International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 12 | Dec -2016 www.irjet.net p-ISSN: 2395-0072

Safeguarding and Protecting contents related to Client's Location Server Data

Shweta Amrutkar[1], M. M. Naoghare[2]

[1]PG Student, Dept. of Computer Engineering, SVIT, Nashik, India
[2]Assistant Professor, Dept. of Computer Engineering, SVIT, Nashik, India

---

Abstract - *When a client sends request to a server requesting for data, the computation scenario arises several problems related to communication between client and server, especially related to security of the client or users. The privacy of the client related to client's location or data may not be protected and this may lead to misusing one's personal information.*

In this paper, a solution to one of the location based query problems is proposed, which is used for securing the user's location that can be obtained from Global Positioning System devices which helps to find the user's exact location. We implement the system by providing security to the position data which will be communicated in encrypted format. A symmetric key encryption will be used for encrypting the user's data and then the data can be decrypted using a key once it reaches the location server. The system will be used for to-and-fro communication during an exchange of data between server and client. Also, the concept of optimal searching based on the user's behavior and the past search strategies will be added. The concept of ranking will also be added as additional feature

Key Words: *Location based services, Data encryption algorithms, Optimal Searching*

1. INTRODUCTION

High-speed wireless networks are now available with most citizens of the world and along with the popularity of portable electronic smart devices using web-based activities have fuelled the development of mobile computing and applications. Compared to traditional computing paradigms, mobile computing enables clients to have unrestricted mobility while maintaining network connection. The ability of users to move and identify their own locations opens up a new kind of information services, called location-dependent information services (LDISs), which produce the answer to a query according to the location of the client issuing the query.

A location based service (LBS) includes information, entertainment and utility service which is accessible by mobile devices like mobile phones, Global positioning system (GPS) devices, pocket personal computers (PCs), and also that operates through mobile network. Based on the geographical position of their mobile device, LBS can offer many services to the user. The services based on a Point of Interest (POI) database are provided by LBS. By retrieving the Points Of Interest (POIs) from the database server, the user gets answers to various location based queries, which include but are not limited to - discovering the nearest ATM machine, gas station, hospital, or police station. In recent years there has been a huge increase in the number of queries for information about POI to the location server from mobile devices. Sometimes users may feel reluctant about disclosing their locations information to the LBS, because it may be possible for a location server to know who is making a certain query by linking these locations with the residential phone book database, as users are likely to perform many queries from home. Location Servers (LS), spends their resources for compiling information about various interesting POIs. Hence, it is expected that the LS should not disclose any information without fees. Therefore, the LBS has to ensure that any unauthorized user is not accessing the LS data. During the process of transmission the users should not be allowed to discover any unpaid information. It is thus important that solutions should be framed to address the privacy of the users issuing queries, sand also prevent users from accessing the content to which they do not have any authorization.

The NIST Computer Security Handbook [NIST95] defines the term computer security as, "The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability and confidentiality of information system resources (includes hardware, software, firmware, information/data, and telecommunications)." Security is the mechanism by which information and services are protected from unintended or unauthorized access, change or destruction. Security in networking is based on Cryptography, the science and art of transforming messages to make them secure and immune to attack [1].

2. RELATED WORK

If you go through the research records, the first applicable solution to the issue of client server address hiding was formulated by Beresford, in which the privacy of the client server is maintained by frequently changing the user's address name or by using a pseudonym within some dedicated mix-zone. Due to the characteristic of the data being handled and exchanged between the user and the server, the instantaneous changing of the user's name

provides little protection for the user's privacy. He also investigated the required number of users required to satisfy the delinking property when there are repeated queries over an interval. This requires careful control on the number of users that are contained within the considered mix-zone, which is difficult to achieve in practice. [2]

Another suggested approach to address hiding is the concept of k-anonymity, which was basically introduced as an efficient technique for hiding and preserving privacy of user when releasing very sensitive records. This is achieved by using two algorithms of computation, that is generalisation algorithm and suppression algorithms, used normally to ensure that a record could not be tracked, identified and distinguished from (k – 1) other available records. The system for LBS uses a recognised and trusted anonymiser to provide anonymity for the actual location data, such that the location data of a user cannot be easily distinguished from (k -1) other users. [3]

An improved trusted anonymiser approach has also been proposed, which allows the users to set their level of privacy based on the value of k. This means that, given the overhead of the anonymiser, a small value of k will assure that there is increase in the process efficiency and speed. Conversely, a large value of k will assure that there is rapid improvement in the privacy, but on the cost of speed and efficiency, if the users feels that their position data is viable to be misused and could be handled maliciously. There have been efforts to make the process natural by adding the concept of feeling-based privacy, where instead of specifying k value directly, it is proposed that the user specifies a cloaking region that they feel will protect their privacy, and the system sets the number of cells for the region based on the popularity of the area. The popularity is computed by using historical footprint database that the server collected. [4], [5]

Methods have also been proposed to confuse and distort the location data, which include path and position confusion. Path confusion was presented by Hoh and Gruteser. The idea was to add uncertainty to the location data of the users, generally at the points the paths of the users cross, making it hard to trace users based on raw location data that was k-anonymised. Position confusion has also been proposed as an approach to provide privacy. The idea is for the trusted anonymiser to group the users according to a cloaking region (CR), thus making it harder for the LS to identify an individual. [6][7] Another method for hiding the address of client, is to use 'dummy' locations during data communication. The basic idea is to confuse the location of the user by sending many random false locations to the server, such that the server cannot distinguish the actual location from the fake locations. This incurs both processing and communication overhead for the user device, which reduces efficiency and reduce speed of operation.[8] One of the recent system devised to achieve privacy of Client address, is the concept of Private information retrieval (PIR) location scheme. The basic idea is to employ PIR to enable the user to query the location database without compromising the privacy of the query. Generally speaking, PIR schemes allow a user to retrieve data (bit or block) from a database, without disclosing the index of the data to be retrieved to the database server. [9]

Focusing on the present scenario, while securing information these days, most organizations focus on securing only the infrastructure holding the data to be secured. They put them behind firewalls and encrypted file stores. Once the infrastructure has been bypassed however e.g. the by a firewall breach or something as simple as an authorized user copying the data off the secured location, the information is there for anyone to view. [10] Using information centric security controls such as DRM, DLP etc. helps secure the information itself regardless of a network or user breach. In most organizations there is a heavy reliance on practitioner intuition, experience, industry lore and best practices, though this may add value to the organization, they do not help management make consistent informed decisions about security.[11] Therefore organizations spend too little or too much time and money in an attempt to mitigate information security risks. Frameworks such as the Fair risk management framework aim to provide a platform for understanding, analyzing and measuring security risks. They propose to analyze these risks in a manner that is easily understood to employees at every level.

3. METHODOLOGY

Whenever a client sends request to a server requesting for data, then there are several problems related to data privacy, client location privacy and others. The privacy of the client related to client's location or data may not be protected and this may lead to misusing one's personal information. Thus, to avoid this, there is a need to protect the details of the client who is sending the request.

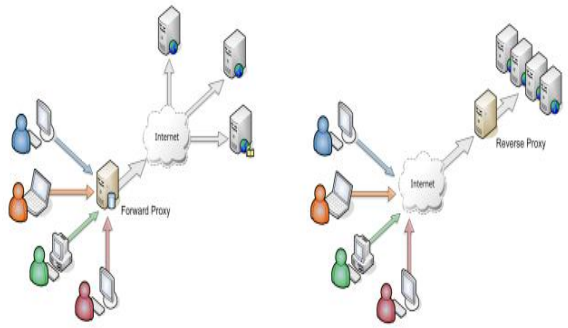

Fig 1: Positioning of Systems in the proposed model

A solution to one of the location based query problems is proposed, which is used for securing the user's location that can be obtained from Global Positioning System devices which helps to find the user's exact location. We implement the system by providing security to the position data which will be communicated in encrypted format. A symmetric key encryption will be used for encrypting the user's data and then the data can be decrypted using a key once it reaches the location server. The system will be used for to-and-fro communication during an exchange of data between server and client. Also, the concept of optimal searching based on the user's behavior and the past search strategies will be added. The concept of ranking will also be added as additional feature. The paper describes a system that consists of a standalone software which will provide a good interface and will be user friendly and it will analyze the outgoing traffic when we are going to use proxy server and will recognize critical information or your personal information if it is accessed by the proxy server or not further on after the recognition it will protect our critical data to be accessed from the illegal sources. The most important thing is to have a proper internet connection because this proposed system completely works on sending and receiving data from client and server. The general system flow is shown in fig.1.

A novel and probabilistic approach is applied to solve the problem of client privacy. The privacy protection can be maintained by allowing a proxy server to act as an intermediate between the clients and the servers. Here all the requested data will be send in encrypted form to the proxy server. Hence, by doing this the details of the client as well as their location can be easily protected. Also, for getting the accurate results, Advanced Encryption Standard (AES) algorithm is used to get the data in the encrypted format so that any other person cannot view the details, henceforth efficiency will be provided. Symmetric key is used meaning the same key is used for both encrypting and decrypting the data. Use of optimal searching based on past queries and user's behavior of a particular query is done. This algorithm has proved more efficient than other encrypting methods.

Network encryption (sometimes called network layer, or network level encryption) is a network security process that applies crypto services at the network transfer layer - above the data link level, but below the application level. The network transfer layers are layers 3 and 4 of the Open Systems Interconnection (OSI) reference model, the layers responsible for connectivity and routing between two end points. Using the existing network services and application software, network encryption is invisible to the end user and operates independently of any other encryption processes used. Data is encrypted only while in transit, existing as plaintext on the originating and receiving hosts. Network encryption is implemented through Internet Protocol Security (IPSec), a set of open Internet Engineering Task Force (IETF) standards that, used in conjunction, create a framework for private communication over IP networks. IPSec works through the network architecture, which means that end users and applications don't need to be altered in any way. Encrypted packets appear to be identical to unencrypted packets and are easily routed through any IP network.

4. DES ENCRYPTION PROCESS

The Data Encryption Standard (DES) is a block Cipher which is designed to encrypt and decrypt blocks of data consisting of 64 bits by using a 64-bit key. Although the input key for DES is 64 bits long, the actual key used by DES is only 56 bits in length. The least significant (right-most) bit in each byte is a parity bit, and should be set so that there are always an odd number of 1s in every byte. These parity bits are ignored, so only the seven most significant bits of each byte are used, resulting in a key length of 56 bits. The algorithm goes through 16 iterations that interlace blocks of plaintext with values obtained from the key. The algorithm transforms 64-bit input in a series of steps into a 64-bit output. The same steps, with the same key are used for decryption. There are many attacks and methods recorded till now those exploit the weaknesses of DES, which made it an insecure block cipher. Despite the growing concerns about its

vulnerability, DES is still widely used by financial services and other industries worldwide to protect sensitive on-line applications.

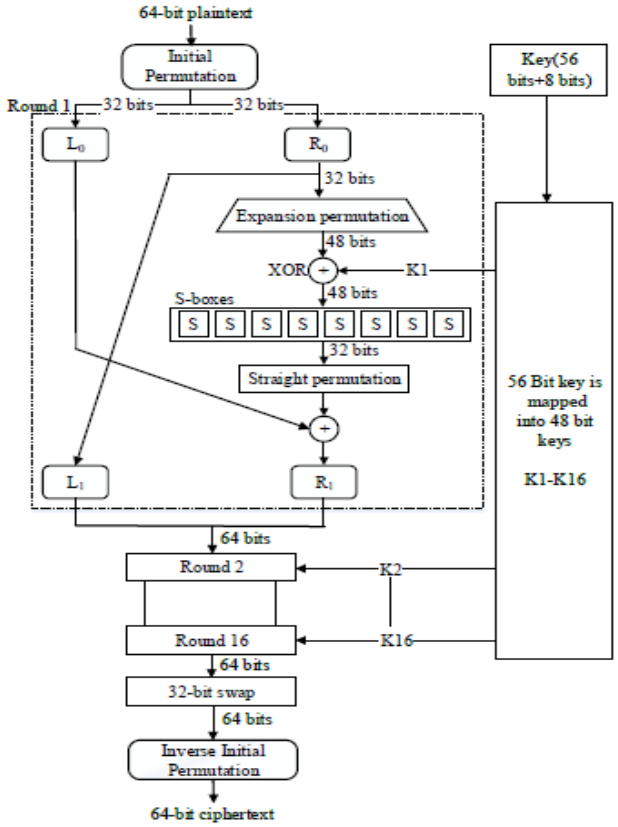

Fig 2: DES Encryption Process

The Feistel (F) function, operates on half a block (32 bits) at a time and consists of four stages:

- Expansion: the 32-bit half-block is expanded to 48 bits using the expansion permutation, denoted E in the diagram, by duplicating half of the bits. The output consists of eight 6-bit (8 * 6 = 48 bits) pieces, each containing a copy of 4 corresponding input bits, plus a copy of the immediately adjacent bit from each of the input pieces to either side.
- Key mixing: the result is combined with a sub-key using an XOR operation. Sixteen 48-bit sub-keys—one for each round—are derived from the main key using the key schedule.
- Substitution: after mixing in the sub-key, the block is divided into eight 6-bit pieces before processing by the S-boxes, or substitution boxes. Each of the eight S-boxes replaces its six input bits with four output bits according to a non-linear transformation, provided in the form of a lookup table. The S-boxes provide the core of the security of DES—without them, the cipher would be linear, and trivially breakable.
- Permutation: finally, the 32 outputs from the S-boxes are rearranged according to a fixed permutation, the P-box. This is designed so that, after permutation, each S-box's output bits are spread across four different S boxes in the next round. The alternation of substitution from the S-boxes, and permutation of bits from the P-box and E-expansion provides so-called "confusion and diffusion" respectively

5. AES ENCRYPTION PROCESS

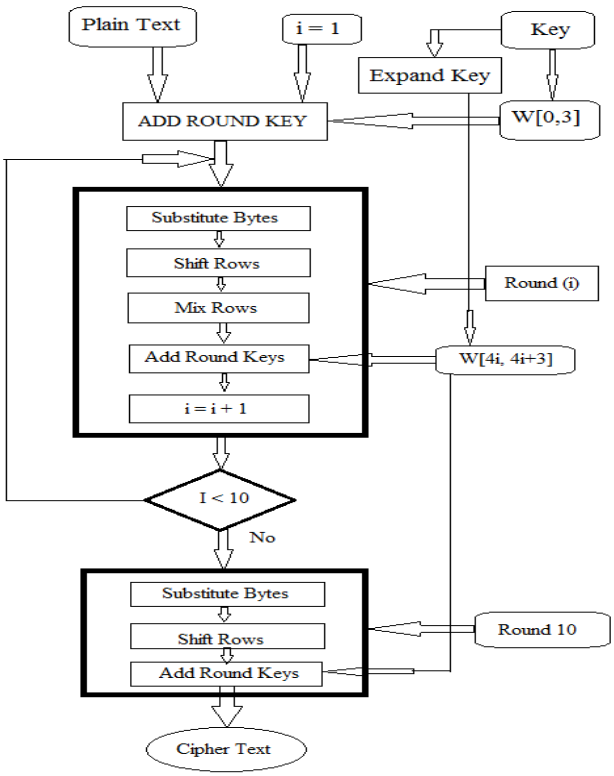

Fig 3- Advanced Encryption Standard (AES) algorithm

Advanced Encryption Standard (AES) is based on a design principle known as a substitution-permutation network, combination of both substitution and permutation, and is fast in both software and hardware. AES operates on a 4 × 4 column-major order matrix of bytes, termed the *state*. AES algorithm can support any combination of data (128 bits) and key length of 128, 192, and 256 bits. The algorithm is referred to as AES-128, AES-192, or AES-256, depending on the key length. During encryption-decryption process, AES system goes through 10 rounds for I28-bit keys, 12 rounds for I92-bit keys, and 14 rounds for 256-bit keys in order to deliver final cipher-text or to retrieve the original plain-text. AES allows a 128 bit data length that can be divided into four basic operational blocks. These blocks are treated as array of bytes and organized as a matrix of the order of 4×4 that is called the state. For both encryption and decryption, the cipher begins with an AddRoundKey stage. However, before reaching the final round, this output goes through nine main rounds, during each of those rounds four transformations are performed. Most AES calculations are done in a special finite field. The key size used for an AES cipher specifies the

number of repetitions of transformation rounds that convert the input, called the plaintext, into the final output, called the *ciphertext*. Each round consists of several processing steps, each containing four similar but different stages, including one that depends on the encryption key itself. A set of reverse rounds are applied to transform ciphertext back into the original plaintext using the same encryption key. The encryption process uses a set of specially derived keys called round keys. These are applied, along with other operations, on an array of data that holds exactly one block of data that is to be encrypted. This array we call the state array. You take the following AES steps of encryption for a 128-bit block:

- Derive the set of round keys from the cipher key.
- Initialize the state array with the block data (plaintext).
- Add the initial round key to the starting state array.
- Perform nine rounds of state manipulation.
- Perform the tenth and final round of state manipulation.
- Copy the final state array out as the encrypted data (ciphertext).

The block to be encrypted is just a sequence of 128 bits. AES works with byte quantities so we first convert the 128 bits into 16 bytes. We say "convert," but, in reality, it is almost certainly stored this way already. Operations in AES is performed on a two-dimensional byte array of four rows and four columns. At the start of the encryption, the 16 bytes of data, numbered D0 - D15, are loaded into the array. Each round of the encryption process requires a series of steps to alter the state array. These steps involve four types of operations called Sub-Bytes, Shift Rows, Mix Columns and XOR Round Key. The cipher key used for encryption is 128 bits long. Where this key comes from is not important here; on key hierarchy and how the temporal encryption keys are produced. The cipher key is already the result of many hashing and cryptographic transformations and, by the time it arrives at the AES block encryption, it is far removed from the secret master key held by the authentication server. Now, finally, it is used to generate a set of eleven 128-bit round keys that will be combined with the data during encryption. Although there are ten rounds, eleven keys are needed because one extra key is added to the initial state array before the rounds start.

When the client sends query to the server, this query and the client's information must be encrypted so that any other person cannot view their details. When client sends query, after performing encryption, the data must go to the proxy server to carry out the intermediate processing. Also when the client sends query, that result must be present in the server so that the server can send reply to the query that has been send by the client. Also there is a possibility of sending one or more queries at the same time by different client and this must also be handled properly by the server.

6. IMPLEMENTATION OF ENCRYPTION

In the proposed system, the exchange of information is done using AES encryption. A software is developed to enable the operation. It works as a mediator for information exchange, but it's enabled with the process of information address hiding. The figures below shows the software screen showing the aspects like search page, server status, & proxy-server status. The contents shown in server are requests received from clients

Fig 4- Home Screen for Software

Fig 5- Software screen showing active server

Fig 6- Software screen for Proxy-server

The following steps are involved in the functioning of the procedure followed for hiding the client address and perform the required data transfer:

1. The start step is marked when client sends a query.

2. This query is first encrypted using the AES algorithm where a key is used to perform the encryption..
3. This encrypted information is stored in the proxy server.
4. The proxy server then sends the location and encrypted query information to the server.
5. This encrypted data is then decrypted again on the server side.
6. Based on the information, searching algorithm is carried out and passed to multiple servers.
7. The server then replies back the required results.
8. The information before communication is encrypted first using the AES algorithm.
9. This encrypted data is send to the proxy server
10. Decryption of information is performed so that the client can receive the information.

7. RESULTS GRAPHS

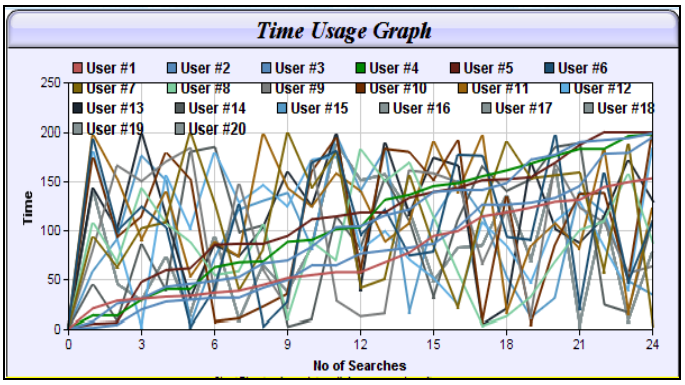

Fig 7: DES Time Graph

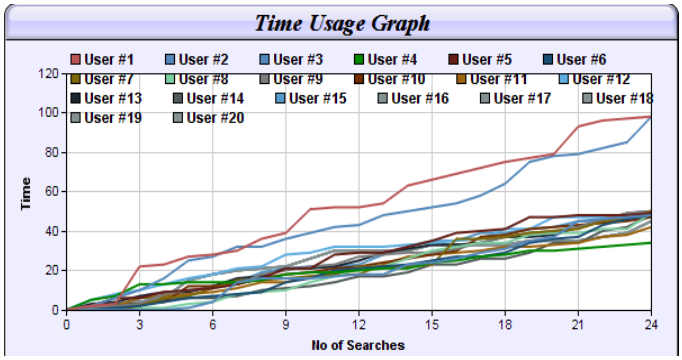

Fig 8: AES Time Graph

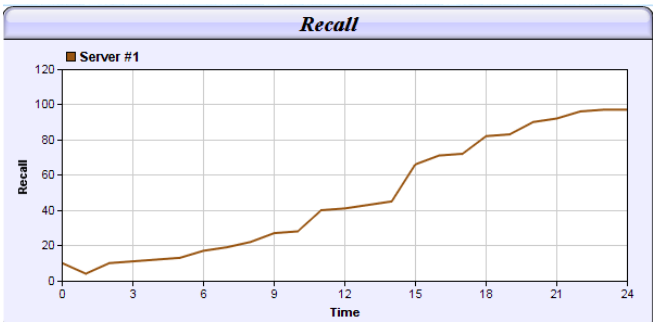

Fig 9: DES Recall Graph

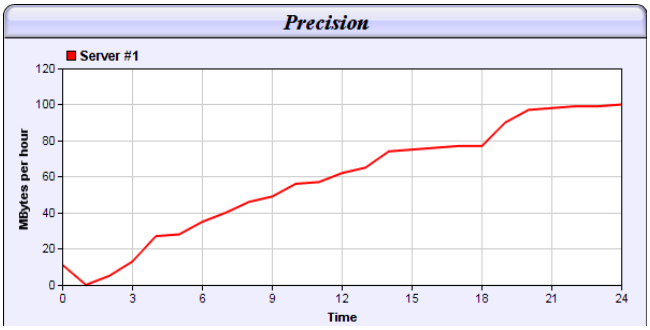

Fig 10: AES Recall Graph

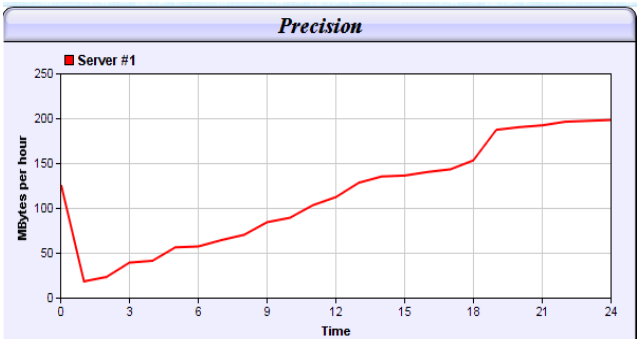

Fig 11: DES Precision Graph

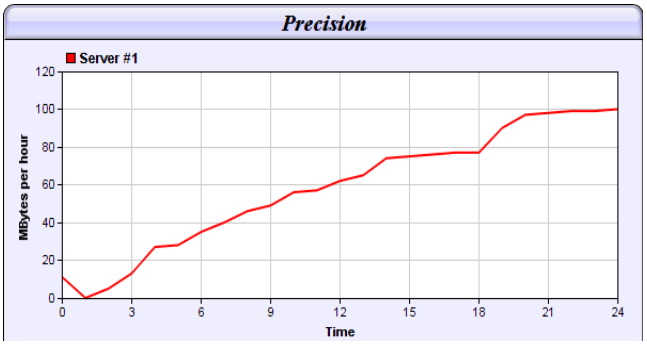

Fig 12: AES Precision Graph

8. CONCLUSIONS

The results reported in this paper conclude that the security systems assigned on cloud system are more efficient than working them on single processor system. For both local and cloud environment, DES is the more time consuming than AES. But it is seen that highest Speed-Up Ratio is obtained in AES for low input file sizes and the Speed-Up Ratio falls sharply as the input file size is increased. DES algorithm consumes least encryption time and AES algorithm has least memory usage while encryption time difference is very minor in case of AES and DES algorithm. In AES, it can be seen that higher key size leads to clear change in the battery and time consumption. Calculation of time for encryption and decryption in different video file format shows that AES algorithm is executed in lesser processing time and more throughput level as compared to DES.

REFERENCES

[1] Behrouz A Forouzan, "Data Communications and Networking", McGraw-Hill, 4th Edition.

[2] S.B. Gosavi, Dr.S V.Gumaste, "Location based Queries for Content protecting and Privacy preserving," International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-1, Issue-4, July 2015

[3] L. Sweeney, "k-Anonymity: A model for protecting privacy," Int. J. Uncertain. Fuzziness Knowl. Based Syst., vol. 10, no. 5, pp. 557–570, Oct. 2002.

[4] L. Marconi, R. Pietro, B. Crispo, and M. Conti, "Time warp: How time affects privacy in LBSs," in Proc. ICICS, Barcelona, Spain, 2010, pp. 325–339.

[5] S. Mascetti and C. Bettini, "A comparison of spatial generalization algorithms for lbs privacy preservation," in Proc. Int. Mobile Data Manage., Mannheim, Germany, 2007, pp. 258–262.

[6] B. Hoh and M. Gruteser, "Protecting location privacy through path confusion," in Proc. 1st Int. Conf. SecureComm, 2005, pp. 194–205.

[7] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing location-based identity inference in anonymous spatial queries," IEEE Trans. Knowl. Data Eng., vol. 19, no. 12, pp. 1719–1733, Dec. 2007.

[8] M. Duckham and L. Kulik, "A formal model of obfuscation and negotiation for location privacy," in Proc. 3rd Int. Conf. Pervasive Comput., H. Gellersen, R. Want, and A. Schmidt, Eds., 2005, pp. 243–251, LNCS 3468.

[9] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private queries in location based services: Anonymizers are not necessary," in Proc. ACM SIGMOD, Vancouver, BC, Canada, 2008, pp. 121–132.

[10] Diaa Salama Abdul. Elminaam, Hatem Mohamed Abdul Kader and Mohie Mohamed Hadhoud, "Performance Evaluation of Symmetric Encryption Algorithms", IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.12, pp. 280-286, December 2008.

[11] S. Pavithra and Mrs. E. Ramadevi, "Performance Evaluation of Symmetric Algorithms", Journal of Global Research in Computer Science, Volume 3, No. 8, pp. 43-45, August 2012.

[12] Zilhaz Jalal Chowdhury, Davar Pishva and G. G. D. Nishantha, "AES and Confidentiality from the Inside Out", the 12th International Conference on Advanced Communication Technology (ICACT), pp. 1587-1591, 2010