Enhancing Data Security in Cloud Computing with WebOS Using TSFS Algorithm

Pooja[1], Kanchan Narula[2]

[1]M.Tech, CSE, HCTM Technical Campus,Kaithal, India

[2]Asst Professor, CSE, HCTM Technical Campus Kaithal, India

Abstract: ***WebOS (Web based operating system) is a new form of Operating Systems. You can use your desktop as a virtual desktop on the web, accessible via a browser, with multiple integrated built-in applications that allow the user to easily manage and organize her data from any location. This paper starts with a introduction of WebOS and its benefits. We have identified some parameters for secure WebOS. A technical review is given with research design of TSFS Algorithm to provide security.***

Keywords: AES (Advanced Encryption Standard), TSFS (Transposition-Substitution-Folding-Shifting encryption)

1. Introduction

Web Operating Systems (WebOS) is: “A software platform that interacts with the user through a web browser and does not depend on any particular local operating system”. Web operating systems are also commonly referred to as Web desktops or WEBTOPS. Web desktop or WEBTOPS is a virtual desktop on the web, running in a web browser as software. The WebOS functions much like a traditional operating system, although it doesn’t include drivers for computer hardware. In cloud computing, users work with Web-based, rather than local, storage and software. These applications are accessible via a browser and look and act like desktop programs. With this approach, users can work with their applications from multiple computers. In addition, organizations can more easily control corporate data and reduce malware infections. Now, a growing number of organizations are adding to the cloud concept by releasing commercial and open source Web-based operating systems.

1.1 Security Issues in Cloud Computing

Abuse and Nefarious Use of Cloud Computing: Abuse and nefarious use of cloud computing is the top threat identified by the Cloud Security Alliance. Attackers can infiltrate a public cloud, for example, and find a way to upload malware to thousands of computers and use the power of the cloud infrastructure to attack other machines. Insecure Application Programming Interfaces: As software interfaces or APIs are what customers use to interact with cloud services, those must have extremely secure authentication, access control, encryption and activity monitoring mechanisms - especially when third parties start to build on them.

1) *Malicious Insiders*: The malicious insider threat is one that gains in importance as many providers still don't reveal how they hire people, how they grant them access to assets or how they monitor them. Transparency is, in this case, vital to a secure cloud offering, along with compliance reporting and breach notification.
2) *Data Loss/Leakage*: Be it by deletion without a backup, by loss of the encoding key or by unauthorized access, data is always in danger of being lost or stolen. This is one of the top concerns for businesses, because they not only stand to lose their reputation, but are also obligated by law to keep it safe.
3) *Account, Service & Traffic Hijacking*: Account service and traffic hijacking is another issue that cloud users need to be aware of. These threats range from man-in-the-middle attacks, to phishing and spam campaigns, to denial-of service attacks.

1.2 Secure Database Encryption in Web Applications

All of today’s organizations store their data in huge databases to retrieve, manipulate and share them in an efficient way. Due to the popularity of databases for storing important and critical data, they are becoming subject to an overwhelming range of threats, such as unauthorized access. Such a threat can result in severe financial or privacy problems, as well as other corruptions. To tackle possible threats, numerous security mechanisms have emerged to protect data housed in databases. Among the most successful database security mechanisms is database encryption. This has the potential to secure the data at rest by converting the data into a form that cannot be easily understood by unauthorized persons. Many encryption algorithms have been proposed, such as Transposition-Substitution-Folding-Shifting encryption algorithm (TSFS), Data Encryption Standard (DES), and Advanced Encryption Standard (AES) algorithms. Each algorithm has advantages and disadvantages.

1.3 Lightweight Cryptography

Lightweight cryptography is a relatively new field aimed to develop more efficient cryptographic implementations in response to typical constraints in the hardware used in Internet of Things (IoT). The hardware used in IoT will likely be constrained in computational power, battery, as well as memory. Lightweight cryptography is tailored for such constrained devices, with the goal of balancing the tradeoffs between low resource requirements, performance, and cryptographic strength. Techniques used to meet this challenge include the adaptation of block ciphers, hash functions, and public key cryptography for lightweight cryptography. Cryptographic technologies are advancing new techniques on attack; design and implementation are extensively studied. Lightweight cryptography is a cryptographic algorithm or protocol tailored for implementation in constrained environments including sensors, contactless smart cards, and health-care devices and so on.

1) Need of Information Security

Information is the most critical resource for many Websites. In many cases, the success of an Website depends on the availability of key information and, therefore, on the systems used to store and manage the data supporting that information. Due to the growth of networked data, security attacks have become a dominant problem in practically all information infrastructures. Database security is the system, processes, and procedures that protect a database from unintended activity. Unintended activity can be categorized as authenticated misuse, malicious attacks or inadvertent mistakes made by authorized individuals or processes. In the presence of security threats, database security is becoming one of the most urgent challenges because much damage to data can happen if it suffers from attacks and unauthorized access. With databases in complex, multi-tiered applications, attackers may reach the information inside the database. Damage and misuse of sensitive data that is stored in a database does not only affect a single user; but possibly an entire organization

2) **Database Encryption**

Encryption can provide strong security for data at rest, but developing a database encryption strategy must take many factors into consideration. For example, where should be performed the encryption, in the storage layer, in the database or in the application where the data has been produced? How much data should be encrypted to provide adequate security? What should be the encryption algorithm and mode of operation? Who should have access to the encryption keys? How to minimize the impact of database encryption on performance? Encrypting sensitive data in databases is no longer an option- it's imperative. Risk management for large scale data breaches, regulatory compliance, protecting intellectual property and maintaining a trusted brand are driving organizations in every industry to seek data level encryption and access controls for an increasing amount of database data. However, concerns about database performance degradation, invasiveness, application support, and managing broad and heterogeneous database encryption implementations too often produce hard barriers to adopting this important database security measure.

3) **Proposed Algorithm Planning**

Enhanced TSFS (transposition, substitution, folding and shifting) algorithm uses four techniques of transformations, which are transposition, substitution, folding and shifting. There are many research studies in the database security field. Some of them have efficient implementations. Also, many encryption algorithms have been proposed, some of which have appealing features but still need further development, one such algorithm is the Transposition, Substitution, Folding and Shifting TSFS algorithm, known as the TSFS algorithm. The TSFS algorithm provides a high degree of security, using a number of features. However, it supports only numbers and alphabetic characters that are not enough to protect different types of sensitive data. There are different steps involved in this algorithm describes as below:

a) *Transposition:* Transposition transformation changes the location of the data matrix elements by using diagonal transposition that reads the data matrix in the route of zigzag diagonal starting from the upper left corner after getting the data and pads it with *s if it is less than 16 digits.
b) *Substitution:* The second algorithm is substitution transformation. It replaces one data matrix element with another by applying certain function. If the element represents an alphabetic character, it then will be replaced with another character. If the element represents a number, it will be replaced with a number and if it represents a symbol, it will be replaced with a symbol.
c) *Folding:* The third algorithm is folding transformation. It shuffles one of the data matrix elements with another in the same entered data, like a paper fold. The data matrix is folded horizontally, vertically and diagonally. The horizontal folding is done by exchanging the first row with the last row. The vertical one is done by exchanging the first column with the last column. The diagonal fold is done by exchanging the inner cells, the upper- left cell with the down-right cell and the upper-right cell with the down-left cell.
d) *Shifting:* The last part of the algorithm is the shifting transformation, which provides a simple way to encrypt. We will use these four steps with our purposed algorithm to encrypt data. TSFS use 1 array while we will increase array size to achieve encryption which will also support special characters.

The main objective is to enhance the TSFS algorithm and accordingly to provide a high security to the databases whilst limiting the added time cost for encryption and decryption by encrypting sensitive data only. The ETSFS algorithm can encrypt the data that consists of alphabetic characters from A to Z, all numbers and the following symbols: (*, -, ., /, :, @ and _). The Enhanced TSFS algorithm is a symmetric encryption algorithm, meaning each transformation or process must be invertible and have inverse operation that can cancel its effect. The key also must be used in inverse order.

Disadvantages

- The Existing TSFS algorithm can encrypt the data that consists of alphabetic characters from A to Z, all numbers and the following symbols: (*, -, ., /, :, @ and _) only.
- Size of data id limited to 16 digits only

2. Literature Review

[1] Qing Zhao (2008), "Study on Security of Web-Based Database", Computational Intelligence and Industrial Application, 2008. PACIIA "08,Page (s):902 – 905
Web database is a combined production with database technology and Web technology, it stores and manages a great deal of data, if they are embezzled or juggled, which

maybe bring enormous political and economic losses to the society. So it is imperative to properly establish security for Web database against illegitimate intrusion. The host identity protocol (HIP) is designed by the Internet Engineering Task Force (IETF), it introduces a separation between the host identity and location identity, and is used to authenticate the host identity of an end system and to set up a limited relationship of trust between two hosts on the Internet. One Web security model is established using the host identity protocol. Its architecture is given. The security of the model is also analyzed and discussed in the paper.

[2] Lianzhong Liu ; Jingfen Gai (2008), “A new lightweight database encryption scheme transparent to applications”, Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference, Publication Year: 2008 , Page(s): 135 – 140

Database encryption, as a mechanism for active security enhancement, is a crucial technique to protect data confidentiality. Two important objectives of designing an encrypted database are high security and performance. In this paper, a new paradigm for database encryption is proposed in which database encryption can be provided as a service to applications with seamless access to encrypted database. Using such an encrypted data management model, applications can concentrate on their core businesses and protect data privacy against both malicious outsiders and the untrusted database service users without need to know encryption details. They propose novel database encryption architecture with flexible data granularity and safe key management for high security and performance of database access. Security dictionary is used to keep encryption metadata safe based on the threat model. Then the implementation details are given to show how to transparently store and query encrypted database fields with the proposed scheme.

[3] Kaur, K. ; Dhindsa, K.S. ; Singh, G. (2009), “Numeric To Numeric Encryption of Databases: Using 3Kdec Algorithm” Advance Computing Conference, 2009. IACC 2009. IEEE International, Publication Year: 2009 , Page(s): 1501 – 1505.

The volume of data storage capacity has changed a lot as compared with earlier times. As most computers were standalone and only the users had access to data, security was not a big concern. All this changed when computers became linked in networks, in form of small dedicated networks to large LANs, WANs and the World Wide Web. With the growth of networking the security of data became a big issue. Data passes through various networks, communication protocols, and devices to ultimately reach to the user which has made data security increasingly important.

3. Proposed Methodology

TSFS supports only numbers and alphabetic characters that are not enough to protect different types of sensitive data Important and critical data as Plain text in database is not secure. The protection of data against unauthorized access or corruption due to malicious actions is one of the main problems faced by web administrators**.**

1) Problem Statement

Security of databases has become increasingly crucial in all application areas. Database security has paramount importance in industrial, civilian and government domains. Organizations are storing huge amount of data in database for data mining and other types of analysis. Some of this data is considered sensitive and has to be protected from disclosure. Challenges for security in database are increased due to the enormous popularity of e-business. In recent years, insider attacks gathered more attention than periodic outbreaks of malware. Database systems are usually deployed deep inside the company network and thus insiders has the easiest opportunity to attack and compromise them, and then steal the data. So data must be protected from inside attackers also. Many conventional database security systems are proposed for providing security for database, but still the sensitive data in database are vulnerable to attack because the data are stored in the form of plaintext only.

The main objective is to enhance the TSFS algorithm which will support special characters to provide a high security to the databases at low time cost for encryption and decryption by encrypting sensitive data only.

Proposed Encryption algorithm

```
Algorithm encryption (String data of any length, [16] keys include special characters)
Pre: Input data of any length.
keys is array that 16 values combination of alphabets, digits and any special characters.
Post: encrypted data is data after encrypting in form of Binary. Data[any] inputData;
String outputData;
if (data length Mod 16 > 0)
pad data by adding *'s to make multiple of 16; Loop: For each Length(inputData) Divide 16:
inputData = Data; //Convert Data in ASCII which will support all special characters.
key = useKeys (keys);
inputData = transposition (inputData);
inputData = substitution (inputData, keys(i),); //Subtract Key value from Data
inputData = folding (inputData); inputData = shifting (inputData);
Convert Data into Binary: inputData= DecToBin(inputData); outputData = inputData;
End Loop
return outputData
End encryption
```

Proposed Decryption algorithm

```
Algorithm decryption (String Binary data, [16] keys include special characters)
Pre: Input Binary Data. Post: Ogirinal Data
Data[any] encryptedData; String originalData;
Loop: For each Length(inputData) Divide 128: inputData = Data; //Convert Binary to Decimal
key = useKeys (keys);
inputData = shifting (inputData); inputData = folding (inputData);
inputData = substitution (inputData, keys(i),); //Add Key value from Data
inputData = transposition (inputData);
```

```
Convert Data into Binary: inputData= DecToChar(inputData);
outputData = inputData;
End Loop
return outputData
End decryption
```

Advantages of New Algorithm

- It Support all Special Characters, Numbers and Digits.
- 16 Digit key which also supports Special Characters, Numbers and Digits. It increase complexity of Algorithm.
- Using ASCII Values of Computation.
- TSFS Steps are different from Existing. Which makes it unique.

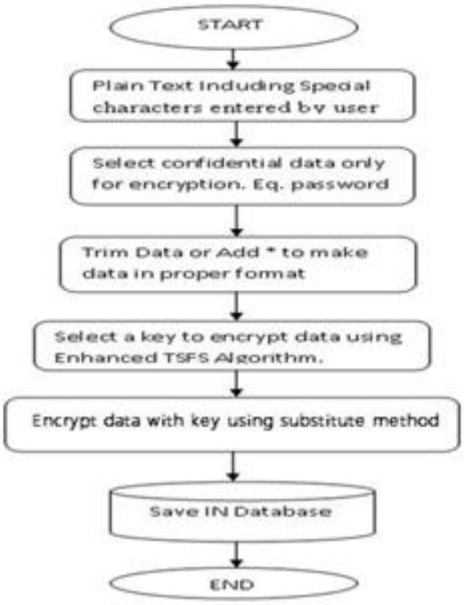
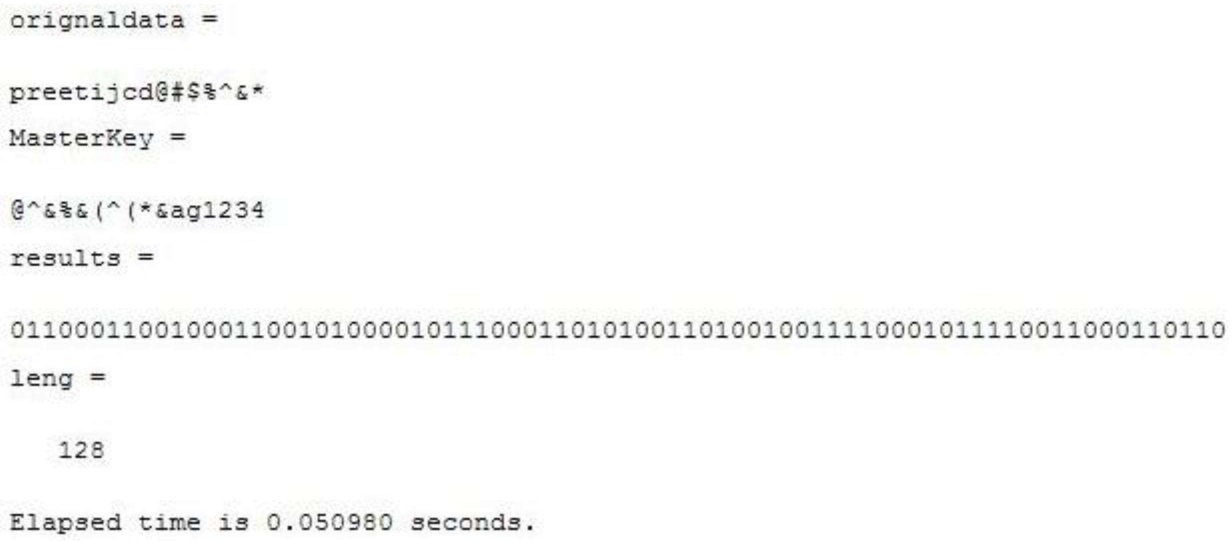
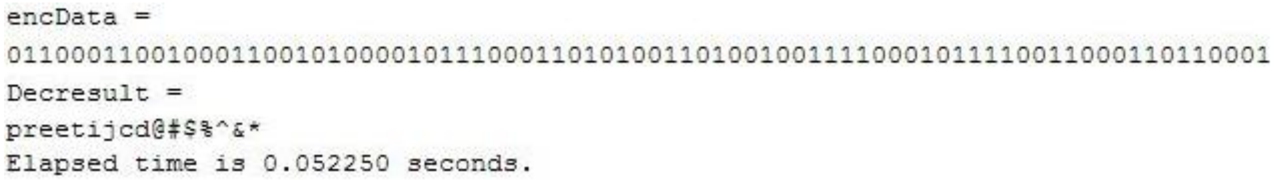

START
Plain Text Including Special characters entered by user
Select confidential data only for encryption. Eq. password
Trim Data or Add * to make data in proper format
Select a key to encrypt data using Enhanced TSFS Algorithm.
Encrypt data with key using substitute method
Save IN Database
END

Figure 2: Flow Chart



```
orignaldata =

preetijcd@#$%^&*
MasterKey =

@^&%&(^(*&ag1234
results =

0110001100100011001010000101110001101010011010010011110001011110011000110110
leng =

   128

Elapsed time is 0.050980 seconds.
```

Figure 3: Password Encryption



```
encData =
0110001100100011001010000101110001101010011010010011110001011110011000110110001
Decresult =
preetijcd@#$%^&*
Elapsed time is 0.052250 seconds.
```

Figure 4: Password Decryption

2) Objective

a) Design an Enhanced TSFS algorithm which will support all special characters also.
b) Proposed Algorithm will be Light weight so that I will take less time to encrypt data.
c) Master key will be used in TSFS. Key can contains special characters, numbers, alphabets.
d) Develop a TSFS Algorithm in MATLAB to evaluate.
e) Analysis of security by Enhanced TSFS Algorithm.

4. Results

Initially the algorithm has been implemented and the crucial attributes has been found in the database. In our approach, the password is the crucial attributes and the information will be saved by algorithm. The web form has been created which includes signup and sign in features. It process multiple of 16. Add x in last in case of data not multiple of 16.

Data Length	*Existing Output Length*	*Proposed Output Length*
16	16	128
20	16	256
35	16	384
40	16	384
55	16	512

Figure 5: Data Length vs Output Length

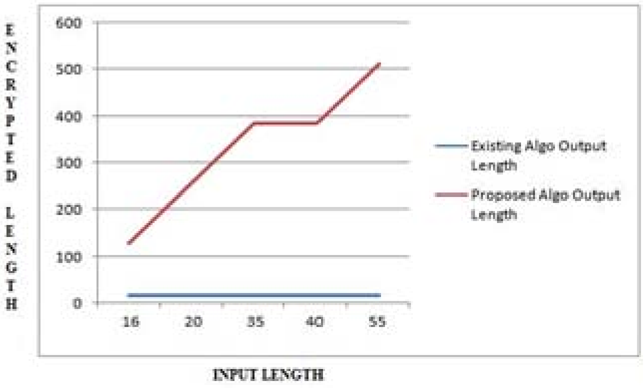


Figure 6: Existing Algorithm Output vs Proposed Algorithm Output

Data Length	Proposed Encryption Time(s)	Existing Encryption Time(s)
16	0.050	0.1
20	0.058	0.2
35	0.059	0.5
40	0.060	0.6
55	0.064	0.8

Figure 7: Existing Encryption Time vs Proposed Encryption Time

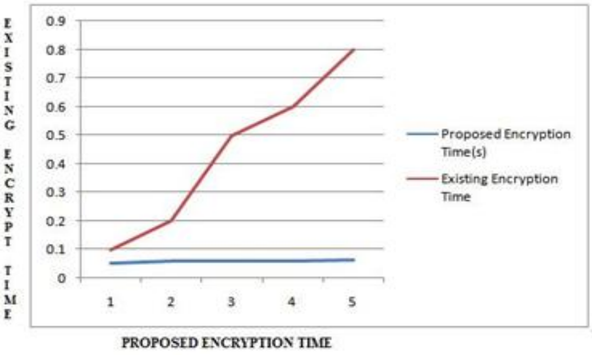


Figure 8: Proposed Encryption Time vs Existing Encryption Time

5. Conclusion

The database of organization is the key part for keep the data available to the client as well as their organization. The plain text can be easily understandable by the intruder so the information needs to be encrypted or convert into another form. Data-storing and exchanging between computers is growing fast across the world. The security of this data has become an important issue for the world. The best solution centered on securing the data is using cryptography, along with other methods. We have been proposed the enhancement of the TSFS algorithm to support the encryption of special characters, correct substitution process by providing more than one modulo factor to differentiate between data types and prevent increasing the data size, as well as correcting the shifting process for the same reasons by providing four 16- arrays. The experimental results have shown that the TSFS algorithm successfully encrypted important symbols, as well as alphanumeric data. The improved performance comes without compromising query processing time or database size. Using well-established encryption algorithms as benchmarks, such as DES and AES, the proposed TSFS algorithm was shown to have consumed the smallest space and encryption time compared to the other algorithms. Our proposed enhanced TSFS algorithm can be improved in terms of security and speed. Encryption of special characters can be done in different methods.

References

[1] Rakesh Agrawal, Jerry Kiernan , Ramakrishnan Srikant , Yirong Xu,(2002), “Hippocratic database”.

[2] Chin-Chen Chang ; Chao-Wen Chan (2003) , “A database record encryption scheme using the RSA public key cryptosystem and its master keys”, Computer Networks and Mobile Computing, Page(s): 345 – 348, IEEE.

[3] Mattsson, U.T (2005), “A practical implementation of transparent encryption and separation of duties in enterprise databases: protection against external and internal attacks of databases”, IEEE, Publication Year: 2005 , Page(s): 559 – 565

[4] Samba Sesay et. al (2005), “A secure database encryption scheme”, Consumer Communications and Networking Conference, 2005. CCNC. 2005 Second IEEE. Page(s): 49– 53.

[5] Gang Chen, “A Database Encryption Scheme for Enhanced Security and Easy Sharing Computer Supported Cooperative Work in Design”, 2006. CSCWD „06. 10th International Conference on Publication Year: 2006, Page(s): 1-6

[6] Yekkala, A.K., “Lightweight Encryption for Images”, Consumer Electronics, 2007. ICCE 2007. Digest of Technical Papers. International Conference on Publication Year: 2007 , Page(s): 1 – 2

[7] Qing Zhao (2008), “Study on Security of Web-Based Database”, Computational Intelligence and Industrial Application, 2008. PACIIA ‟08,Page (s):902 – 905

[8] Lianzhong Liu ; Jingfen Gai (2008), “A new lightweight database encryption scheme transparent to applications”, Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference, Publication Year: 2008 , Page(s): 135 – 140

[9] K. Kaur, K. Dhindsa, G. Singh,(2009), “Numeric to numeric encryption: using 3KDEC algorithm”.

[10] Kaur, K. ; Dhindsa, K.S. ; Singh, G. (2009), “Numeric To Numeric Encryption of Databases: Using 3Kdec Algorithm”, Advance Computing Conference, 2009. IACC 2009. IEEE International, Publication Year: 2009 , Page(s): 1501 – 1505.

[11] Zhu Yangqing ; Yu Hui ; Li Hua ; Zeng Lianming (2009), “Design of a New Web Database Security Model”, Electronic Commerce and Security, 2009. ISECS „09. Second International, Publication Year: 2009, Page(s): 292 – 295

[12] D. Manivannan, R .Sujarani, (2010) Light weight and secure database encryption using TSFS algorithm.

[13] Wu Xing-hui ; Ming Xiu-jun, “Research of the Database Encryption Technique Based on Hybrid Cryptography”, Computational Intelligence and Design (ISCID), 2010 International Symposium, Publication Year 2010 , Page(s): 68 – 71

[14] Jiang Yu-yan ; Pi Xiao-yan; Xing Guo-Zheng , “Database Encryption and Confirmation Mechanism Research”, Multimedia Technology (ICMT), 2010 International Conference, Publication Year: 2010 , Page(s): 1 –4.

[15] Jacob, S. (2010), “Cryptanalysis of a fast encryption scheme for databases”, Information Theory Proceedings (ISIT), 2010 IEEE International, Publication Year: 2010 , Page(s): 2468 – 2472

[16] Zhao Yong-Xia, “The Technology of Database Encryption Multimedia and Information Technology (MMIT)”, 2010 Second International Conference, Publication Year: 2010 , Page(s): 268 – 270.

[17] Yoshino, M. ; Naganuma, K. ; Satoh, H., “Symmetric Searchable Encryption for Database Applications”, Network-Based Information Systems (NBiS), 2011 14th International, Publication Year: 2011 , Page(s): 657 – 662.

[18] Yanhua Pan, “Research on network database encryption technology”, Communication Software and Networks (ICCSN), 2011 IEEE 3rd International, Publication Year: 2011 , Page(s): 690 – 693

[19] Inkyung Jeun ; Hyun-Chul Jung ; Nan Ki Lee ; Dongho Won (2012), “Database Encryption Implementation and Analysis Using Graphics Processing Unit”, Mobile, Ubiquitous, and Intelligent Computing (MUSIC), Page(s): 109 – 113.

[20] Hanan A. Al-Souly, Abeer S. Al-Sheddi, Heba A. Kurdi.,(2013) Lightweight Symmetric Encryption Algorithm for Secure Database.