Def Sci J, Vol 37, No. 4, October 1987, pp 489-494

File Secrecy in a Multi-User Environment

Jawaharlal M. and Vivekananda Shetty
Gas Turbine Research Establishment, Bangalore–560 001

ABSTRACT

This paper deals with a method that provides in a multi-user environment file security of the order that even the system's people will be unable to break. The technique dealt with here is ASCII character by character encrypting of file using a KEY that is not physically stored anywhere in the magnetic media. This paper also covers problems encountered in the computer environment when using this technique.

1. INTRODUCTION

Computer Systems are being used more and more in all the fields of industries and Defence. With the increasing amount of sensitive information being concentrated at computer systems,the 'Information Security' becomes a primary concern. Access to sensitive information by unauthorised persons may lead to serious consequences, especially when it is related to national security.

The problem of information security becomes unmanageable when the computer environment is multiprogramming. Most of the Defence establishments in our country, besides using their own computer systems use centralised computer systems, organized and managed by dedicated computer centre like DRDO Computer Centre.

A number of measures are available on most of the mini and main frame computers to assure file secrecy and prevent the unauthorised access to other user files. Despite the amount of effort put on maintaining file secrecy, deliberate attempts can break the existing file secrecy maintaining procedure. Therefore it is highly important for the user to find a solution to keep the file secrecy independent of the features inbuilt in the system and the assurances and facilities offered by the system management people.

Received 20 July 1987

2. CURRENTLY AVAILABLE FILE SECURITY SCHEMES

2.1 Password Scheme

Most current systems use a type of password scheme with a high number of possibilities, so that the person unaware of the particular password cannot access the files. But this scheme could be broken by manual trial and error method. If it is very time consuming, a small computer could be used to break the code and breaking of code of four letters by this method would hardly take any amount of time.

2.2 Password Encryption

Some systems encrypt password itself so that it cannot be broken by above mentioned method. Since the password in this type of protection is encrypted just before putting the file on the disk, it could be broken by anybody who has a good knowledge of operating system.

2.3 Time Dependent Password

According to this, the password can be used only for a predetermined, definite time period. After the specified time is over, the password becomes invalid. This is useful, when an outside user is allowed to work at the site for a temporary period.

2.4 Invalid Login Record

When an invalid login attempt is made, it is immediately recorded at the central site, so that the system management people can come and identify who is trying to make an unauthorised login to the system.

2.5 Secure Login Procedure

This procedure was developed as a counter measure against the pseudo-login procedure successfully used by several students to find user names and passwords running a program on the terminal which simulated a login procedure. After the desired information on password and user names were stored on a file the system faked a crash and user would login this time into the operating system. This problem is taken care in some systems.

2.6 Security Count

A security count can be set for every user and when the user makes an unauthorised attempt, his security count goes down by one. When the security count reaches zero that user cannot anymore use the system unless the system management personnel reset his security count.

2.7 Memory Cleaning

As soon as the user leaves the terminal, the files (local files) in the solid state memory devices are completely scavenged and nothing is left for anybody to pirate.

2.8 Directing File Access Control

The users can define the files as say, private/semipublic/public, so that only himself or a specified group of people can access specified files using a password which may be time dependent.

2.9 Access Levels

There can be different File Access Levels by which a Lower Access Level (LAL) user can pass on information to Higher Access Level (HAL) without reading the HAL files, whereas HAL user can read the LAL files without downgrading classified information to a lower access level.

2.10 Access Categories

There can also be different Access Categories which can be allocated to different user groups. In this method, one user group cannot access the files of other groups and inside every group there can be different access levels.

3. SECURITY LAPSES

The above mentioned File Protection Measures are generally available on most of the current mini and mainframe computers. Looking at these counter measures, one can understand that there are many loopholes in the system.

(a) Tape storage – Files stored on magnetic tapes are easily accessible to unauthorised persons, especially if it is taken to a different computer environment.

(b) Software – Some software do not support many of the protection schemes. They can be used and copied by all.

(c) System management personnel – In the security chain, assuming that everything is perfect, the weakest link will be the human factor. In most of the systems, system management personnel can see user passwords and therefore can access confidential files. In some other systems, except the user nobody else including the systems personnel can know the password. Though apparently a good system this allows the systems personnel to change users passwords to a new one without knowing the old one and access all files. Therefore, a user can ultimately only be at the mercy of systems people.

Hence in a Defence R & D environment with enormous amount of important data being processed, the user needs to have a foolproof system – a system that could protect the file secrecy from all types of possible breaking.

To overcome the above limitations a facility to encrypt all the data in a given file is suggested. The field of encryption is a well developed science.

Before coming to the algorithm, it would be better to look at what type of data one would encounter in a computer environment. The data that is used by a computer is in two forms namely, (a) Set of machine instructions in binary form, (b) Text or data-base data and program listings.

The binary form will start with a beginning of information (BOI) and end with an end of information (EOI) while a text or data base file will also contain markers like end of record (EOR) and end of file (EOF).

It is also important to examine the physical locations that this data could exist magnetic tape, magnetic disk, semiconductor memory, and paper or cards. It is obvious that the only way to protect a printed output is by physical security and in all cases the semiconductor memory is cleared when the user logs out. Therefore it can be seen encrypting a file is necessary only to protect the file when it is on hard disk or on magnetic tape.

As already mentioned, the data when stored in a file, consists of ASCII characters. Even in the case of an executable binary file, the instructions will be a combination of ASCII characters. A set of 6 to 8 bits can be taken to represent an ASCII character. This means that the file can be encrypted by encrypting the ASCII characters into a modified set of ASCII characters. There is a disadvantage of embedding the password in the file in that the person who developed the program will be able to retrieve it due to his technical awareness of the system.

4. ALGORITHM FOR FILE ENCRYPTION

The algorithm suggested by us works in this way:

(i) User runs encrypting/decrypting program.

(ii) Program asks for password to execute (this password is encrypted on the disk).

(iii) If password was keyed O.K. the program is executed else security count is decremented by one.

(iv) The user is prompted for encryption or decryption both taking two different path in the program.

(v) The user is then prompted for data file name.

(vi) The user is prompted to identify the file type (Text or Binary).

(vii) Then the KEY using the operation is prompted for, once the key is entered, which like all password facility is not displayed on the console. The KEY is then verified. This is done because the KEY is not verified from the disk like a password. In case the KEY was entered wrongly this is the only way to verify, since KEY does not exist on the disk.

(viii) The KEY is then standardised to a specific length so that the process of operation is easy. This may be done by generating the remaining portion of

the key by a mathematical manipulation of the KEY that was typed in. e.g. in case a 5 letter key is standard and 3 letters were entered then the 4th character may be the length of the typed in KEY and 5th character the ASCII sum of the typed in key wrapped around from maximum ASCII value to zero ASCII value.

(ix) Using this standardized key the file is encrypted doing a one byte file to one byte of KEY manipulation so that Ist file byte is encrypted by Ist KEY byte and the 2nd file byte by second key byte and once the KEY length is over then the KEY is repeated till the whole file is completely encrypted or decrypted.

(x) In case it was a text file then the logic may be slightly changed to incorporate only printable ASCII character instead of the whole ASCII range.

(xi) In case the file was to be decrypted the whole process would be the same except that the mathematics of byte by byte encrypting would be exactly in the reverse direction. (i.e. if the encrypting logic was an ASCII value addition decrypting would be an ASCII value subtraction using KEY).

(xii) Once the operation is over the file will be saved onto the hard disk. And this could later be saved on to the magnetic tape for backup purposes.

In the process of encrypting, it can be seen that what ever logic you use one may generate EOI markers which will cause the program or data loader to load only a part of the full file. To over come this problem two techniques are suggested. The first is applicable if the EOI is either at the top or the bottom of the ASCII set, then this along with markers like BOI, EOF, EOR will occur at the same set and these may be eliminated by choosing a samller ASCII character set to use while encrypting. In case the ASCII, EOI marker is embedded in between the ASCII set, then one to one mapping with one of the ends of the ASCII table may be devised so that a smaller character set may be used and in case an EOI is generated during the encrypting procedure then a one to one mapping with the last or first character is done even though it is outside the given character set range. This can be taken care during the decrypting session.

5. LIMITATIONS

The main limitation of this system is that loss of the KEY would mean loss of the entire data since the KEY is not resident on the magnetic media.

The second limitation is the time required to encrypt and decrypt the data. This would consume a time equal to two read stages and one write stage to encrypt the data and the same to decrypt it every time the data file is used.

BIBLIOGRAPHY

Walker, B.J. & Blake, I.F., Computer Security & Protection Structure (Dowden Hutchinse & Ross Inc), 1977.

NOS Version 2, Reference set, Vo. 3, Publication No. 60459680, CDC Computer systems, 1985, pp. 2-1 to 2-17, 3-11 to 3-17.

NOS Version 2, Security Administrator's Hand Book Publication No. 60460410, CDC computer systems, 1985.