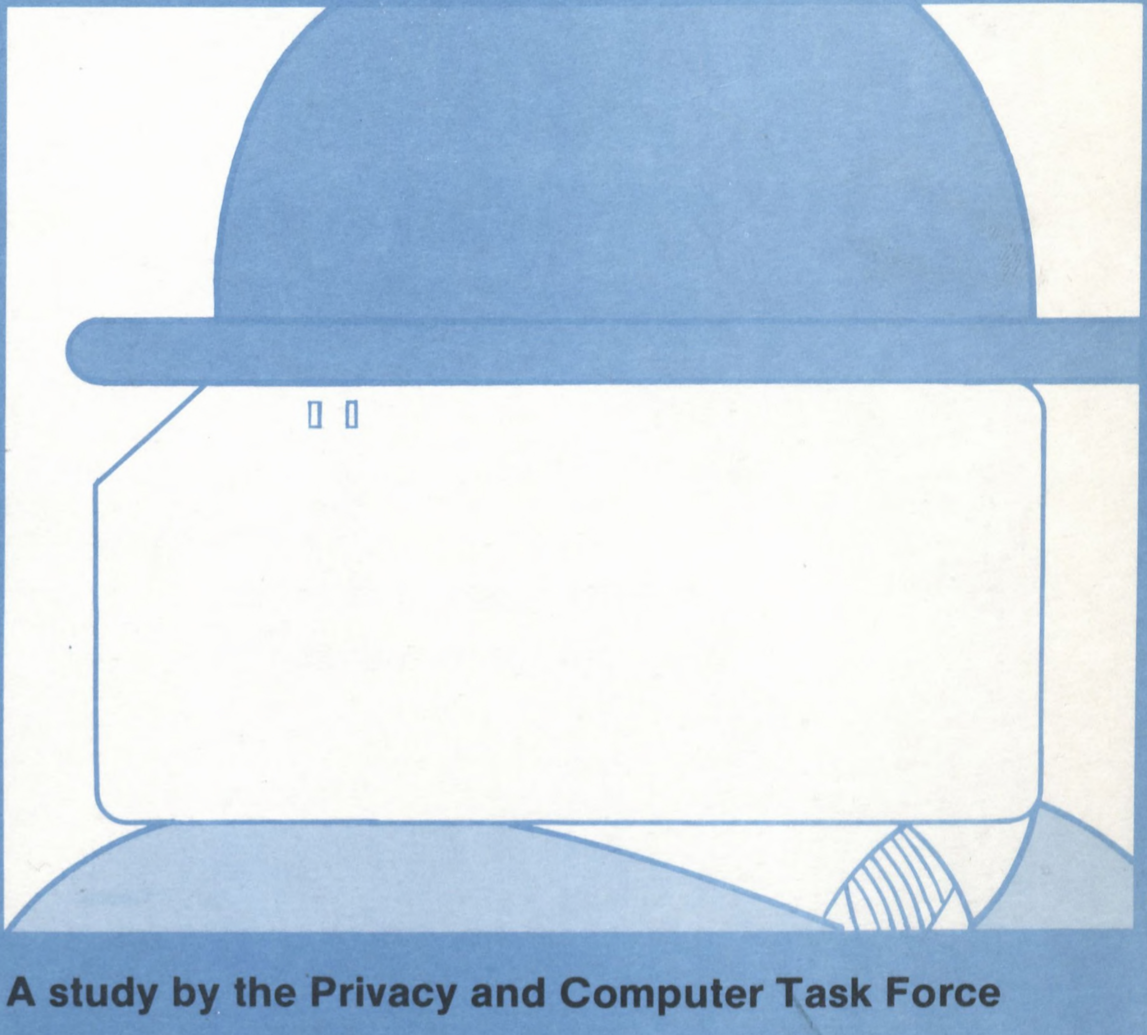
# SYSTEMS CAPACITY FOR DATA SECURITY

C.C. GOTLIEB AND J.N.P. HUME



4 A study by the Privacy and Computer Task Force

new

# SYSTEMS CAPACITY FOR DATA SECURITY

A STUDY FOR THE

PRIVACY AND COMPUTERS TASK FORCE

DEPARTMENT OF COMMUNICATIONS

DEPARTMENT OF JUSTICE

C.C. Gotlieb
J.N.P. Hume

This report was prepared for the Privacy and Computers Task Force, an inquiry sponsored by the Department of Communications and Justice, and should not be construed as representing the views of any department or of the Federal Government. The views expressed herein are exclusively those of the authors, and no inference of any commitment for future action by any department or by the Federal Government should be taken from any recommendations contained herein.

This report is to be considered as a background working paper and no effort has been made to edit it for uniformity of terminology with other studies.

# TABLE OF CONTENTS

Part I

# Methods for Maintaining Data Security In Large, Shared Computing Environments

## 1.1 Introduction

The protection of data against the deliberate or accidental access of unauthorized persons is achieved by data security. It is the purpose of this study to examine the methods by which data security can be assured for automated information systems. Ultimately the security of data depends on some combination of locks - access control measures - for which certain users possess the key. No such combination is completely secure; the extent of security is really a matter of the cost, to the intruder, of bypassing the combination of locks relative to the value, to him, of obtaining data in this way. In turn, for someone wishing to maintain the security of data, the cost of devising and implementing a combination of locks on the data must be small relative to the cost of a breach of security. In the case of, for example, military intelligence data banks, the information contained in them is considered to be of such value that almost no cost is spared to ensure data security. Such systems, however, are clearly exceptional and this paper deals instead with commercial or public data banks where there are clear limits to that amount of high-cost security measures that can be justified.

An attempt will be made to estimate the cost of various security measures; the comparison of cost of combinations of locks with value of the integrity of the data itself can only be made by the user. As a general rule all reasonable security measures should be taken whenever data is sensitive; the more sensitive the data, the more sophisticated (and costly) can be the protections. Some suggestions for such measures follow. It is to be noted that in a computer system the protection of the data itself and of software search and retrieval programs are treated almost entirely in the same manner; thus we discuss the same safeguards for program security as we do for data security.

A list of general references that proved valuable to this Study is given at the end of this report. The work of Lance J. Hoffman was particularly valuable in his survey paper on "Computers and Privacy" in Computing Reviews. Dr. Hoffman has also offered helpful criticism of this manuscript. The authors participated in several of the Task Force site interviews and information from these is included. Specific references were not made to companies since it was understood that the identification was to be confidential. The summary of current practices of Part II was done by Mr. Frank Tompa under our direction.

## 1.2 Classification of Degree of Confidentiality

In this section the term "user" stands for a single person, or a group of persons all of whom have equal rights with respect to accessing a particular body of data and a common identity to the system. We will define three classes of data for an automated system: public, limited-access, and private.

Type 1 Public data is open to all users. For this data, no security measures are necessary as far as reading is concerned. When access is restricted to reading of the data, as it should be where data must remain unchanged, writing should be prevented. If it is not possible to prevent writing, checksums (a simple total of all data items) can be kept with data that should remain constant, refreshing the data from a secure copy whenever a test total of the data does not agree with the checksum. If the data may be altered by users, a lock must be maintained on the system to ensure that while one user is making a change no other user is permitted to access the data since one user's alterations must be completed before another may begin.

Type 2 Limited-Access data is open only to authorized users. This means that an authorization table must be kept in the system indicating, for each body of data, the identity of all users with access rights. When a user requests access (a) his identity should be authenticated, (b) the authorization table should be checked to see that he has appropriate access rights, (c) a record in a log should be made of the event. This log would provide an audit trail or record which could be consulted whenever any trouble is suspected. All unsuccessful attempts to access data should be logged in order to provide an indication of a possible security leak. If the frequency of unsuccessful entry is larger than normal error warrants, an alarm might be generated.

Type 3 Private data is open to a single user only. When access to data is requested, the identity of the user should be authenticated to verify the fact that the data is his. Here again a record of all unsuccessful attempts at entry should be logged.

In this report, we will be concerned primarily with Types 2 and 3 as it is only in these two cases that there is a confidentiality issue. (We have used the term private only for Type 3 but in a sense the Type 2 limited-access data is private to a group of users. Many systems permit the user with a private body of data to enlarge the access rights to the limited-access category and even to the public category).

The maintenance of security has larger costs for limited-access data than for the other two types. It would be simpler if data could be classified as either public or private but there are many reasons for wanting the middle ground of limited-access, for example, in accessing an inventory file of a company or institution.

## 1.3 Access Rights to Data

We have referred to a body of data and indicated that this may be a program. Usually data that is not a program is organized into discrete files. A file is composed of a number of records, or factual statements, each relating to a particular thing or, in a file containing personal data, to a particular individual. We will restrict ourselves to files containing personal data, i.e. those with some sort of personal identification in each record.

A record, in turn, is subdivided into fields. A field is a precisely defined location in a record where information may be recorded. For example, many printed forms have spaces where information is to be entered. Such spaces could be called fields. Each field of a record contains an item of information pertaining to the person. Certain fields enable the reader of the record to identify the person; this may be his name or address or some other number that is unique to the individual. Records in a file are organized systematically according to one of the fields which is called the key for that particular organization. Records may be rearranged using another field as a key. The layout of information of a record is usually constant from one record to the next although some fields may be of variable length, for example those giving the names of children of a person.

It has been suggested that access to a file of personal information should be on the basis of "need to know" and that access to a particular record in a file should be on the basis of an explicit or implicit consent of the individual to whom the record pertains. It would therefore follow that if a person having access to a record needs to know only the information in certain fields of the record he should not have access to other fields in the same record. For example, persons who are preparing statistical summaries from files, do not need to know the identity of the person to whom each record applies and therefore should not have access to identifying fields.

Most often, persons having access to a file have access to <u>all</u> fields of <u>all</u> records. In a manual file in which records are maintained in a manila folder it is difficult to arrange to do otherwise. In a computerized system, however, access can be permitted to the entire file or can be restricted to certain fields of the file.

Access rights might be to

1) read an item (e.g., file, record, or field);
2) write an item so as to produce a change -- either
   (i) a new item added, or
   (ii) an existing item changed;
3) delete an item.

The access rights of a user must be explicitly denoted in any situation where partial rights exist, e.g., for a limited-access file, or where reading is permitted but changes and deletions are not. It is possible to have a table or matrix stored with the data (or separately) showing a list of authorized users of the data and their access rights. Access to this table must be strictly limited to persons authorized to modify the table, usually only the owner of the data himself. In many cases access control is assigned to the system itself since in most computer systems operations pertaining to the read or write functions are already under system control.

## 1.4 Physical Storage of Data

Data in an automated system can exist in many physical forms. Some storage devices are an integral part of the computer hardware, some may be attached as a part of the hardware and others are read or written by the computer system on a medium that may be removed. We will classify the storage media into five categories as follows:

1. Hard copy is a term used for a recording of the data that is more or less permanent and that can be stored, read, or written by humans independently of the computer hardware. Included in this category are printed pages, punched paper tape, punched cards, and microfilm. The security of hard copy is similar to the conventional security associated with manual

files. The interpretation of the data by an intruder is usually very simple. Also the destruction of the data requires the destruction of the medium. Machines for shredding hard copy are available for $400-$4000 depending on their capacity.

2. Display devices are devices on which data may be exhibited to a user but on which it has an evanescent form. As soon as it is no longer required it will disappear. An example of this form of storage is a cathode ray tube display. If such devices display sensitive data, they may need to be used in secure rooms where unwanted cameras or persons cannot observe them. When electric circuits are arranged to display images on one cathode ray tube, stray electromagnetic radiation from these circuits might be amplified to produce a similar display on another such device that has no connection to the first. Display devices like printers or card readers should be appropriately shielded to guard against the possibility of electromagnetic eavesdropping.

Carroll notes that the possibility of exploiting the near-field radiation from computers and their associated input/output devices has been thoroughly explored by the Government of Canada. The extent of this threat and countermeasures against it have been defined in detail. This information is available from the Industrial Security Branch of the Department of Supply and Services to computer users having a bona fide need to know. Carroll also points out the possibility of telephone instruments acting as pickup devices for such radiations even when on the hook.

3. Magnetic tape and mountable disks are media on which data can be recorded as variations in magnetization. These media can be erased and used repeatedly although, as with any erasing process, small traces of previously recorded information may persist. When the tapes or disks are not mounted on a computer they cannot be read but they can be erased or destroyed, for example by strong magnets or fire. A careful banking system in secure rooms under strict control must be maintained to prevent loss or violation of security during off-line storage. When tapes or disks are mounted on the computer they become identical in nature to the next category (#4 below) of physical storage; when off, they are identical in nature to hard copy.

For one large shared computer system that was investigated in Canada it was admitted that off-line storage was the only way that data could be kept secure. Since tapes and disks were only mounted upon proper authorization from the owner the possibility of them being violated while on-line was much diminished over permanent on-line storage.

4. Mounted magnetic tapes, disks, and drums are integral parts of the on-line storage system of a computer. They are usually classed as the secondary store since access to information stored on them requires a length of time that is long compared to the basic operation rate of the computer. Usually the time of reading from, or writing on, them is overlapped by other operations. This means that the individual user does not direct the reading or writing

himself but goes through the intermediary of the computer operating system. Access control almost always resides in the operating system.

5. Magnetic core store is the main first-level store of most computers. A single core stores one bit of information and may be used again and again. To operate, a program must be in the core store; to be acted upon, data must be in the core store. Thus, in the final analysis, on-line access to data must first be controlled by controlling access in the core store. Since users are permitted to use core store one after another, it is important to erase any sensitive data that is there before allowing the next user's program to have control. Some operating systems do this clean up job automatically; others do not.

## 1.5 The Operating System

It is a common saying that "computers will do only what they are told to do." There is always a person involved in giving instruction to the computer. If the instruction is that the computer disclose, alter, or delete information that it has in its possession, then the right of the person giving the instructions to know, change, or erase that particular information should be challenged.

Present-day computers are usually operated in a way that permits a number of different users to have sets of instructions (programs) active concurrently (multiprogramming). These would be accessing different pieces of information stored in the computer, either in its immediate access store (usually magnetic cores) or in its secondary storage devices (usually magnetic drums, tapes, or disks). Thus to avoid chaos, the problem of keeping the users that are sharing the computer from interfering with each other's information has had to be met, regardless of the confidentiality issue. The method used is to have a set of instructions that is a part of the computer system itself always present and actively supervising or monitoring the operation of the users' programs. This program is called the operating system (or supervisor, or monitor program). All, or some portion, of the operating system must always be resident in the core store of the computer, the rest of the core being shared by a number of users (see Fig. 1).

| | | | | |
|---|---|---|---|---|
| Operating System | User 1 | User 2 | User 3 | Shared or Common Store |

Fig. 1. Allocation of Core Store

## 1.6 Protection of Data in Core Storage

Each user is assigned a region of the core store as his own private domain. The right to read from or write into this area of the store is protected by a key (through a hardware device). A directory of keys is kept in a table in the core area assigned to the operating system itself. This means that the user cannot alter the directory entry pertaining to his own core area. If he could, it might permit him to access some other user's core area. Each user core area is thus private to an individual user.

Sharing data and programs can be achieved by having an area of core that is common to the users concerned. If only specified users may share the data in this area an authorization table must be maintained. Often it is sufficient to declare the area as public in order to ensure that no access control is necessary.

It has been suggested that the operating system might exist in a read-only store, i.e., where the writing operation is inhibited. This might protect the operating system program from being altered but should not be necessary if proper memory bound protection is available and the operating system is error free, which unfortunately is never the case.

## 1.7 Protection of Data in Secondary Storage

In a secure system, all requests to read and write on secondary storage must pass through the input-output control of the operating system. In order to issue a read or write instruction to a file in secondary storage it is necessary for a user to alert the operating system that he intends to perform operations on the file by issuing an instruction. At this time, his access rights to the file are examined. Private files are usually labelled with the name and system identification number of the user and may even contain in their label a password that must be matched against one provided by the user at the time of issuing the access request. Only the owner of a limited-access file should be permitted to change the password. It may or may not be possible for the operating system to read a file that has been password-protected. If it cannot, the file will become useless if the user forgets his password.

When a file has limited access, the access information is frequently stored separately from the data. If the file is stored on magnetic tape it is usually accessed serially, one record after another. In this case when one user is accessing the file, no other user may access it as the positioning of the read-write heads on the tape would be altered. Files on a single magnetic tape cannot be used concurrently by two users so the system must lock out all requests after the first until the first user has closed his file and rewound the tape. Disks and drums are constantly in motion and files on them may be accessed by multiple users more or less concurrently.

## 1.8 Protection of Data in Transmission

Wiretapping or electromagnetic eavesdropping is a security threat whenever data travels over wires that are not in a secure area. Many systems use common carrier facilities and here the problems are well known. Sensitive data that is to be transmitted from one location to another should be transformed, i.e., encrypted to make it private. Privacy transformations that involve static methods of encryption require a certain amount of work to break the code but are usually subject to decoding after some effort. The best encryption techniques involve keys that are as long as the data to be encrypted. The string of characters for the key is generated from a basic starting number, just as a sequence of pseudo-random numbers can be generated. The same starting value yields the same sequence every time. It is nearly impossible to determine the starting value and the generating algorithm from eavesdropping on the transmission. The work factor to break the code is very high.

If the data is produced at one end of the line and is to be processed at the other there needs to be a device at each end of the transmission wire, at one end to encode, at the other to decode. Each device must be able to generate the same key string, which requires that each have the starting number. The starting number should not be transmitted but instead sent by a secure method (e.g. voice) from one end of the line to the other. It may be changed as frequently as desired provided both ends know the pattern of change to be used. Often simpler encryption methods are used using shorter keys or ones that do not change. They have a much lower work factor.

## 1.9 Protection of Data Off-line

Stored data in the form of hard copy, or on either magnetic tape or removable disks must be kept in a strictly controlled environment. Protection against accidental or wilful damage or theft must be ensured. Access to the data will be basically through a manual system managed by a person charged with its security. There should be a record kept of all deposits and withdrawals from the data bank, it being assured that the person making the transaction has the appropriate access rights. Frequently data banks are located near the main computer installation where tapes are requested frequently, but it is common also for systems to have a separate repository of tapes containing data vital to regenerating the system. A remote storage vault in a protected location is essential for basic business or industrial data. This, however, is not a question of confidentiality but of preservation.

This study is largely related to the security of data as it relates to privacy and thus the preservation of data against destruction is not emphasized since, if data no longer exists, there can be no question of confidentiality. There is the possibility of large scale theft of files. As far as many companies are concerned, loss of company records by destruction or theft is more catastrophic than a privacy invasion. One large Canadian insurance company keeps its data processing operation in a "relatively bomb-proof" location miles away from its main business operation and expects in the future to move it "underground".

## 1.10 Integrity of Hardware

The fashion of having computers prominently displayed to the public is dying out. The need for precise environmental control has always meant that hardware was housed in special rooms but it is increasingly apparent that protection and control of access to the rooms is critical to security. Not only could the equipment be destroyed but also data could be compromised as it was being printed or displayed. All persons having access to the rooms where hardware is kept should be properly identified and "need to be present". Systems of identification badges are common. Sometimes access is controlled by a security officer, sometimes by locks opened by badges or by combinations. The advantage of locks operated by badges or push button combinations is that the combination may be more easily changed than changing locks operated with ordinary keys. Thus if there is any suspicion of a compromise of a lock the combination may be altered. When the key is a badge the rightful owner may have his picture displayed on it for a double check.

Many installations require visitors as well as regular staff to wear authorization badges when in a computer center. Unfortunately many of these badge systems are unenforced and provide only a semblance of security.

Where a piece of hardware is attached to the main computer hardware through a remote connection, the terminal equipment is often under minimal or no surveillance. As a result highly sensitive data is rarely handled in a computer system with remote terminals. It is important that remote users be properly identified and that the terminals be properly identified not only at the time of beginning a "conversation" but from time to time during any extended interaction. This is rarely the case except in defense systems. Remote terminals provide an infiltrator with some of the largest loop-holes in a system.

## 1.11 Integrity of Software

We have indicated that the security of data within the computer depends on the operating system. Many existing systems are complex, and the impression exists that their complexity protects them from invasion. And it does, to some extent. However, accidental access routes (trap doors) into the system have been found. When access routes via trap doors are found it is possible to cause the system to become inoperative; many of the breakdowns that occur daily in systems are the result of accidental entry into the operating system by an unsuspecting user. For smooth system operation, it is important to eliminate bugs that cause breakdowns. Hardware failures account for some, but surprisingly few relative to software failures. Most operating systems are up-dated regularly, sometimes to make improvements, frequently to correct (or attempt to correct) mistakes.

To be secure, an operating system's structure should be cleanly designed and the documentation openly available. Secrecy should not be a requirement for a secure system. The incentive to have better operating systems is great quite aside from privacy considerations.

It has been suggested that only critical parts of operating systems need be under strict security control, e.g., tables of access rights and the programs that validate these rights, to ensure data security.

Meanwhile the provision of audit trails of activity and alarms that are triggered by unusual behaviour of a user should be a part of all operating systems. A record of use is a part of the accounting procedure of all systems but it should also include events that may entail no charge to the user but be of significance to a security officer, such as incorrect passwords.

Whether it is part of the accounting procedure or not, it is important to have a record of all accesses to limited-access data to provide an audit trail that might, after the fact, either help catch the unauthorized user or improve the system. Such a record might show whether other users were trying to seek an unauthorized way into files. An alarm can be given to users or to the system security officer by the operating system if peculiar events occur. Action, by way of refusing further service to any suspect, might accompany the alarm.

## 1.12 Integrity of Personnel

Anyone who is entrusted with access rights to data is a potential security leak for that data. The most direct method of access for an intruder is often through a person in a position of trust. This problem is well known and many methods are used to decrease the probability of a betrayal of trust. Personnel are investigated to determine trustworthiness, stability, freedom from behaviour that might lead to their being compromised, and so on. The clearing of personnel is essential where data is highly sensitive. It should also be made explicit what the penalty is for breach of trust, e.g., firing, so that there will be a deterrent against disclosing sensitive information. Above all, chance of accidental disclosure should be minimized by having precise security procedures with regard to labelling of sensitive data, locking of file cabinets, etc. The problems here are identical with those of security of data in a manual system.

Part II

## Summary of Current Practices

### 2.1 Introduction

Data have been gathered on security measures in several computer systems. The systems examined are principally time-sharing systems, text-editing systems and multiprogramming systems. For each system the information is tabulated by means of certain headings so that a comparison can be made of the various features. An attempt is made to show the cost of security measures in terms of bytes of program required to administer the measure, number of accesses to secondary storage (disk) required, and number of bytes required to store data such as access tables. The systems described, while designed in the United States, are being used in Canadian installations. Some are still in construction and one is a hypothetical ideal system.

The systems on which information has been gathered and is displayed in the following sections are:
ATS (Administrative terminal system, IBM), FRESS (File retrieval and editing system, Brown University), A Problem Solving Facility (Hsiao: Ph.D. thesis, University of Pennsylvania), APL-PLUS (A programming language, I.P. Sharp & Assoc. Ltd.), CPS (Conversational programming system, IBM), MTS (Michigan terminal system, University of Michigan), Friedman's Proposed System (IBM System Journal, 1970), PDP/10 Monitor (DEC), CP/CMS (Control program/Cambridge monitor system, IBM), O/S (Operating system, IBM).

## 2.2 *ATS - Text Editor for /360*

| CATEGORY | | | BYTES OF PROGRAM | DISK ACCESSES | BYTES OF DATA |
|---|---|---|---|---|---|
| | System | ATS -- text editor for /360 | 160K | - | 80K |
| SIGNON SIGNOFF | Identification | account number; changeable password | 1000 | 2 | 24 |
| | Authentication | none | 0 | 0 | 0 |
| | User options | none | 0 | 0 | 0 |
| | Accounting | duration of use; (needed for billing anyway) | 500 | 2 | 6 |
| FILE USAGE | Determination of file names | files in account listed on request | 4600 | 5 | 16 |
| | User identification | five characterspasswordsfor another account's file | 48 | 0 | 10 |
| | User options | none | 0 | 0 | 0 |
| | Accounting | date last stored | 50 | 0 | 8 |
| | Type of capabilities | read-only, delete-only (separate passwords) | - | - | - |
| | Determination of capabilities | available through passwords | - | - | - |
| | Security scope | file | - | - | - |
| | Cryptography | none | 0 | 0 | 0 |
| PROTEC-TION | Separation from data | kept in separate directory | - | - | - |
| | Integrity considerations | no user programs | - | - | - |
| | Protection from concurrent systems | very poor -- directory may be erased | - | - | - |
| | Back-up | separate working storage | 1700 | 7 per 256 bytes | 99+266 per 256 bytes |
| | Residual information protection | password of file made unmatchable after delete | 6 | 0 | 0 |
| VIOLA-TIONS | Standard response | messages only | - | - | - |
| | Non-standard responses | disconnected for error in signon | - | - | - |
| | Comments | information obtained from U of Toronto computing ctr. | | | |

## 2.3 *FRESS - Text Editor for /360*

| CATEGORY | | | BYTES OF PROGRAM | DISK ACCESSES | BYTES OF DATA |
|---|---|---|---|---|---|
| | System | FRESS -- text editor for /360 | 200K | - | - |
| SIGNON SIGNOFF | Identification | account number; fixed password | 100 | 1 | 120 |
| | Authentication | none | 0 | 0 | 0 |
| | User options | none | 0 | 0 | 0 |
| | Accounting | data and time of previous signon | 10 | 0 | 8 |
| FILE USAGE | Determination of file names | provided by user only | 0 | 0 | 0 |
| | User identification | several changeable passwords per file | 500 | 1 | 40 per user |
| | User options | none | 0 | 0 | 0 |
| | Accounting | user's initials and date of use being implemented | 150 | 1 | 1000 |
| | Types of capabilities | every command is a separate capability | | | 40 per protected field |
| | Determination of capabilities | password-related keys stored with each protected field | 500 | 1 | 1000 |
| | Security scope | field | - | - | - |
| | Cryptography | file names encrypted | 6 | 0 | 29 |
| PROTEC-TION | Separation from data | separate software page for protection information | | | |
| | Integrity considerations | none -- debugging system is a trapdoor; no user programs | - | - | - |
| | Protection from concurrent systems | poor -- only encryption of file names | - | - | - |
| | Back-up | every edit written directly onto the disk file | 10 | 1 to 5 | 0 |
| | Residual information protection | none | 0 | 0 | 0 |
| VIOLA-TIONS | Standard response | messages only | - | - | - |
| | Non-standard responses | none | - | - | - |
| | Comments | system still under development at Brown University | | | |

## 2.4 *Information Retrieval for PDP/10*

| CATEGORY | | | BYTES OF PROGRAM | DISK ACCESSES | BYTES OF DATA |
|---|---|---|---|---|---|
| | System | information retr. for PDP/10 | 28K | - | |
| SIGNON SIGNOFF | Identification | name, project number, project name | | 1 | 30 |
| | Authentication | none | 0 | 0 | 0 |
| | User options | none | 0 | 0 | 0 |
| | Accounting | none | 0 | 0 | 0 |
| FILE USAGE | Determination of file names | provided by user only | 0 | 0 | 0 |
| | User identification | none | 0 | 0 | 0 |
| | User options | own signon procedure with file | | | |
| | Accounting | none | 0 | 0 | 0 |
| | Types of capabilities | read-only, read-write, etc. owner has all capabilities | | | |
| | Determination of capabilities | list of fields and capabilities stored with each user | | | 8 + variable per field |
| | Security scope | field | - | - | - |
| | Cryptography | none | 0 | 0 | 0 |
| PROTEC-TION | Separation from data | protection information in separate file | | | |
| | Integrity considerations | good -- need to use system routines; restricted entry | - | - | - |
| | Protection from concurrent systems | | | | |
| | Back-up | | | | |
| | Residual information protection | none | 0 | 0 | 0 |
| VIOLA-TIONS | Standard response | | | | |
| | None-standard responses | | | | |
| | Comments | Hsiao's Ph.D. thesis at U. of Pennsylvania | | | |

## 2.5 CPS - Time Sharing for /360

| CATEGORY | | | BYTES OF PROGRAM | DISK ACCESSES | BYTES OF DATA |
|---|---|---|---|---|---|
| | System | CPS -- time sharing for /360 | 300K | - | - |
| SIGNON SIGNOFF | Identification | account number; fixed password | | | |
| | Authentication | none | 0 | 0 | 0 |
| | User options | none | 0 | 0 | 0 |
| | Accounting | duration of use; (needed for billing anyway) | | | |
| FILE USAGE | Determination of file names | able to find all file names in system | | | |
| | User identification | six character password | | | |
| | User options | none | 0 | 0 | 0 |
| | Accounting | date last stored; date last accessed | | | |
| | Types of capabilities | read-only | | | |
| | Determination of capabilities | available through password | 0 | 0 | 0 |
| | Security scope | file | - | - | - |
| | Cryptography | none | 0 | 0 | 0 |
| PROTEC-TION | Separation from data | password kept in file directory | | | |
| | Integrity considerations | poor - open to all OS trapdoors; uses separate storage keys and designated disks | - | - | - |
| | Protection from concurrent systems | none | - | - | - |
| | Back-up | user explicitly saves files | | | |
| | Residual information protection | zeros core used | | | |
| VIOLA-TIONS | Standard response | messages only | - | - | - |
| | Non-standard responses | two signon errors result in disconnection; some errors abend CPS | - | - | - |
| | Comments | information obtained from U. of Toronto computing ctr. | | | |

## 2.6 *APL-PLUS - Time Sharing for /360*

| CATEGORY | | | BYTES OF PROGRAM | DISK ACCESSES | BYTES OF DATA |
|---|---|---|---|---|---|
| | System | APL-Plus time sharing -- /360 | 110K | - | 138K |
| SIGNON SIGNOFF | Identification | account number; changeable password | 200 | 0 | 100 |
| | Authentication | none | 0 | 0 | 0 |
| | User options | none | 0 | 0 | 0 |
| | Accounting | duration of use; (needed for billing anyway) | | | |
| FILE USAGE | Determination of file names | files to which some access is allowed are listed on request | | | |
| | User identification | password (integer) for a file; unlockable seals for functions | | | |
| | User options | able to use integer password to allow own checking fcn. | | | |
| | Accounting | date and time last stored; who stored it; amount of storage used | | | |
| | Types of capabilities | read-only, append, read-write, etc. | 300 | 1 | 98 |
| | Determination of capabilities | matrix of user numbers vs. capabilities stored with file | | 1 | 75 per file |
| | Security scope | file | - | - | - |
| | Cryptography | mnemonic passwords encrypted | | | |
| PROTECTION | Separation from data | access matrix separated from file; entry in core also separated | | | |
| | Integrity considerations | good -- no remote job entry; there exist batch programs to examine and alter files | - | - | - |
| | Protection from concurrent systems | good -- uses DOS protect features | - | - | - |
| | Back-up | every file update written directly onto disk file | - | - | - |
| | Residual information protection | nothing on disk readable before writing; | | | |
| VIOLATIONS | Standard response | error in function gives msg. and halts; file error logged | - | - | - |
| | Non-standard responses | a particular error in array subscripting disconnects user and makes workspace open to system | 130 | - | - |
| | Comments | information obtained from I.P. Sharp Assoc., Ltd. | | | |

## 2.7 *Time Sharing for PDP/10*

| CATEGORY | | | BYTES OF PROGRAM | DISK ACCESSES | BYTES OF DATA |
|---|---|---|---|---|---|
| | System | PDP/10 time sharing | 20K | - | 1K per 4 users |
| SIGNON SIGNOFF | Identification | project number; programmer number; fixed password | | | |
| | Authentication | terminal number can be read but it's not used | | | |
| | User options | none | 0 | 0 | 0 |
| | Accounting | duration of use; (needed for billing anyway); core usage | | | |
| FILE USAGE | Determination of file names | able to find all file names in system | | | |
| | User identification | password | | | |
| | User options | none | 0 | 0 | 0 |
| | Accounting | date last used; date and time created | | | 8 |
| | Types of capabilities | read-only, execute, append, write, etc.; by programmer number, project number, other. | | 1 | 8 |
| | Determination of capabilities | list of user passwords stored with file | | 1 | |
| | Security scope | file (field added at U of W. Ont.) | - | - | - |
| | **Cryptography** | **none** | - | - | - |
| PROTEC-TION | Separation from data | none | - | - | - |
| | Integrity considerations | good - nobody able to get into EXEC state | - | - | - |
| | Protection from concurrent systems | | | | |
| | Back-up | user explicitly saves files | - | - | - |
| | Residual information protection | core zeroed; deleted file erased on disk | | | |
| VIOLA-TIONS | Standard response | message only | - | - | - |
| | Non-standard responses | none | - | - | - |
| | Comments | information obtained from University of W. Ontario | | | |

## 2.8 *MTS - Time Sharing for /360*

| CATEGORY | | | BYTES OF PROGRAM | DISK ACCESSES | BYTES OF DATA |
|---|---|---|---|---|---|
| | System | MTS time sharing for /360 | 200K | - | - |
| SIGNON SIGNOFF | Identification | account number; changeable password | | | |
| | Authentication | none | 0 | 0 | 0 |
| | User options | none | 0 | 0 | 0 |
| | Accounting | duration of use (needed for billing anyway); date of last signon; resource usage | | | |
| FILE USAGE | Determination of file names | files in account listed on request; others provided by user | | | |
| | User identification | none | 0 | 0 | 0 |
| | User options | a <u>program</u> file can check the user identification | | | |
| | Accounting | date last used; location and size of file; date created | | | |
| | Type of capabilities | read-only for all shared files (including own acct) | | | |
| | Determination of capabilities | not applicable | - | - | - |
| | Security scope | file | - | - | - |
| | Cryptography | none | - | - | - |
| PROTECTION | Separation from data | kept in directory | | | |
| | Integrity considerations | poor -- privileged users can read all files and find all passwords; not usages logged | - | - | - |
| | Protection from concurrent systems | none -- other jobs under UMMPS can access all files | - | - | - |
| | Back-up | file editor uses explicit saves | | | |
| | Residual information protection | nothing on disk readable before a write | - | - | - |
| VIOLATIONS | Standard response | messages only | - | - | - |
| | Non-standard responses | three signon errors result in disconnection; some return codes abend a terminal | - | - | - |
| | Comments | new version is currently being prepared | | | |

## 2.9 *Idealized Time Sharing System*

| | CATEGORY | | BYTES OF PROGRAM | DISK ACCESSES | BYTES OF DATA |
|---|---|---|---|---|---|
| | System | idealized time sharing sys. | | | |
| SIGNON SIGNOFF | Identification | not applicable | - | - | - |
| | Authentication | not applicable | - | - | - |
| | User options | not applicable | - | - | - |
| | Accounting | not applicable | - | - | - |
| FILE USAGE | Determination of file names | | | | |
| | User identification | account number results in matched security profile | | | |
| | User options | none | 0 | 0 | 0 |
| | Accounting | none | 0 | 0 | 0 |
| | Types of capabilities | Hoffman's thesis (another ideal system) is more general and more expensive | | | 2 bits per field |
| | Determination of capabilities | group tag stored adjacent to field | 100 | 0 | 18 bits per field |
| | Security scope | field | - | - | - |
| | Cryptography | none | - | - | - |
| PROTEC- TION | Separation from data | authorization system separated; protected data separated; security profiles kept separate | 8<br>( -- | 1<br>per user | 130<br>-- ) |
| | Integrity considerations | good -- need to use system routines; bufferin of data; built - security | - | - | - |
| | Protection from concurrent systems | not applicable | - | - | - |
| | Back-up | none | - | - | - |
| | Residual information protection | none | - | - | - |
| VIOLA-TIONS | Standard response | logging of security violations | | | |
| | Non-standard responses | | | | |
| | Comments | proposed by Friedman, IBM Sys J #4, 1970 | | | |

## 2.10 CP - Operating System for /360

| CATEGORY | | | BYTES OF PROGRAM | DISK ACCESSES | BYTES OF DATA |
|---|---|---|---|---|---|
| | System | CP operating system for /360 | 100K | - | |
| SIGNON SIGNOFF | Identification | account number; fixed password | 160 | 1 | 8 |
| | Authenticiation | none | 0 | 0 | 0 |
| | User options | changeable signon procedure | 40 | 1 to 2 | 0 |
| | Accounting | duration of use (needed for billing anyway); user identification | 20 | ½ number of valid users | 12 |
| FILE USAGE | Determination of file names | files in account listed on request | 4000 | 0 | 32 per file |
| | User identification | none | 0 | 0 | 0 |
| | User options | none | 0 | 0 | 0 |
| | Accounting | date last stored | 20 | 0 | 4 |
| | Types of capabilities | all capabilities on own "machine"; read-only on others | | | |
| | Determination of capabilities | not applicable | - | - | - |
| | Security scope | file | - | - | - |
| | Cryptography | none | - | - | - |
| PROTEC- TION | Separation from data | | | | |
| | Integrity considerations | good -- no user can access CP core (only "virtual") | - | - | - |
| | Protection from concurrent systems | not applicable | - | - | - |
| | Back-up | user explicitly saves files | | 1 per 800 bytes | |
| | Residual information protection | core zeroed at signon; virtual "mini-disks" have volume labels erased at signoff | | | |
| VIOLA- TIONS | Standard response | messages only; some errors will deadlock a "machine" | - | - | - |
| | Non-standard responses | three signon errors resulted in disconnection; some severe line errors cause CP abend | - | - | - |
| | Comments | information obtained from Brown University computing ctr. | | | |

## 2.11 *OS - Operating System for /360*

| CATEGORY | | | BYTES OF PROGRAM | DISK ACCESSES | BYTES OF DATA |
|---|---|---|---|---|---|
| | System | OS -- operation system /360 | | | |
| SIGNON SIGNOFF | Identification | account number; initials | | | |
| | Authentication | none | 0 | 0 | 0 |
| | User options | none | 0 | 0 | 0 |
| | Accounting | installation dependent | - | - | - |
| FILE USAGE | Determination of file names | able to list directory of all files on a disk pack | | | |
| | User identification | password | | | |
| | User options | none | 0 | 0 | 0 |
| | Accounting | date created; size of file | | | |
| | Types of capabilities | read-only | | | |
| | Determination of capabilities | available through password | 0 | 0 | 0 |
| | Security scope | file | - | - | - |
| | Cryptography | none | 0 | 0 | 0 |
| PROTEC-TION | Separation from data | stored in file directory | - | - | - |
| | Integrity considerations | very poor - able to enter privileged state; several debugging tools; trapdoors | - | - | - |
| | Protection from concurrent systems | not applicable | - | - | - |
| | Back-up | none | 0 | 0 | 0 |
| | Residual information protection | none | 0 | 0 | 0 |
| VIOLA-TIONS | Standard response | end of jobstep | - | - | - |
| | Non-standard responses | possible to trap error and apply own corrections | - | - | - |

Comments

## 2.12 Experimentation with Security Measures

As a specific attempt to assess the cost of security measures, four simple security features were programmed in assembly language in a form suitable to being incorporated into an existing system. In this way an estimate of the amount of time and storage space added to the system was obtained.

The paging subsystem of the Data Structures Programming System (DSPS: SHARE Program Library Agency #360D-06.8.003) was chosen as the base system, as it is written in assembly language and provides a convenient environment for including security measures. The DSPS paging system allows for variable length pages to be treated as single logical and physical units. (Denning refers to this as "unpaged segmentation" in his article "Virtual Memory" in the Surveys ACM, June 1969). Items within pages are referenced by pointers which contain the secondary store address of the start of the page, the total length of the page, and the displacement of the item within the page. Each page has a header which is composed of six words: the links of a two-way list linking all pages currently in core, the secondary store address and length of the page, and the start of the free list for the page, among other bookkeeping information (see figure 1).

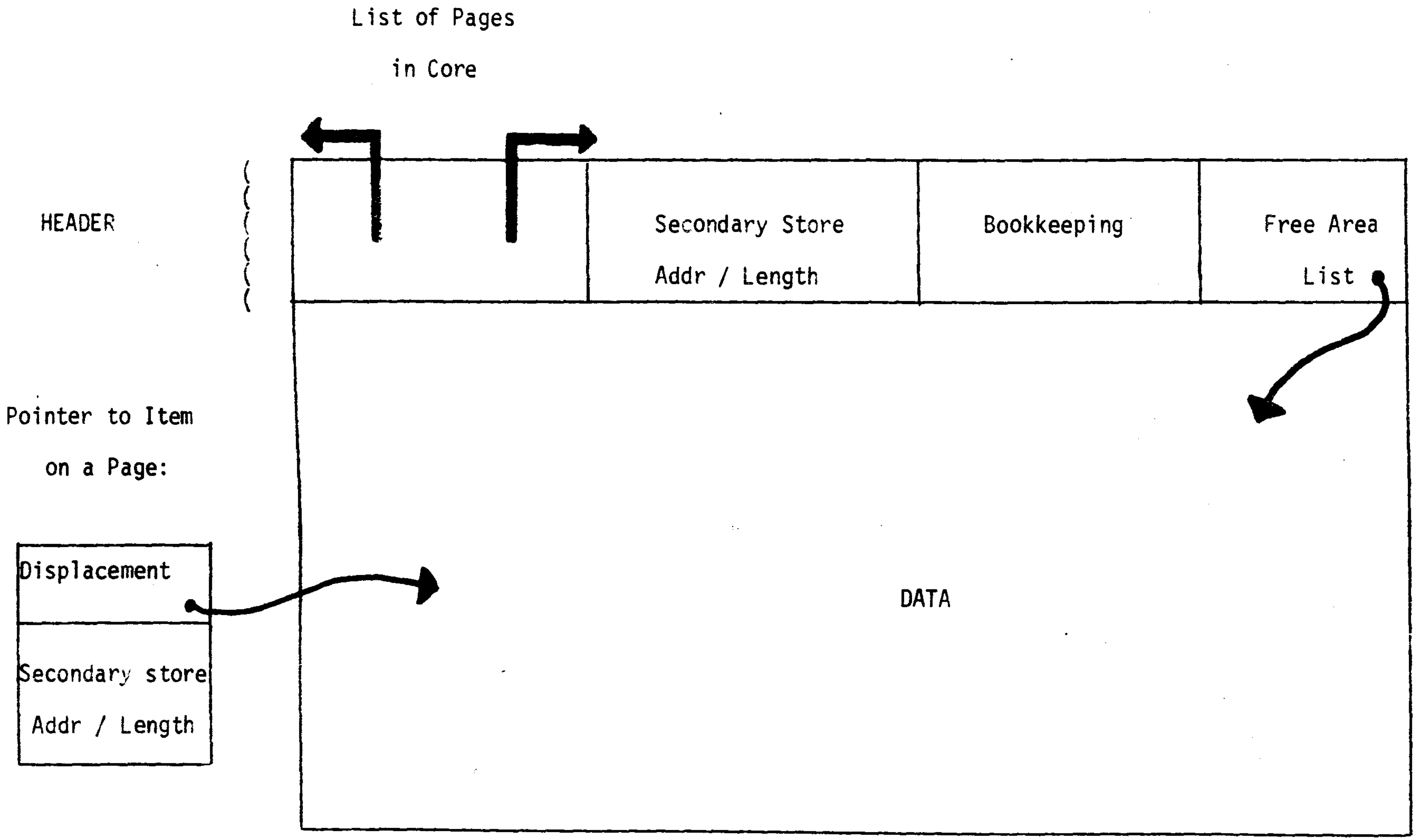


Figure 1. A DSPS page

Measure 1 Separation of the header from data

With the existing page format, any user can accidentally (or intentionally) alter the contents of his page header. In some cases this may destroy the linked list of pages, which would abnormally terminate the paging system. In other cases it may alter the secondary store address, which would result in the page being rewritten incorrectly (possibly destroying the permanent copies of some other users' pages).

As a first step towards a more secure system, the header was separated from the data itself. This allows it to be located in a separate (protected) region of core where no user can overwrite it.

When the page is read in from secondary store, the header is copied into the protected region and a pointer is established from that header to the data. The header area which is adjacent to the page must then be overwritten with hash in order to prevent its unauthorized reading. The paging system may access an item on a given page by first chaining through the headers in the protected region until the appropriate header is reached, and then using the pre-established pointer to obtain the address of the start of the data area from which the item is displaced.

Before a page is swapped to secondary store, the page header is recopied into the header area adjacent to the page. The single unit is then written out, and both regions of core which were occupied are returned to the available space list.

Measure 2 Elimination of residual information

In order to make the system more secure against browsing, the residual information left in core and on secondary store by deallocated pages must be overwritten with hash. Residual information occurs within the DSPS paging system at three distinct times: when a page is newly created, each time it is swapped out of core, and when it is deleted.

As a result, the data area of a newly created page is always overwritten before releasing it to a user. In addition, both the header and the data areas of core are zeroed after each time that a page is swapped to secondary store (i.e., before the areas are returned to the available space list). Finally, all primary and secondary store areas used by a page are overwritten when the page is deleted by a user. Thus there is never any residual information in the system available to an unauthorized user.

Measure 3 Encryption of data

In order to discourage active browsing further and to minimize the probability of any security compromise through such techniques as wire-tapping, stealing secondary store packs, etc., it was felt that data encryption was necessary. However, one problem with encrypted data is that performing simple arithmetic operations is very expensive, in general, as it involves first decoding the data item, then performing the manipulation, and finally encoding the result. Since most operating systems provide separate (protected) regions of core for each user, it was felt that the data need not be encrypted while actually in core, but only when on I/O channels and in secondary store.

An "infinite key" transformation was applied using a multiplicative pseudo-random number generator and logically adding (through an "exclusive or") the resulting string with the data. Infinite key encryption is much more difficult to decipher than any short, fixed-key transformation. Furthermore the overhead involved in random number generation is not as great as it may seem. Since each page is only decrypted and encrypted when it is swapped in and out of core respectively, and since the computation time is small compared to I/O time, a page's encryption can be overlapped with the "seek" operation which precedes page I/O.

In a fully implemented security system, each authorized user could be assigned a starting seed for the random number generator. This seed would probably be stored in the user's directory along with other authorization information. For the purposes of this experiment, though, it was adequate to assign a seed for starting the random number generator to each page at its creation and to keep that seed in the page header for regeneration of the random string for later decoding. (Needless to say, this implementation is not actually secure, as the key is provided along with the page itself).

Measure 4 Checksums for data

In order to check whether the contents of a page have been altered without authorization, a form of a "checksum" was implemented. Since the check was intended as a security measure against active infiltration as well as the conventional use against machine errors, merely adding together all the words on a page is not enough as the infiltrator need only include the additive inverse to keep the sum unchanged. To compensate for this, while calculating the checksum, each partial sum is shifted by a random amount (depending on the same random number seed used for the encryption) before adding the next word. This solution also prevents the infiltrator from rearranging the words on the page, as the sum is very order-dependent. Naturally, the sum is stored in the header of the page to prevent its unauthorized reading.

Cost summaries

Table 1 shows a measure of the cost of these simple security techniques.

TABLE 1

| SECURITY MEASURE | BYTES OF PROGRAM | BYTES OF DATA | DISK ACCESSES |
|---|---|---|---|
| Separation of page header | 212 | 24 per page | 0 |
| Elimination of residual info. | 212 | 6 | 0 |
| Encryption of data | 226 | 8 + 4 per page | 0 |
| Checksums for data | 308 | 4 per page | 0 |
| Total cost increase | 958 bytes (33%) | 4 bytes (2%) +<br>32 bytes per page | 0 |

Although this size of the DSPS paging system was increased by 33%, most of the coding involved was localized to independent subroutines. That is, no matter what the size of the original program, the security measures would require approximately an additional 1000 bytes, thus lowering the percentage increase for large programs.

Increasing the amount of storage required for each page by 32 bytes is almost negligible in most cases. For a page containing 1000 bytes of data and a 24 byte header, this increase is only 3%.

It should be noted that the execution time of a program will not be increased on account of additional disk accesses. In fact, in a short test program which did almost no computations but rather exercised the security features almost exclusively, the CPU time was increased by only 12% due to the additional bytes of program.

In general the additional costs might be larger than 12% since one would often have to add other security features beside the four specific measures that we have incorporated, for example, field level protection.

Part III

## Assessment and Recommendations

### 3.1 Introduction

There are two extreme attitudes to security of data in automated systems. One approach, usually taken in a highly-sensitive installation, is to guard the entire computer system, permit only persons with security clearance near the hardware, and use encryption whenever transmission of data is required outside the secured area. The other extreme is found in a time-shared computer with many remote terminals, (often available on a dial-up basis) where the access rights to files may be assigned by one user to another and where security measures such as password protection are available, but not mandatory.

Both of these extremes really dodge the problem of protection that exists where data that is not sensitive and data that is sensitive exist in the same system. It is not possible, on the one hand, to clear all personnel and, on the other, to permit a laissez-faire attitude by the users and expect the system to maintain protection of the sensitive data.

There needs to be some consistency of behaviour for the protection of sensitive data. To limit the access to personal data to those who "need to know" requires some regulation with the computer system and perhaps for all systems storing such information.

## 3.2 Protection by Passwords

There is a certain air of adventure that surrounds the word "password". But we are all familiar with the phrase "Joe-sent-me" as a door opener and realize how easily passwords may be passed around. The mechanism of a password is nevertheless effective, if treated with care. Passwords must be stored in two places, with the user and with the system. In each place they are subject to disclosure. The integrity of the system's copy depends on the integrity of the operating system. The integrity of the user's copy depends on his integrity and that of his storage location. Passwords may be intercepted during transmission so that a system of one-time passwords has been suggested. Each time the system requests the password from the user the next word on a pre-defined list is offered. Intercepting one password does not then provide the intruder with the next. The user of one-time passwords in computer systems is rare, perhaps because the intergrity of transmission lines is rarely questioned or perhaps because passwords are not taken seriously. According to IBM officials interviewed, most users claim it would be too much bother to keep track of the list.

The resistance to security measures is no doubt related to a general belief in the inherent honesty of other users. As Carroll remarks: "The security of Canadian corporate, institutional, and some government information systems in general is minimal. The only thing that stands in the way of substantial loss is the essential character of the Canadian citizen - peace-loving, law-abiding, honest and upright."

All computer systems require a user to have an authorization (account) number to which charges can be assigned. This number is not adequate as a password. The reasons for having a password in addition to an account number are:

1) the account number must be printed for accounting purposes and for identification of output,

2) it would be difficult to change an account number if it had been compromised. Also different privileges can accrue to different users of same account number. Passwords should be able to be changed as often as wanted by the user.

Passwords can serve for authentication of a user's identity or can be used to authenticate the authorization of a user to access a file. The password for a file might be stored in the table of access rights alongside the user's account number or it might be part of the label stored with the file. It is likely that password-protection of a file is really not of much use if the file is open to several users (on a limited-access basis). If the identification of individual users has been done properly the users identification in the access rights table of a file should be sufficient. The password on a file serves as a second line of defence against masquerading.

Passwords are most commonly used in interactive systems and are a useful deterrent to snoopers. However, few people believe that sensitive data is safe in such an environment. For example, in one large Canadian time-shared installation all files are kept off-line rather than on-line with password protection. Data that is waiting in the core upon receipt from or before transmission to the customer is encrypted using a simple transformation. This latter was adopted to prevent accidental rather than deliberate invasion of privacy. The cost of password protection is very small and cost is certainly not the reason why it is not universal. It would seem to be a must in normal practice with sensitive data both for user authentication and file access. It undoubtedly must play an important part in preventing the compromise of data.

## 3.3 Protection by Encryption

Devices now exist, and have been used for a long time for diplomatic messages, for transforming a string of characters into another (encoded) string where the key to the transformation is generated by the device from a starting key. The starting key is set up on buttons or wheels that are kept under lock and key (in the ordinary sense). When data is to be encrypted the starting key is set by a security officer if it has not been permanently fixed and transmission begun. At the other end the encrypted data passes through a similar device, or a computer, that has had the same starting key set on it. The device is really a special purpose computer and the operation of it could be programmed on a mini-computer (cost about $8,000) if desired. The mini-computer would then be kept secure and under the supervision of a system security officer. The computer function at the central end of the transmission could be done by the main computer itself so that there would be no need to have a separate computer at headquarters, only at the remote stations.

It is possible to arrange such devices to encode the input string or not according to special control characters in the string itself. For instance in a personal file the identifying fields might be encrypted, the numerical fields not. Global data processing operations could then take place on the file without decoding it at all as the fields that would have arithmetic operations performed on them would be in the clear. No one reading the file as stored in the computer could make any sense of it. The decoding would occur when it was returned to the remote station. In this case the remote terminal might be a card-reader/printer kept secure by a personnel department. The Datacoder model DC-110 by Datotek Inc. is a device that works in this manner.

The view has been expressed that all sensitive files should be kept entirely in encrypted form except when actual operations are occurring. However, the trouble with complete encryption of files is that either the whole file must be read to change a single field to synchronize the generation of the long key, or a short key would have to be used. Short keys are really very effective even though, like passwords, they can be compromised. Again a simple device, used, is much better than none at all.

## 3.4 Limited-Access Control

An operating system must be able to ensure the integrity of private files and limited-access files.

If a particular file is password-protected each user who has access rights must know the password and be aware when it is changed. The problem of calling for a change then arises. Unless there is one user of the file declared as the owner there will be thequestion of who should take the initiative. In many ways it would be better for one user to have the responsibility of the security of the file and another to either act as his delegate or only access the file through a file reader program belonging to the prime user.

The file would then be private to the file reader program and all the security measures due to the sharing of the file would be in the file reader program rather than in the operating system program. The cost of going through an intermediate program for limited-access information has been studied by Lance Hoffman who suggests that access handling programs be modular so that cost-effectiveness of various components can be estimated.

All persons to whom the privilege of accessing a file is extended must be understood to be capable of passing it along, if not within the machine environment, in hard copy form. Here a personnel security problem exists. It must be made plain that it is against the regulations for someone who is not the owner of a file to pass it to another person. The information should be classified as "limited-access".

The problem of allowing access to certain fields of a file, and not to others, is again something, we believe, best handled by a file reader program where the file and the file reader program are private to the file security officer. The extent of access rights of others are his responsibility and he controls the file reader program. This might mean that there is less economy in file amalgamation if two private files become one field limited-access file but the actual costs would depend on the specific system.

## 3.5 Audit Logs

All computer systems have accounting systems that keep a log of all events significant to charging for computer services. These logs do not always record events that have no charge generation function. Since security cannot be absolute even though operating systems improve enormously, audit trails are essential to detecting security violations. A straight log is often very difficult and time-consuming to interpret; the analysis should be done by the computer system. If the analysis of suspicious events, such as incorrect passwords or attempts to read beyond the assigned range of core addresses, is done as the events take place, an alarm can be set off. There are various levels of severity of action on alarm from expelling a user or shutting down a transmission line to locking all files. Some systems have modest alarm schemes and these should be standard with sensitive data. Data that is highly sensitive might warrant more extreme action. The audit log can be sorted to reveal any particular kind of peculiarity and should be a standard tool available to a system security officer. The cost of an audit log and reasonable alarms would involve some dedicated secondary storage and computer time for analysis of the record.

### 3.6 Physical Security

No secure computer hardware should be open to the public. It should all be limited-access. This immediately cuts down the problem of infiltration for privacy violation or destruction.

The methods of access control to physical facilities through password, badge, key, combination and human guards are well established and will not be laboured here. Site protection from external hazard or attack must be considered but is not really a problem with regard to privacy.

The destruction of data when no longer needed is standard for hard copy but is often neglected with data on tapes or in core store. Since the recording operation pre-erases the previous recording it is custom to leave tapes and core with whatever information was last recorded on them. If the data is sensitive it should be erased by the user by writing meaningless strings of characters (hash) into core store. Tapes and disks are more difficult to erase and might require several writings of hash to remove residual information.

## 3.7 Personnel Security

As with physical security the problems of personnel security are a part of any information system, automated or not. It is perhaps true that fewer people "need to know" information in the automated system and thus the probability of a leak is not as great. On the other hand, vast quantities of data may be leaked by one person in a very short time due to the speed of the system. The establishing of trustworthiness is something that persons in charge of security should control.

Having established that certain persons are to have access to files, the system must be sure that impersonation is not possible. When requests come by telephone from a person, it should be standard to hang up and call the person back at the number listed opposite his name in the directory. Unique identification should be a goal; finger print or voice print devices are not perfected yet and may not be the answer since even these are reduced to digital form for transmission and could be duplicated. Personal identification is possible when the user can be seen by another authorized person and perhaps "two-person" access to sensitive data should be employed. Passwords, badges, or keys remain the common means of authenticating identity but are easily used by someone else.

## Recommendations

The following is a list of recommendations found in this report.

1) Confidential data should be subject to data security provisions at all times until it is erased or destroyed.

2) Where highly sensitive data is read, displayed, or printed, hardware input-output devices should be adequately shielded both from visual observation and eavesdropping on stray electromagnetic radiation.

3) Confidential files maintained off-line in computer readable form should be maintained in a careful banking system under strict control.

4) Confidential files maintained on-line in a time shared computer system should be:

   a) password protected,

   b) have some form of encryption that at least prevents accidental disclosure. As a minimum, a short key transformation on identifying fields of its records should be standard.

5) A file reader program of an operating system should be used for access to all files that are to be made accessible on a field-limited basis. Where such a file reader is not provided, files should be available or not to a user only as a complete unit.

6) All user programs that deal with confidential data should ensure that any copies that are made are properly erased after use.

7) Operating systems should be constructed as far as possible to be inaccessible to users. This is now difficult to achieve and may be for some time the weakest link in the security chain within the computer system.

8) Tables of access rights should be under control of the operating system and not directly accessible to users.

9) During transmission of sensitive data, encryption should be sufficient at least to prevent accidental disclosure if the data is intercepted.

10) The more sensitive the data the more complex should be the encryption for transmission purposes.

11) Access to rooms containing computers that handle confidential data should be carefully controlled.

12) The identity of users requesting confidential data from remote terminals should be authenticated, for example, by a callback procedure.

13) An audit log should be kept of all unsuccessful attempts to obtain data from the system and all occasions when limited-access data is obtained by a user.

14) Standard precautions should be taken with regard to personnel having access to confidential data in the computer system and to auxiliary storage of data out of the computer system. Physical security and personnel security are by far the most vulnerable parts of any information system.

Appendix I

## QUOTATIONS ON COST ESTIMATES FOR DATA SECURITY

"Right of privacy and medical computing" Dr. E.R. Gabrieli DATAMATION, April '70, p. 173

Wm. Holmes, director of Comp. Sci. Div. of Cornell Aeronautical Lab., Cheektowaga, N.Y.

"The cost of securing adequate privacy includes the privacy administration, programming, incremental cost in computer memory, incremental running costs due to privacy programs, and 'nuisance cost' for identification of users. It also should include a one-time incremental storage cost of $10,000-20,000, about 10% of additional programming, and a 2-10% increase in running cost of programs providing the privacy features."

Anthony L. Mondello, general councillor of the Civil Service Commission.

"Data security at the computer level represents a 5-10% cost increase, covering both hardware and software expenses."

"Maintaining confidentiality of data in educational research: a systematic analysis" R.F. Boruch to be publ. in Amer. Psychologist

"For example, in one case Hoffman (1970) estimated a cost of 8¢ per record for scrambling card-image data, a device which eliminates the possibility of accidental disclosure, and, to the extent that data cannot be translated, protects against deliberate tampering."

"Safeguarding time-sharing privacy - an all-out war on data snooping", anon. ELECTRONICS, April 17, '67

of Walter Bower, pres. of Data Products Corp's, subsidiary Informatics, Inc.

"He anticipates a day when 10% of a computer's memory will be devoted to routines needed to qualify users requesting information. In some applications, according to Bower, the figure could be as high as 20%."

"Data Security in the CDB", anon. <u>EDP Analyzer</u>, May '70

C. Weissman, SDC, "Weissman estimates that the security portions of ADEPT-50 required about 5% of the total design time (in man-hours) and about 10% of the coding. About 80% of the code in the security portions is local to just five components of the total system. About 2% of the CPU time is spent in performing security checks."

"Authorization problem in shared files" T.D. Friedman, <u>IBM Sys. J.</u> 4, '70

- "If we assume that the average entry is 50 bytes (400 bits) it follows that the proportion of information in the secured shared file devoted to protection is 18 bits divided by 400 bits, or 4.5 per cent of the total file."
- "Since the average user is assumed to hold privilege for 200 groups, storage for an average profile is 120 + 200(18+2) + 18 (or 4138) bits. Since 20,000 users are recognized 82,760,000 bits or 10,345,000 bytes of storage are required for a complete set of profiles."
- "The authorization system will impose a delay which is expected to be small in comparison with the file search delay."

"Security techniques for EDP of multilevel classified information" H.W. Bingham, Burroughs Corp., 4424-65-112, 2 October 1965

"For modular multiprogramming multiprocessing systems of apparent future development, the hardware techniques suggested for security protection represent about a 10 per cent increase in EDP hardware over that necessary for the basic processing task performed in a multiprogramming multiprocessing system with on-line users. A corresponding small increase in memory is required for security routines and tables. The individual execution time for security routines is small compared to the ECP or service routines within which they are imbedded."

"Fast 'infinite-key' privacy transformation for resource-sharing systems" J.M. Carroll & P.M. McLelland, FJCC '70

> "The test of processing speed was carried out in the time-sharing environment of the PDP-10/50 system under full load with 'swap' times included. It was found that 135,168 five-byte words were produced in 18.28 seconds (37,000 bytes per second) using the high-security (slower) procedure. It is felt that this test represents worst case conditions and that double the observed speed can easily be realized."

Appendix II

## GLOSSARY OF TERMS

Adapted from "IFIP-ICC Vocabulary of Information Processing", North-Holland Publishing Company, 1966.

FILE
A collection of DATA complete, in some sense, for the purpose of a particular job. For example, in stock control a file could consist of the complete set of invoices for a given period.

RECORD
A file may be considered, as composed of a number of RECORDS, each record containing the data relating to one particular part of a job. In the stock control example, each invoice could constitute one record.

FIELD
A record may be further sub-divided into FIELDS, each field being the smallest quantity of data considered as an entity for the purpose of the job. There is a hierarchy FIELD∈RECORD∈FILE.

RECORD LAYOUT
The arrangement both as regards sequence, and lengths in characters or machine words, of the fields in a record.

KEY
A field of a record used to identify that record in a particular file organization.
or A device that will open a LOCK.

LOCK
A device that prevents access by anyone not possessing a key to some location. In computers, the locations are places where data can be stored called storage locations.

PASSWORD
A string of characters that is used as a lock and key device. To unlock the lock the user must present, as a key, the same string as is stored in the computer as the lock. Keys must match locks to work.

ACCESS RIGHTS
Privileges held by a user with regard to particular items of data, that is, files, records, or fields. The rights are usually read only; read and write; read, write, and change the protection (for example, password).

MATRIX OF ACCESS RIGHTS
A table each row of which corresponds to a particular user and each column of which corresponds to a particular access right. Each item of data must have such a table when access is limited to particular users with particular privileges. The entries in the table are simply yes or no and could be stored as one binary digit for efficiency.

AUTHORIZATION
Permission from a valid authority to perform certain actions.

AUTHENTICATION
Validation of the claimed identity of a user by some checking mechanism.

## References

1 Baran, Paul "On Distributed Communications IX Security, Secrecy, and Tamper-Free Considerations" Contract No. AF19(638)-700, The Rand Corp., 1964.

2 Bingham, H.W. "Security Techniques for EDP of Multilevel Classified Information" Contract No. AF30(602)-3596, Burroughs Corp., Oct. 1965.

3 Carroll, John M. and Philip M. McLellan "The Data Security Environment of Canadian Resource-Sharing Systems" INFOR, Vol. 9, No. 1, March 1971.

4 Friedman, T.D. "The Authorization Problem in Shared Files" IBM Systems Journal, Vol. 4, No. 4, 1970.

5 Hoffman, Lance J. "Computers and Privacy: A Survey" Computing Reviews, Vol. 1, No. 2, June 1969.

6 Hoffman, Lance J. "The Formulary Model for Access Control and Privacy in Computer Systems" SLAC Report No. 117, Stanford University, May 1970.

7 Katzenbach, N. de B. Statement before Subcommittee on Constitutional Rights of the Committee on the Judiciary, United State Senate, March 1971.

8 Peterson, H.E. and R. Turn "System Implications of Information Privacy" AFIPS Proc of Spring Joint Computer Conference, 1967, pp. 291-300.

9 "Computer Security" Industrial Security, Dec. 1969.

10) "The Considerations of Data Security in a Computer Environment" IBM.

11 "Data Security in the Corporate Data Base" EDP Analyser, May 1970.

12 "Safeguarding Time-Sharing Privacy" Electronics, April 1967.

13 "Security Checklist" DCF Systems Ltd., Toronto.

14 Official Publication
"Security of Data Processing Systems - An Interim Policy" Security Services Branch, Department of Supply and Services, May 1971 (restricted).

15 Carroll, John M. "Personal Records: Procedures, Practices, and Problems" Study for the Privacy and Computers Task Force.

## STUDIES COMMISSIONED BY THE TASK FORCE

The Nature of Privacy - D.N. Weisstub and C.C. Gotlieb.

Personal Records: Procedures, Practices, and Problems - J.M. Carroll and J. Baudot, Carol Kirsh, J.I. Williams.

Electronic Banking Systems and Their Effects on Privacy - H.S. Gellman.

Technological Review of Computer/Communications.[1]

Systems Capacity for Data Security - C.C. Gotlieb and J.N.P. Hume.

Statistical Data Banks and Their Effects on Privacy - H.S. Gellman.

Legal Protection of Privacy - J.S. Williams.

Vie Privée et Ordinateur Dans le Droit de la Province du Québec - J. Boucher.

Regulation of Federal Data Banks - K. Katz.

Regulatory Models - J.M. Sharp.

Ordinateur et Vie Privée: Techniques et Contrôle - C. Fabien.

The Theory and Practice of Self-Regulation - S.J. Usprich.

Privacy, Computer Data Banks, Communications and the Constitution - F.J.E. Jordan.

International Factors - C. Dalfen.

[1] A joint Study by the Privacy and Computers Task Force and the Canadian Computer/Communications Task Force, to be published by the latter.