimedia Systems and Techniques (B. Furht, ed.), ch. 7, pp. 217-254, Kluwer Academic Publishers, Norwell, MA, 1996. To be published. Also available as Visual Computing Lab Technical Report VCL-95-107.
2. A. Gupta, T. Wevmouth, and R. Jain. "Semantic queries with pictures: the- VIMSYS model," in Proc. 17th Internationals Conference on Very Large Data Bases, pp. 69-79, Sept. 1991.
3. V. N. Gudivada and V. V. Raghavan, "Content-based image retrieval systems," IEEE Computer, vol. 28, pp. 18-22, Sept. 1995.
4. C. Faloutsos, R. Barber, M. Flickner, J. Hafner, et al., "Efficient and effective querying by image content," Journal of Intelligent Information Systems: Integrating Artificial Intelligence and Database Technologies, vol. 3, pp. 231-62, July 1994.
5. A. Pentland, R. Picard, and S. Sclaroff, "Photobook: Tools for Content-Based Manipulation of Image Databases," in Proceedings of the SPIE: Stomge and Retrieval for Image and Video Databases II, San Jose, CA, vol. 2185, pp. 34- 47, Feb. 1994.
6. C. Faloutsos, M. Ranganathan, and Y. lManolopoulos, "Fast Subsequence Matching in Time-Series Databases," in Proceedings of the ACM SIGMOD Intemational Conference on the Management of Data, pp. 419-429, June 1994.
7. S. F. Altschul, W. Gish, W. Miller, E. Myers, and D. J. Lipman, UBasic local alignment search tool," Joumal of Molecular Biology, vol. 215, pp. 403-410, Oct. 1990.
8. S. Berchtold, D. A. Keim, and H.-P. Kriegel,, "Fast searching for partial similarity in polygon databases." Submitted for publication at SIGMOD '96.
9. C. K. Riesbeck and R. C. Shrank, inside case-based reasoning. Hillsdale, NJ: Lawrence Eribaum Associates, 1989.
10. G. Salton and M. McGill, Introduction to Modem Information Retrieval. McGraw Hill International Company, New York, 1989.
11. C. Faloutsos, "A Survey of Information Retrieval and Filtering Methods," Tech. Rep. CS-TR-3514, Dept. of Computer Science, Univ. of Maryland, Aug. 1995.
12. S. Arya and D. M. Mount, "Algorithms for fast vector quantization," in Proc. of DCC '93: Data Compression Conference (J. A. Storer and M. Cohn, eds.), pp. 381-390, IEEE Press, 1993.
13. M. Turk and A. Pentland, "Eigenfaces fair recognition,"Journal of Cognitive Neuroscience, vol. 3, no. 1, 1990.
14. J. R. Bach, S. Paul, and R. C. Jain, "A visual information management system for the interactive retrieval of faces," IEEE Transactions on Knowledge and Data Engineering, I. Ka.me1 and C. Faloutsos, "On packing R-trees," in Proc. 2nd Intemational Conference on Information and Knowledge Management (CIKM-93), (Arlington, VA), pp. 490-499, Nov. 1993.
15. D. M. Gavrila, "R-tree index optimization," in Advances in GIS Research (T. Waugh and R. Healey, eds.), Taylor and Francis, 1994. Also, CS-TR-3292, University of Maryland, College Park, 1994.
16. J. H. Friedman, J. H. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time, "ACM" actions on Mathematical Software. vol. 3, pp. 209-226, Sept. 1977.
17. A. Henrich, "A distance-scan algorithm for spatial access structures," in 2nd ACM workshop on Advances in Graphic Information Systems, (Gaithersburg, Maryland), pp. 136-143, Dec. 1994.
18. H. Samet, The Design and Analysis of Spatial Data Structures Addison-Wesley, 1989.
19. J. Nievergelt, H. Hinterberger, and K. C. Sevcik, "The grid file: an adaptable, symmetric multikey file structure," Proceedings of the ACM TODS, vol. 9, pp. 38-71, Mar. 1984.
20. A. Guttman, "R-trees: a dynamic index striucture for spatial searching," in Proceedings of the ACM SlGMOD International Conference on the Management of Data, pp. 47- 57, June 1984.
21. N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, "The R*-tree: an efficient and robust access method for points and rectangles," in Proceedings of the ACM SIGMOD International Conference on the Management of Data, pp. 322-331, May 1990.
22. J. Haifner, H. Sawhney, W. Equitz, M. Flickner, et al., "Efficient color histogram indexing for quadratic form distance functions," IEEE Z'bansactions on Pattem Analysis and Machine Intelligence, vol. 17, pp. 729-736, July 1995.
23. K.-I. Lin, H. Jagadish, and C. Faloutsos, "The TV-tree - an index structure for high-dimensional data," VLDB Journal, vol. 3, pp. 517-542, Oct. 1994.
24. Panduranga H T, Naveen Kumar S K, "Selective image encryption for Medical and Satellite Images", International Journal of Engineering and Technology (IJET), Vol 5 No 1 Feb-Mar 2013 pp.115-121.
25. G. Salton and A. Wong, "Generation and search of clustered files," ACM Zhnsactions on Database Systems, vol. 3, pp. 331-346, Dec. 1978.
26. H. Zhang and D. Zhong, "A scheme for visual feature based image indexing," in Proceedings of the SPIIS: Stomge and Retrieval for Image and Video Databases lII, San Jose, CA, vol. 2420, pp. 36-46, Feb. 1995.
27. N. Roussopoulos, S. Kelley, and F. Vincent, "Nearest neighbor queries," in Proceedings of the ACM SIGMOD International Conference on the Management of Data, (San Jose, CA), pp. 71-79, June 1995.
28. M. Ester, H.-P. Kriegel, and X. Xu, "A Databbase Interface For Clustering in Large Spatial Databases," in Proceedings of the 1st Intemational Conference on Knowledge Discovery and Data Mining (KDD-95), Aug. 1995. vol. 5, pp. 619-628, Aug. 1993.
29. N a Cheng , Yan Wang, Xuehong Zhao, Ning Li," The Digital Fingerprint of XML Electronic Medical Records based on HMAC-SHA2S6 Algorithm", IEEE 3rd International Conference on Communication Software and Networks,2011,pp. 338-340.
30. Dilli Ravilla, Chandra Shekar Reddy Putta," Implementation of HMAC-SHA256 Algorithm for Hybrid Routing Protocols in MANETs", International Conference on Electronic Design, Computer Networks & Automated Verification (EDCAV), 2015, pp.154-159.
31. Ahmed Mahmood, Charlie Obimbo, Tarfa Hamed, Robert Dony, "Improving the Security of the Medical Images", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol 4 NO. 9 ,2013, pp.137-146.