P: ISSN No. 0976-8602 RNI No.UPENG/2012/426228 VOL.-III, ISSUE-IV, October-2014
E: ISSN No. 2349 - 9443
Asian Resonance

Object Oriented Implementation of DES for Security in E-Learning

Abstract

Now-a-days, E-learning has become the most emergence technique of learning. Though it cannot replace the necessity of teachers and instructors, it can enhance the process of teaching. But, like many other online systems, online learning has also some security problems. Learner and developer are two most important components of e-learning system here authors have made an attempt to send the study material submitted by teachers to the learner can be changed or destroyed by the hackers during transaction between developer and learner.

In this paper, we have wrapped Data Encryption Standard (DES) algorithm in an object oriented model for implementation of confidentiality of the study material utilizing the benefits of object oriented analysis and design, which is the recent trend in software engineering..

Sunil Karforma
Associate Professor,
Deptt. of Computer Science,
The University of Burdwan,
Burdwan, West Bengal

Soumendu Banerjee
Research Scholar,
Deptt. of Computer Science,
The University of Burdwan,
Burdwan, West Bengal

Keywords: E-learning, Use case diagram, Data flow diagram, DES algorithm, Class diagram, Sequence diagram.

Introduction

Since the internet is a public network, it presents some privacy and security issues and therefore in the development of e-learning[7,9,12] systems, security plays an important role. Generally, online transaction or online learning can face significant risks to any institutions like Carnegie Mellon University[14], Cambridge E-learning Institute (CEI)[13] etc. as well as individuals.

E-learning and other types of online educational materials transfer offers many advantages such as improve students' efficiency, knowledge and of course educational qualification or degree. E-learning relies on a networked environment. Network access can be performed through the combination of devices such as personal computers, telephones, interactive television equipment and card devices with embedded computer chips. The connections are completed primarily through telephone lines, cable systems and in some instances wireless technology. These systems, whether informational or transactional, facilitate interaction between the institute and the learner, often with the support of third party service providers. However, not all networks carry the same degree of risk and not all networks are equally vulnerable[6].

It is worth nothing that the internal attacks are the most damaging because if the study materials generated and sent by the institute or developer can be hacked, then it is no more valuable to a student and also make a bad image for the institution. The attackers could misuse the study materials for other reasons inappropriately. That is why, the first thing that an educational institute should do in case of starting online courses, is to review and evaluate the security of internal networks. Again, the messages or the study materials should be delivered to the learner securely; otherwise there will always be a chance of hacking and misuse. Cryptography[11] may be applied to impose against those kinds of attacks.

Also the developer should pay attention on the key distribution of key, which is used for message encryption and decryption process. This key should be distributed secretly between developer and learner before transmission of study material. In this paper authors have made an attempt to object oriented implementation of DES algorithm for secure transmission of study material from developer to learner.

In the section II, we have briefly outlined the DES algorithm. Section III covers the architecture of a secured e-learning system and also the Data Flow Diagram at level1 and level2 for study material encryption and decryption processes. Section IV contains the proposed object oriented models[8] with the help of some important object oriented analysis and design diagrams such as use case; sequence diagram and class diagram use case models, class hierarchy and sequence diagrams. Finally, we have concluded in section V by citing some future scopes of improvement.

Digital Encryption Standard (DES) algorithm for study material encryption

DES[1,3] algorithm may be used to encrypt the study material during transaction between developer and learner is outlined as follows:

Step1

The developer of the e-learning system, i.e. the institution or the developer will send an encrypted study material. For security purpose they will use DES algorithm to make the password protected and keep it secret from others. To do this, developer will take a 64 bit plaintext, of which 8 bits are parity bits.

Step 1.2

In the next step, the parity bits are discarded and reduce it to 56 bits. Now, make a permutation on 56 bits. Split the permuted data into two halves and also calculate the 16 sub keys using following step.

Step 1.3

Perform one or two circular left shifts on both the halves. Permutated the concatenation to get the key, which is 48 bits long and repeat this step until 16 sub keys have not been calculated.

Step 2

Encode each 64 bit block of data.

Step 2.1

Get 64 bit data block. Perform an initial permutation on it. Split the block into two halves, one is left half and the other is right half.

Step 2.2

The left output is simply a copy of the right input. The right output is the bitwise XOR of the left input and the key for this stage.

Step 2.3

The function applied above, consists of four steps. First, a 48 bit number, E is constructed by expanding the 32 bit according to a fixed transposition and duplication rule. Second, E and K_i are XORed together.

This output is then partitioned into eight groups of six bits each, each of which is fed into a different S box. Each of the 64 possible inputs to an S box is mapped onto a 4 bit output. Finally, these 8*4 bits are passed through a P box[1].

Step 3

Finally, a permutation has been done with the left and right block.

Step 4

An inverse transposition of the initial transposition has been taking place, to make the study material encrypted.

The decryption of the study material is quite similar with the encryption process. The developer secretly shares the secret key to the learner before the transmission of the study material. Learner has to follow the above steps in the reverse way to make the study material decrypted.

Architecture of a secured e-learning system

The system has two components: client end and service system. The client sends requests to server and service end gives reply to the client. The architecture of a secured e-learning system has been shown in the Fig 1, in the annexure.

The client and service system are connected through Internet. The client sends request for study material through the web browser using internet. The developer stores the study materials to the database. The application server gets the study material from the back end database and the web server gets the study material from the application server.

The network administrator may control access of resources through Firewall. But Firewall system cannot resolve all kind of risks. Additionally, block cipher encryption algorithm, DES may be used to encrypt data. Similarly, this algorithm may be used as to decrypt data.

Data Flow Diagram DFD

The data flow diagram[11] is a graphical representation of a system, which contains the input data to the system, processes which has been carried out on these data and also the output data generated by the system. Here we two levels of DFD, level1, shown in Fig: 2, in the annexure, consists the study material transmission from developer to learner. Level2 DFD contains two parts, which has been shown in Fig: 3 and Fig: 4, in the annexure. The first part discusses about the study material encryption process and the second part shows the decryption[3] process of the study material. In the level2 DFD the Expansion and XOR function has been executed 16 times, which is difficult to present through the picture. This iteration has been done for 16 times in case of both the encryption and decryption (one iteration is shown here).

Proposed Object Oriented Model of DES algorithm

This section contains the proposed object oriented model with the help of different object oriented analysis and design diagrams such as use case models, sequence diagrams and class diagrams.

Use case Model

In the use case model[11], we select two types of objects like, Developer and Learner. Here developer is encrypting the study materials and sending the key. This key is distributed secretly between developer and learner before the transmission of study materials. Here, we use two use case models.

In the first use case model, shown in Fig: 5, discuss about the encryption of the study materials. Developer is selecting the secret key and this step also includes the initialization of the key and also requires some left shifts of the 28 bits data.

Another use case, in the developer end, is the study material encryption, which includes the make cipher process, which is discussed in the class diagram.

In the next use case model, shown in Fig: 6, the learner gets the secret key from the developer and this step includes two others use cases, which are: initialization of the key and make required number of left shifts of the 28 bits data.

The learner also has the use case of decrypting the study materials and this step includes the make cipher process.

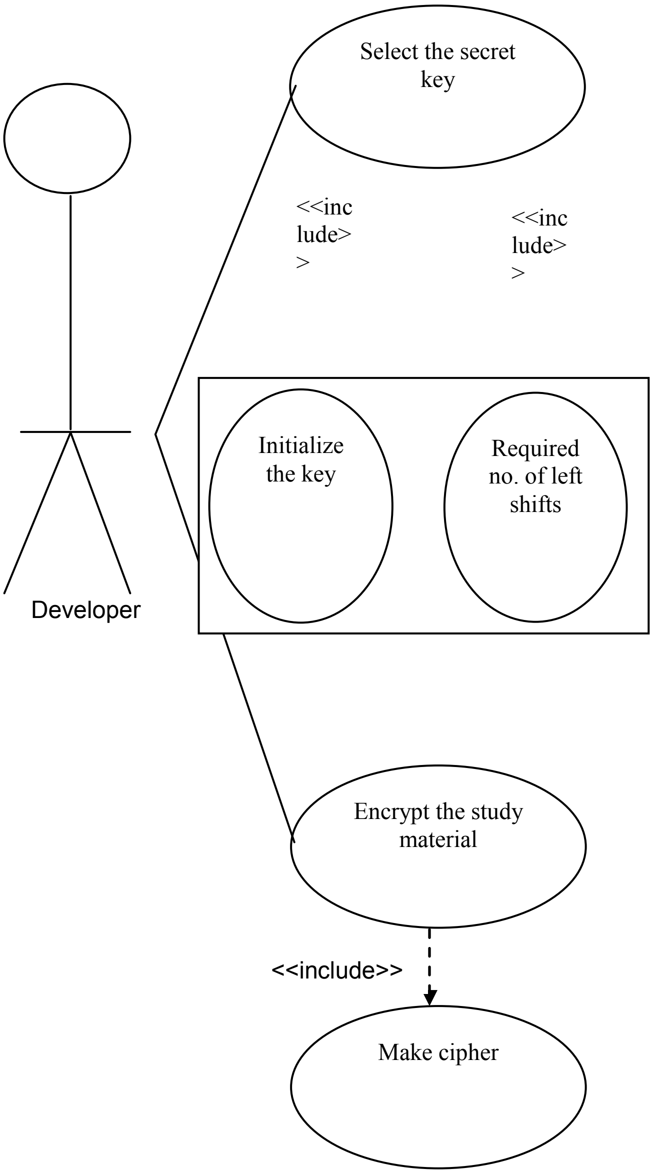

Fig: 5
Use case diagram for study material encryption

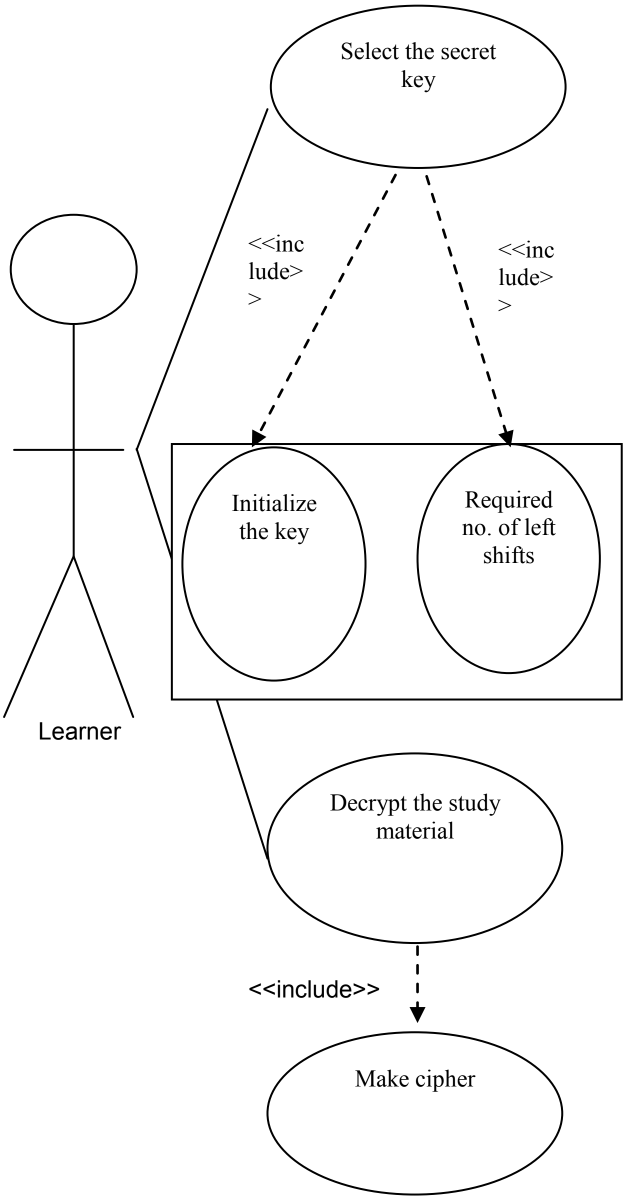

Fig: 6
Use case diagram for study material decryption

Sequence Diagram

A sequence diagram[5,11] shows the interaction among objects as a two dimensional chart. The chart is read from top to bottom. The sequence diagram of study material encryption has shown in the Fig: 8, in the annexure. Here we only sketch the diagram for study material encryption. Here Learner and Developer are two classes. The interaction between learner and developer regarding study material is shown here. In this diagram, like DFD, we cannot show the 16 iterations steps of the expansion and XOR. Here we include one iteration. In this diagram, student will login to the system by giving valid user id and password. If the id or password does not match, then an error message will be displayed. After login, user will request for study material. The developer will encrypt this study material using the steps of DES algorithm and send the encrypted study material and the secret key to the learner to decrypt the material. The decryption process can be drawn in similar manner.

Class hierarchy diagram and its explanation in respect of DES algorithm

The class diagram[2,4,6] of DES algorithm using object-oriented approach is shown in Fig: 7 along with the associated data members (data parts) and member functions (function part).

Both the Developer and Learner classes are publicly inherited from the Des_base class and thereby all the public member functions of Des_base class such as des_init(unsigned char *), left_shift(unsigned char[]), make_cipher(int *r, int cnt, int *fout) will be added and reused in Developer and Learner classes in addition to those special functions which are explicitly defined with in the class Developer and the class Learner. The use of functions dev_enc(unsigned char *input) are defined in the Developer.

Class Developer use a dev_key() function of its own to get its secret key and class Learner use a lrnr_key() function of its own to get its secret key.

The function void dev_enc(unsigned char *input) is used to encrypt the study material sent by the object of Developer to an object of the Learner class.

An object of Learner class will use the function void lrnr_dec(unsigned char *input) to decrypt the received study material using the same secret key which was used by the object of Developer class for encryption of the study material. Developer cannot decrypt the original study material if other secret key is used during decryption.

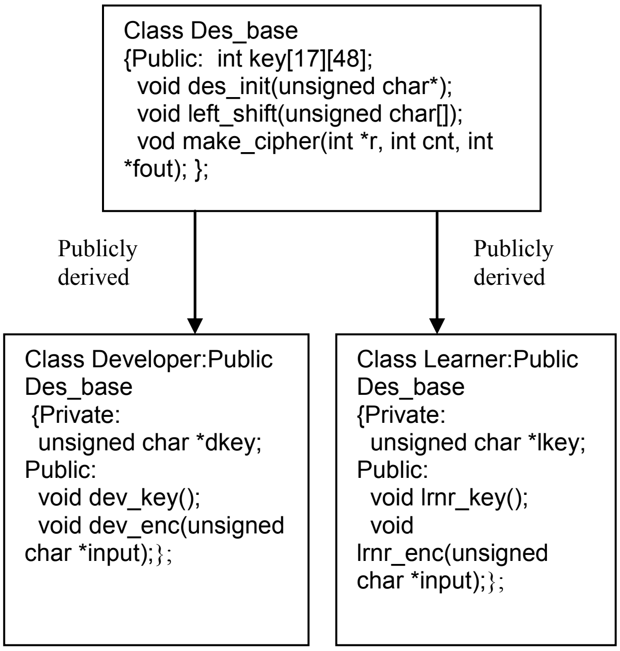

Fig: 7
Class hierarchy diagram of DES algorithm

Class hierarchy diagram and its explanation in respect of DES algorithm

The class diagram[2,4,6] of DES algorithm using object-oriented approach is shown in Fig: 7 along with the associated data members (data parts) and member functions (function part).

Both the Developer and Learner classes are publicly inherited from the Des_base class and thereby all the public member functions of Des_base class such as des_init(unsigned char *), left_shift(unsigned char[]), make_cipher(int *r, int cnt, int *fout) will be added and reused in Developer and Learner classes in addition to those special functions which are explicitly defined with in the class Developer and the class Learner. The use of functions dev_enc(unsigned char *input) are defined in the Developer.

Class Developer use a dev_key() function of its own to get its secret key and class Learner use a lrnr_key() function of its own to get its secret key.

The function void dev_enc(unsigned char *input) is used to encrypt the study material sent by the object of Developer to an object of the Learner class.

An object of Learner class will use the function void lrnr_dec(unsigned char *input) to decrypt the received study material using the same secret key which was used by the object of Developer class for encryption of the study material. Developer cannot decrypt the original study material if other secret key is used during decryption.

Class Des_base
{Public: int key [17] [48];/*the following function is used forDES key processing */void des_init (unsigned char*); void left_shift (unsigned char[]);//left shift of the key

```
 void make_cipher(int *r, int cnt, int *fout);//to process
64 bit data block};Class Developer:Public
Des_base{unsigned char *dkey;Public:
 void dev_key();//getting secret key
 void dev_enc(unsigned char *input);//used for
encryption of the study material};Class Learner :
Public Des_base{ unsigned char *lkey;Public:
 void lrnr_key();//getting secret key from developer
 void lrnr_dec(unsigned char *input);//used to decrypt
the study material};
```

The **driver program** segment is given below:

```
main()
 {Developer d;//d is an object of Developer class
Learner l;//l is an object of Learner class
unsigned char *data;int n;d.dev_key();//to get the
secret keyprintf("\nEnter your message: ");
gets(data);//getting study materiald.dev_enc(data);//d
is invloking dev_enc() to encrypt the material
printf("\nMessage after encryption: ");
puts(data);l.lrnr_key();//to get the secret key
l.lrnr_dec(data);//decryption of the study material
using the same secret keyprintf("\nMessage after
decryption: ")puts(data);}
```

The program implemented using OOP gives satisfactory result for confidential study material transfer between Developer and Learner classes using DES algorithm. The most striking feature of software engineering is the reuse of code, which has been achieved here efficiently.

Conclusion

Secure transmission of study materials must ensure privacy and confidentiality. To achieve this goal, we have applied DES algorithm by wrapping in Object Oriented Model using UML. The proposed system has efficiently implemented inheritance, which is reusing code, as well as, enhances the quality of the proposed system. Here des_init(), make_cipher() and left_shift() member functions are inherited from the base class and for the security purpose, we have defined dkey and lkey as private, utilizing data hiding feature of Object Oriented Programming. The proposed Object Oriented Modeling of DES algorithm, thus implements an efficient way of secure transfer of study materials from developer to learner. No system is absolutely secure. To increase the level of security of the proposed system, triple DES algorithm may be used.

References

1. Andrew, S. Tanenbaum (2005), Computer Networks, Pearson Prentice Hall
2. Balagurusamy, E (2006), Object Oriented Programming with C++, Tata McGraw Hill, New Delhi
3. Behrouz, A Forouzan (2006), Data Communication and Networking, Tata McGraw Hill
4. Dutta Atanu, Karforma Sunil and Anamul Hoda SK (Nov. 2010), Object Oriented Modeling of DES Algorithm for E-Governance Security, Proceedings of ICCS-2010, Dept. of Computer Science, The University of Burdwan, pp 122-126.
5. Karforma Sunil and Ghosh Ambalika (September, 2013), Object Oriented Modeling of SSL for secure information in E-learning, ICCS-2013, Department of Computer Science, The University of Burdwan, West Bengal, India, pp 62-66.
6. Karforma Sunil and Mukhopadhyay Sripati (July, 2005), A Study on the application of Cryptography in E-Commerce, The university of Burdwan, W.B, India.
7. Karforma Sunil and Nikhilesh Barik (January, 2012), Risks and Remedies in E-learning System, International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.1, pp 51-59.
8. Karforma Sunil, Roy A and Banik S (2011), Object Oriented Modeling of RSA Digital Signature in E-Governance Security, International Journal of Computer Engineering and Information Technology (IJCEIT), Summer Edition 2011, Vol 26 Issue No. 01, pp 24-33.
9. Kumar Gupta Sapan and K. Kuriachan Juneesh, Issues and Solutions in E-learning System, International Journal in Multidisciplinary and Academic Research (SSIJMAR), Vol. 2, No. 2
10. Rajib Mall (June, 2006), Fundamentals of Software Engineering, Prentice Hall of India, New Delhi
11. Schneier Bruice (2008), Applied Cryptography, Wiley India Edition
12. Weippl, R.E (2005), Security in E-Learning, Springer
13. http://www.cambridge-elearning.com
14. http://oli.cmu.edu

Annexure

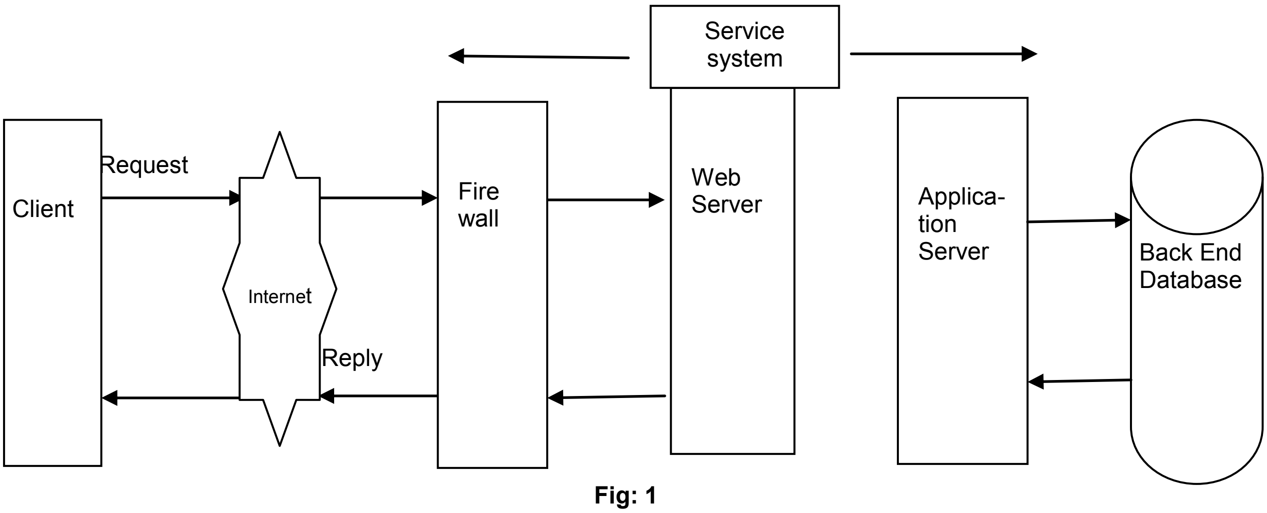

Fig: 1
Architecture of a secured e-learning system

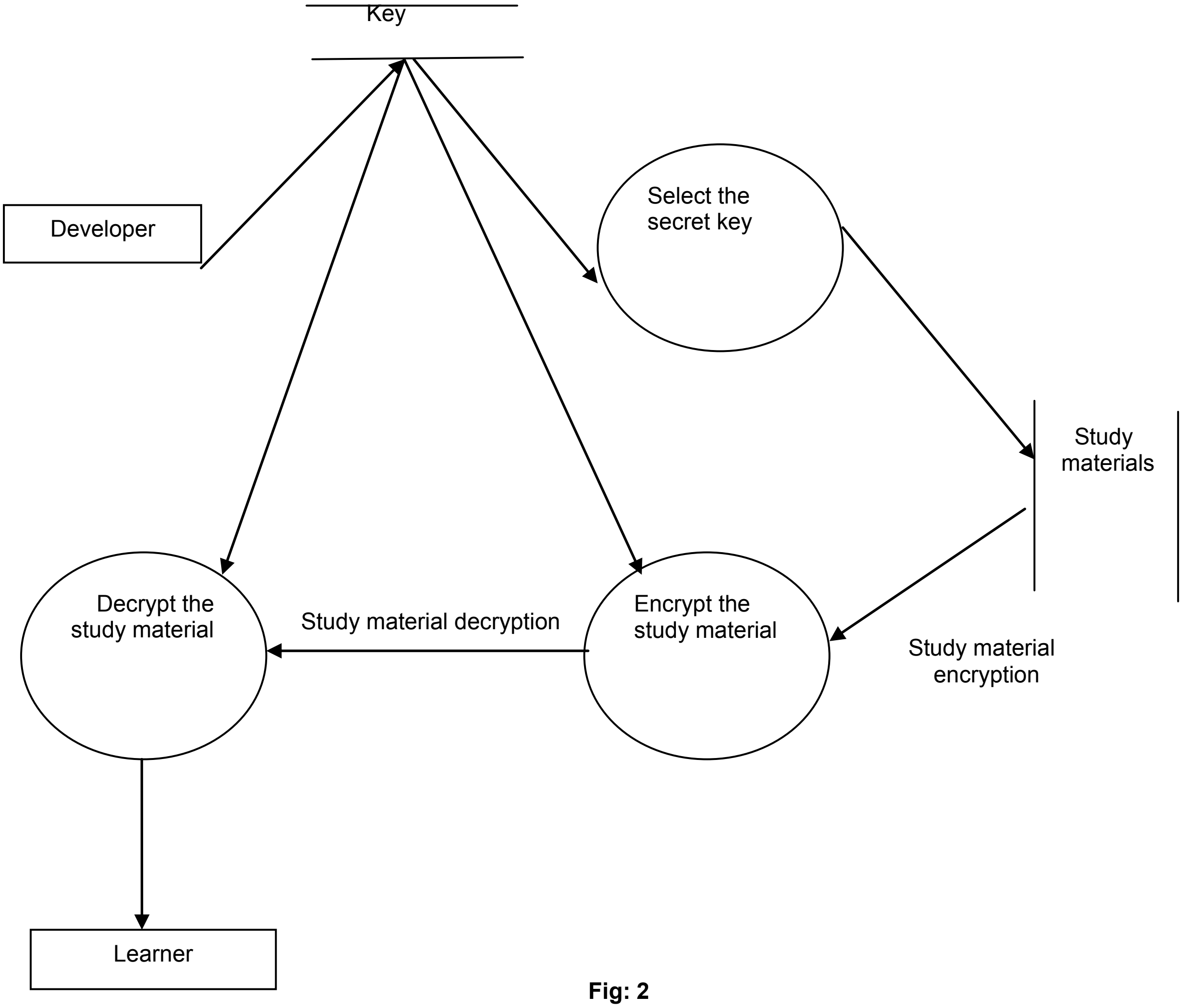
Key
Developer
Select the secret key
Study materials
Decrypt the study material
Study material decryption
Encrypt the study material
Study material encryption
Learner

Fig: 2
Level1 DFD.for study material transmission

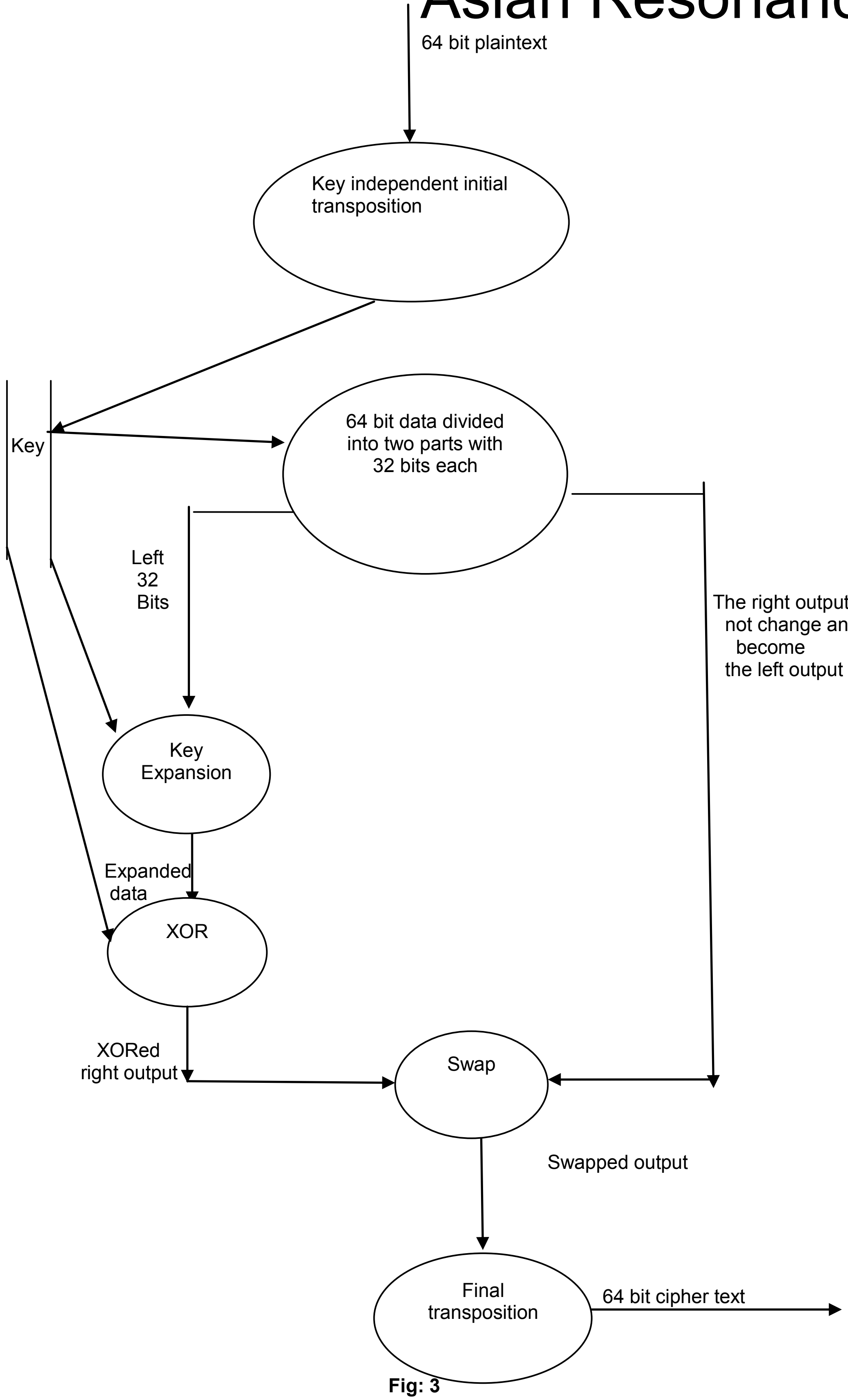
64 bit plaintext
Key independent initial transposition
Key
64 bit data divided into two parts with 32 bits each
Left 32 Bits
The right output does not change and become the left output
Key Expansion
Expanded data
XOR
XORed right output
Swap
Swapped output
Final transposition
64 bit cipher text

Fig: 3
Level2 DFD for study material encryption

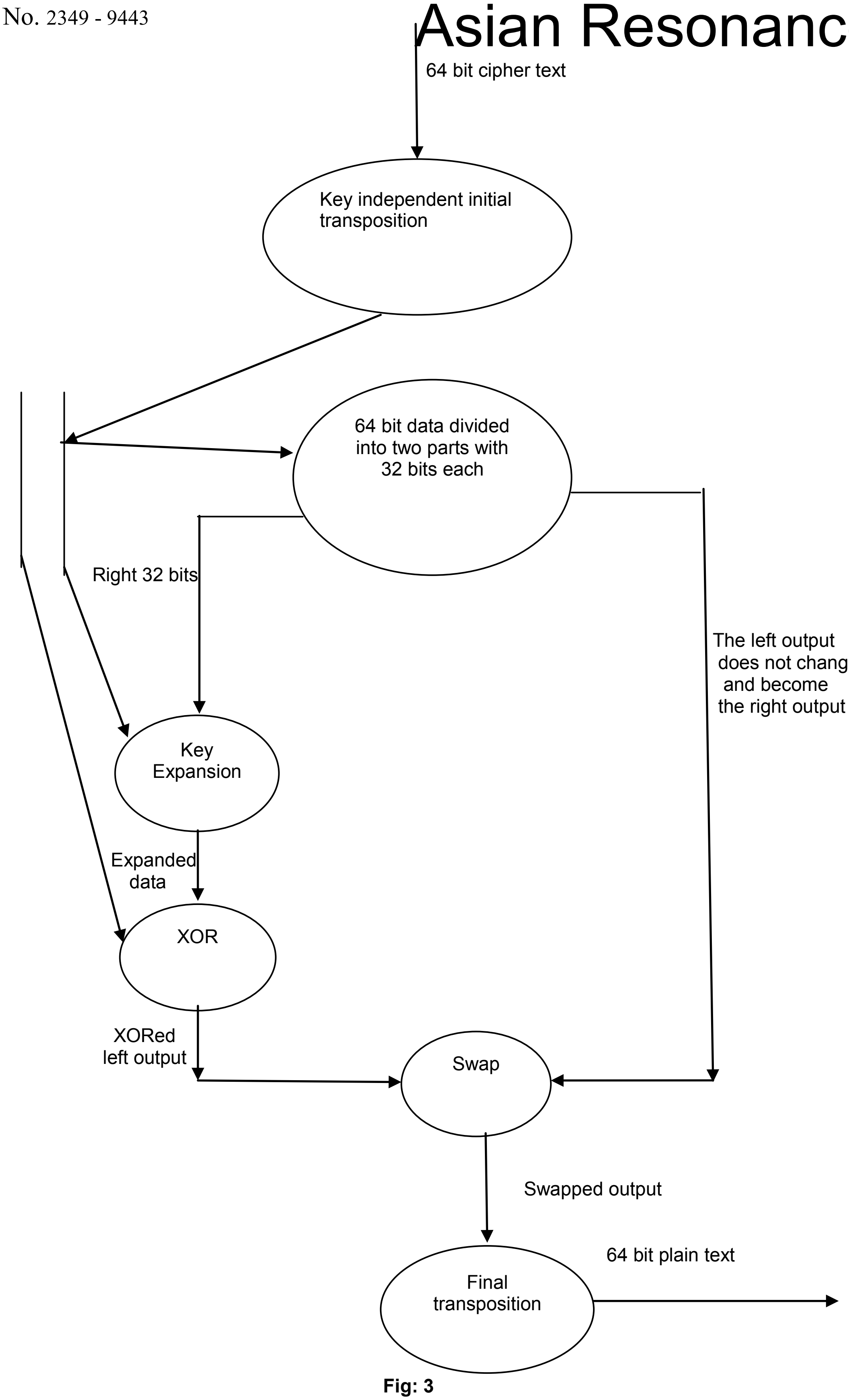
64 bit cipher text
Key independent initial transposition
64 bit data divided into two parts with 32 bits each
Right 32 bits
The left output does not change and become the right output
Key Expansion
Expanded data
XOR
XORed left output
Swap
Swapped output
64 bit plain text
Final transposition

Fig: 3
Level2 DFD for study material decryption

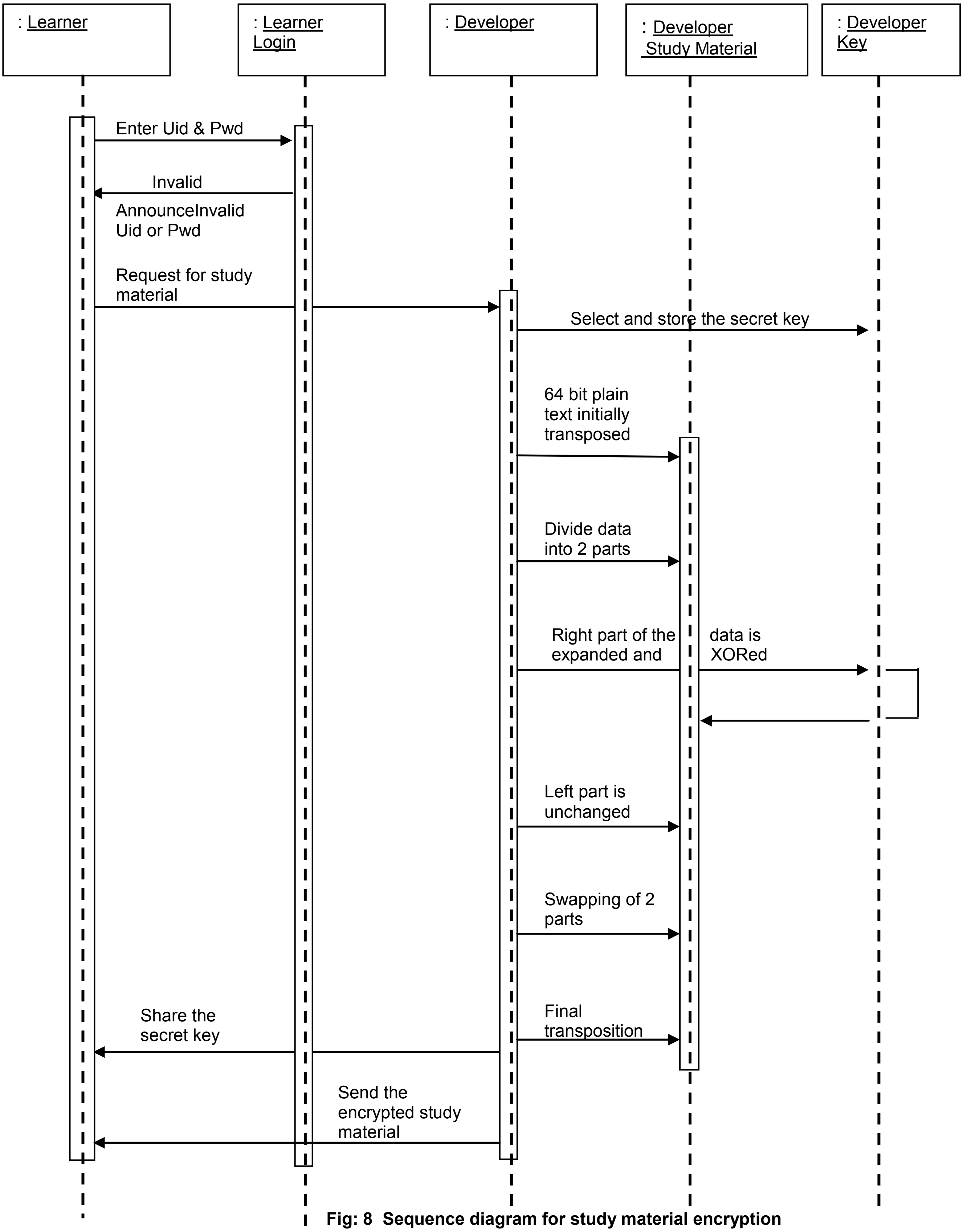
: Learner
: Learner Login
: Developer
: Developer Study Material
: Developer Key
Enter Uid & Pwd
Invalid
AnnounceInvalid Uid or Pwd
Request for study material
Select and store the secret key
64 bit plain text initially transposed
Divide data into 2 parts
Right part of the expanded and
data is XORed
Left part is unchanged
Swapping of 2 parts
Final transposition
Share the secret key
Send the encrypted study material

Fig: 8 Sequence diagram for study material encryption