ABSTRACT:

Storage systems are increasingly subject to attacks. Cryptographic document frameworks alleviate the threat of uncovering information by utilizing encryption and integrity security strategies and ensure end-to-end security for their customers. This paper describes a generic design for cryptographic file systems and its realization in a distributed storage-area network (SAN) file system. Key administration is incorporated with the meta-information administration of the SAN file system. The execution bolsters record encryption and trustworthiness security through hash trees. The two systems have been executed in the customer record framework driver. Benchmarks illustrate that the overhead is recognizable for some falsely developed utilize cases, however that it is little for normal document framework applications.

INTRODUCTION:

Data Security has always remained an integral and indispensable part of operating system. OS security refers to specified steps or measures used to protect the OS from threats, viruses, worms, malware or remote hacker intrusions. OS security encompasses all preventive-control techniques, which safeguard any computer assets capable of being stolen, edited or deleted if OS security is compromised. So a computer system must be protected against unauthorized access, malicious access to system memory, viruses, worms etc.

- Authentication
- One Time passwords
- Program Threats
- System Threats
- Computer Security Classifications

The basis of OS protection is separation. The separation can be of four different kinds:

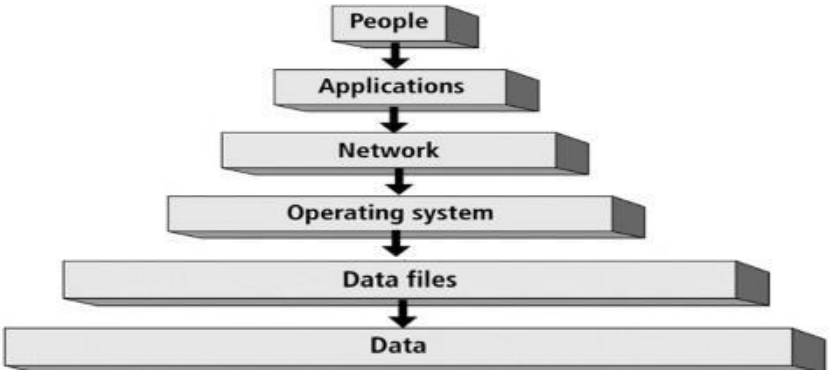
– Physical: physical objects, such as CPU's, printers, etc.

– Temporal: execution at different times

– Logical: domains, each user gets the impression

– Cryptographic: hiding data, so that other users cannot understand them

OPERATING SYSTEM SECURITY



Key areas that need to security measures in operating system:

1. User Accounts
2. Account Policies
3. File System
4. Network Services

OBJECTIVE:

- Understanding the need of data security and integrity to be included during design of distributed operating system.
- Types of cryptography modules used by different operating system store data and authorize users.
- Implement a file encryption technique using DES for the windows platform.
- Comparing the time and space aspects of decrypted file and encrypted file.

ALGORITHMS:

- RSA Algorithm
- S-DES Algorithm

OPERATING SYSTEM SECURITY STANDARDS

S.N.	Classification Type & Description
1	**Type A** Highest Level. Uses formal design specifications and verification techniques. Grants a high degree of assurance of process security.
2	**Type B** Provides mandatory protection system. Have all the properties of a class C2 system. Attaches a sensitivity label to each object. It is of three types. • **B1** − Maintains the security label of each object in the system. Label is used for making decisions to access control. • **B2** − Extends the sensitivity labels to each system resource, such as storage objects, supports covert channels and auditing of events. • **B3** − Allows creating lists or user groups for access-control to grant access or revoke access to a given named object.
3	**Type C** Provides protection and user accountability using audit capabilities. It is of two types. • **C1** − Incorporates controls so that users can protect their private information and keep other users from accidentally reading / deleting their data. UNIX versions are mostly Cl class. • **C2** − Adds an individual-level access control to the capabilities of a Cl level system.
4	**Type D** Lowest level. Minimum protection. MS-DOS, Window 3.1 fall in this category.

Development of A Secure OS

The development of secure OS can be made in six steps:

- Analyze of the system
- Choose/define a security policy
- Choose/create a security model (based on the policy)
- Choose implementation method
- Make a (conceptual) design
- Verify the correctness of the design
- Make an implementation
- Verify the implementation

There are feed-back loops between all of the above steps. Errors may occur in all above steps.

Trusted Operating System Concepts

There are a few basic concepts that are fundamental when dealing with trusted OS: the **kernel**: is the part of the OS that performs the lowest-level functions

- **Security kernel**: is responsible for enforcing the security mechanisms of the entire OS
- **Reference monitor** (RM): is the part of the security kernel that controls access to objects
- **Trusted computing base** (TCB): is everything in the trusted OS necessary to enforce the security policy

SCOPE

- First of all, user fulfill the agreement form of this program then user insert a text massage or include text massage file.
- Encryption key as an input as a result, this program will encrypt and decrypt this text massage and save it in a file.

HARDWARE CAPABILITY:

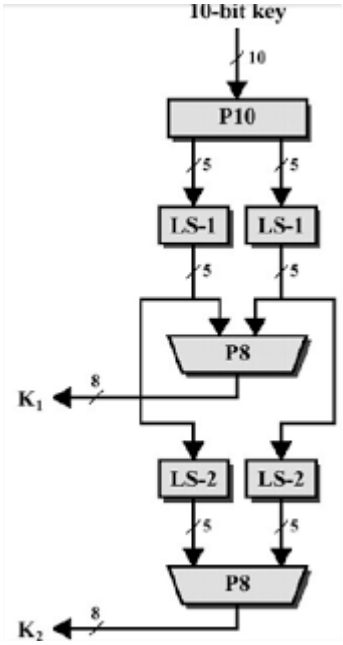
- The hard requirement for this project is minimum (Pentium 2) and onwards with 128Mb Ram and 2Mb hard disk.

ANALYSIS:

- **Key generator**
- We have analyzed the project that we will take an input (encryption key) from user, we will convert this key in to 10-bit binary. Then we will divide the 10-bit binary key in 5-5 bits. we will do LS-1(LEFT [1bit]) of these 5-5 bits. we will merge and arrange these bits according to the following rule.

Through this we will get a key(K1) of 8-bit.

- From those 5-5 bits of (LS-1), along with generating the key (K1), we will do LS-2(left shift [2bit]) by the help of above rule (P8) we have merged (LS-2), we will get key(key2).



Flow chart of key generator

Encryption detail

- We will take any alphabet or integer from the user and will convert into an 8 bit (IP BINARY), we will divide this 8-bit binary in to 4-4 bit as left 4bit and right 4bit **IP.** We will convert right 4bit binary into 8-bit binary and arrange it according to the following rule.

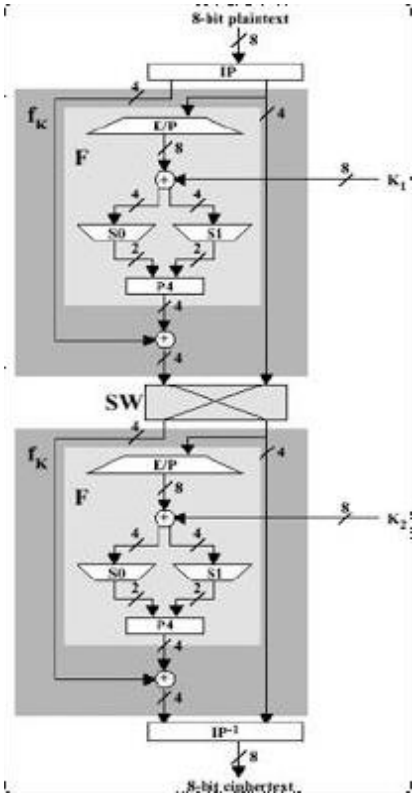
- After applying this rule, we will add (key 1) with it which we have already generated. we will divide these 8 bits into 4-4 bits from first 4 bots we will find so and from remaining 4 bits, we will find S1.

$$S0 = \begin{array}{c|cccc} P_{0,0}\,P_{0,3} \backslash P_{0,1}\,P_{0,2} & 0 & 1 & 2 & 3 \\ 0 & 1 & 0 & 3 & 2 \\ 1 & 3 & 2 & 1 & 0 \\ 2 & 0 & 2 & 1 & 3 \\ 3 & 3 & 1 & 3 & 2 \end{array} \qquad S1 = \begin{array}{c|cccc} P_{1,0}\,P_{1,3} \backslash P_{1,1}\,P_{1,2} & 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 2 & 3 \\ 1 & 2 & 0 & 1 & 3 \\ 2 & 3 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 & 3 \end{array}$$

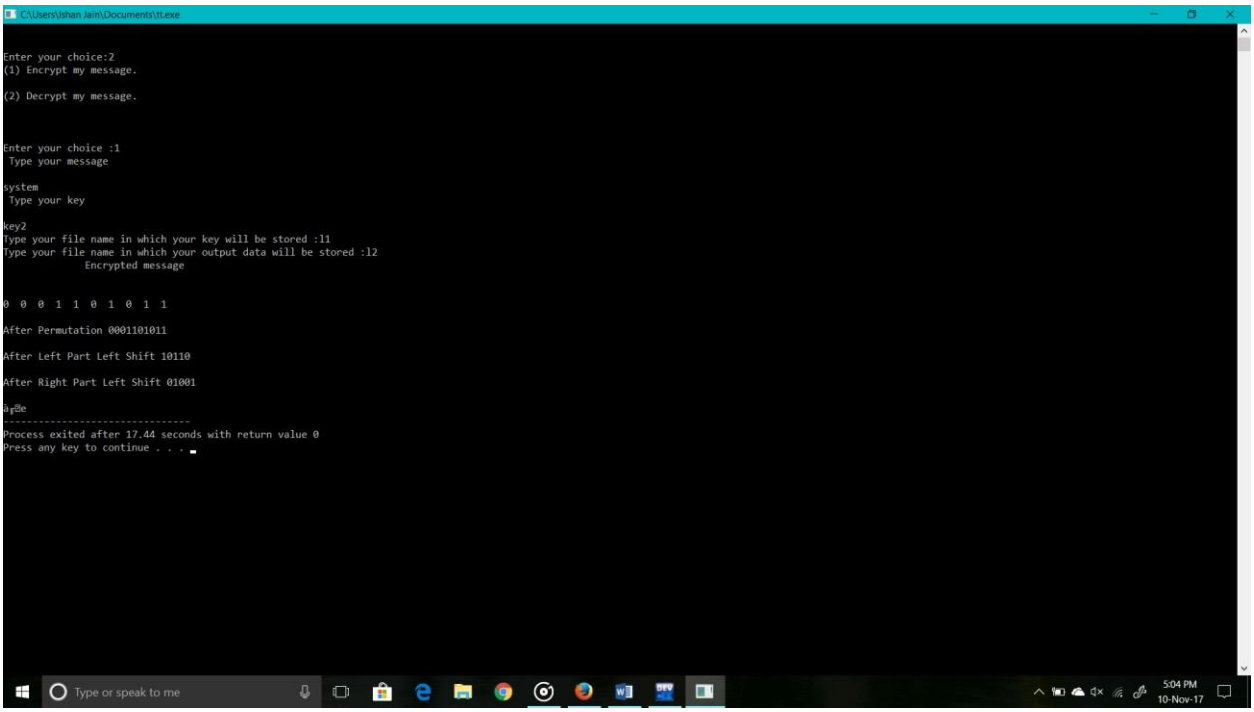
- After solving S0 and S1, we will get 2-2 bit binary and arrange them through RULE P 4.

P4			
2	4	3	1

- Now we will add the left 4-bit IP with this arranged 2-2 bit binary. by using switch function, we will repeat this method and use key 2 instead of key 1 as a result of this whole scheme the user 's given massage will encrypt.



OUTPUT:


C:\Users\Ishan Jain\Documents\tt.exe
Enter your choice:2
(1) Encrypt my message.
(2) Decrypt my message.
Enter your choice :1
Type your message
system
Type your key
key2
Type your file name in which your key will be stored :11
Type your file name in which your output data will be stored :12
Encrypted message
0 0 0 1 1 0 1 0 1 1
After Permutation 0001101011
After Left Part Left Shift 10110
After Right Part Left Shift 01001
Process exited after 17.44 seconds with return value 0
Press any key to continue . . .
Type or speak to me
5:04 PM
10-Nov-17