PRIVACY AND SECURITY IN DATA BANKS

By

William A. Garrison and C. V. Ramamoorthy
Department of Electrical Engineering
and Computer Sciences

Technical Memorandum No. 24
November 2, 1970

INFORMATION SYSTEMS RESEARCH LABORATORY

ELECTRONICS RESEARCH CENTER
THE UNIVERSITY OF TEXAS AT AUSTIN
Austin, Texas 78712

PRIVACY AND SECURITY IN DATA BANKS

By

William A. Garrison and C. V. Ramamoorthy
Department of Electrical Engineering
and Computer Sciences

Technical Memorandum No. 24
November 2, 1970

INFORMATION SYSTEMS RESEARCH LABORATORY

ELECTRONICS RESEARCH CENTER
THE UNIVERSITY OF TEXAS AT AUSTIN
Austin, Texas 78712

*Research sponsored in part by the Joint Services Electronics Program under Research Grant AFOSR 69-1792.

ABSTRACT

The problems and implications of privacy in today's computer oriented society are many and diverse. The right to privacy is not a formal right guaranteed by the Bill of Rights. How are we to insure that privacy will not become a non-existant commodity? Will the creation of large data banks containing personal information result in "automated blackmail"? Will the end result be a police type dossier on every citizen in the country?

Will the day come when we will evolve into "cashless" society, where all financial transactions are recorded by a computer? How can we accrue the benefits that a large data bank can bring about without the fear that "Big Brother" is watching us.

Some of the possible solutions both legal and technical are discussed along with some current schemes being employed to insure the privacy and security of information. Additionally, some proposals are discussed which might be used to guarantee file integrity. The application of cryptography to the security problem is also discussed. Classification of the various levels of protection is made with suggested environments in which they might be applicable. A suggestion is made as to how overall system performance might be monitored as a result of implementing high level security and auditing routines.

TABLE OF CONTENTS

Privacy and Security in the Data Bank

LIST OF FIGURES

PART I

INTRODUCTION

1.1 The Concept of Privacy in Today's Society:

The concept of privacy is a sadly neglected value in our society. It was ill defined and rarely discussed until modern times; and while not a new concept it is often treated as such by commentators and social scientists. Our belief in privacy has developed from a tradition of limiting the surveillance power of authorities over most personal and group activities. Alan F. Westin [15] defines privacy as the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is to be communicated to others. Every person at one time or another desires to temporarily withdraw from society whether physically or psychologically.

Specialized disciplines such as law, psychology, or political science undoubtedly have different interpretations as to what the essential concepts of the diminishing concept should be. Professor Alan F. Westin in analyzing privacy in an individual's life recognizes four significant functions that privacy performs: (1) Personal autonomy (2) emotional release, (3) self evaluation; and (4) protected communication. Privacy is vital to an individual's way of life and is essential to his psychological well being and therefore a basic human right.

Despite the importance of privacy, the law has been reluctant to grant privacy the same status as other protected rights such as freedom of speech, freedom of the press, and etc. This is no doubt due to the fact that in any political regime certain patterns of privacy and surveillance are necessary for the survival of the social order. [15] This becomes very clear when one examines the concept of privacy in a modern totalitarian state such as communism or facism and compares it to the concept of privacy in a democratic or republican government. A totalitarian state requires absolute secrecy for itself, but maximum disclosure for its subjects in order to exercise control. In a democracy privacy is not required to be an absolute right. Since democracy requires participation by its members at least some of the time, nonparticipation or some private acts can endanger the whole society. Unreasonable privacy may threaten internal security unless lines are drawn. The problem of effective police controls over crime result in a conflict of the rights of the society versus the individual. Thus a democratic state is always searching for checks and balances of private versus public interests while in a totalitarian state private interests are usurped as privacy is attacked as "immoral", "antisocial" and "part of the cult of individualism".

As a result of this reluctance to grant privacy a special status, no United States or English court even ruled on the issue of privacy until after 1890. Some progress has been made in recognition of privacy's

importance as twenty states and the District of Columbia now recognize privacy as cause for a civil suit.

The present status of privacy as a right is unclear. The Supreme Court has recognized the duty to protect against invasions into "constitutionally protected areas" and has refused to permit evidence seized in contravention of the Constitution but, has so far refused to incorporate a right of privacy into a Bill of Rights.

1.2 Why are People Worried? The Data Explosion.

The age of computers has given rise to an enormous capability to collect, collate, classify and process data about anything and everybody. While many of the records available are not new, such as birth certificates, security clearances, tax roles, employment records, and credit records, they have taken on a new and menacing character. Contemporary American society has always been organized and record conscious, and with generally beneficial results. However, with the computer, man has created the capability to collect, examine and analyze records concerning an individual in seconds that would previously have taken a full time investigator months or years to collect. Thus while the ability to collect information is not new the computer has put the spotlight on millions of people who never would have attracted anything more than a cursory glance by a private or governmental agency. Nevertheless, there are potential advantages to the use of the computerized data bank. Thus the desirability and availability of the computer utility and data bank has started people to thinking about the

implications that readily available information on large section of the population might have. Computer utilities are here with a large number of on-line time-sharing systems currently operating. The large utilities make the advent of a data bank possible and practical.

The public data bank is thought to be inevitable, just as the railroad, power, and telephone utilities were. The reasons are that it is less expensive to pool your resources and share data rather than generate and store your own records.

Paul Baran lists some more easily followed examples on the desirability of keeping so many records, among these are:

a. Tax auditors might want to check records of associates of a man under scrutiny.

b. A company may want to check its personnel records before making a reference.

c. Veterans Administration may want to examine a man's military record to validate a claimed service connected disability.

d. A lawyer may wish to search jail records, arrest records, and credit records of all witnesses for the plaintiff.

e. Professional licensing boards may want information concerning a man's character or qualifications.

f. Military may check a man's background by perhaps checking what library books he checks out to determine his suitability for a sensitive position.

g. Medical data banks could enable a transient to get improved medical aid by allowing the physician access to his medical history.

However, there exist no guidlines for the handling and dissemination of private information. Yet the clamor is to do it now. There are obvious advantages to pooling information as instantaneously available data can improve the accuracy and speed of vital policy decisions. Such decisions are more likely to be accurate and better correlated as they are not subject to individual distortions. Some of the factors to consider are:

a. Important historical records are sometimes lost because of the absence of a consistent policy and procedure for establishing and maintaining archives.

b. The absence of appropriate standards and procedures for file maintenance and documentation leads to low quality files that contain many technical limitiations in a statistical usage.

c. Many useful records are produced as a by product of administrative or regulatory procedures by agencies that are not equipped to perform a general purpose statistical function.

d. No adequate reference exists that would allow users to determine easily whether or not records have the characteristics of quality and compatibility that are appropriate to their analytical requirements.

e. Procedures for collecting, coding and tabulating data that were appropriate when developed now lead to some incompatibility in record association and usage required by current problems and the attendant solutions made possible by computer techniques.

f. There are serious gaps in existing data records that stand in the way of bringing together records of greatest relevance for today's problems.

g. The need to bypass problems of record incompatibility in developing statistics appropriate for policy analysis places severe strains upon regulations restricting the disclosure of information about individuals. However, technical possibilities for using the computer to satisfy these statistical requirements without in any way violating personal privacy have not generally been developed and made available by the agencies.

It is obvious that input data would have to be validated before it is entered into the system. It is also true that unless legal precautions are taken an individual might not be aware of any derogatory information that is put into his dossier. It also is possible that false and slanderous data could be inputted without the individual's knowledge. Thus it is obvious that the individual must be part of any validation process.

To most people a computer printout looks quite official and irrefutable, yet computer printouts can be faked. Thus it is possible that a person's reputation could be jeopardized by a faked printout. The magnitude of information that it is technically feasible to assemble is enormous. Laser technology will probably allow a 20 page dossier to be kept on 200 million people on a single plastic tape reel. Under such conditions it might be easier to keep data rather than destroy it, thus bad or derogatory data might not get destroyed if it were at some point repudiated by authorities, invalidated, and ordered removed from all records. Thus according to Baran [1] we have a balance problem that must be resolved in any implementation of the data bank.

1.3 Legal and Administrative Safeguards:

Many people are worried about the possibility of such dossiers being kept in such a data bank and reacting strongly against the proposed Federal data center. Overreaction might prevent the system from getting off the ground. Without public trust the data banks might be fed false data by a suspicious public thereby rendering them useless. Therefore it is in everyone's interest that safeguards be built into any system and most importantly that they work.

The Federal government is more aware of the problem than most of the states with the possible exception of California which has an intergovernmental agency that solicits opinions on the problem. The big problem comes not from the machines however, but from men. The motives and ambitions of political executives and managers and careless technicians might result in the degradation of an individual's right to privacy and dignity. Points to consider are:

a. The need to study the problem while there is still time and urge the adoption of an industry code for those who design and operate the computer processes.

b. If self regulation is not imposed, government regulation will result in the computer becoming a villain instead of a friend.

c. The Washington D. C. chapter of the Association of Computing Machinery (ACM) has gone on record as opposing a data bank until it can be demonstrated that a safe system can be economically built. (This is thought to be a minority view within the chapter).

d. The right to privacy is not protected by the Constitution and is subject to interpretation of

local courts and the legislature which makes it a very inconsistently defined concept.

With these problem in mind John McCarthy suggested a computer bill of rights. Some of the proposed rights were:

1. The rules governing access to files are definite and well publicized, and the programs that will enforce these rules are open to any interested party; including for example the ACLU.
2. An individual has the right to read his own files, to challenge certain kinds of entries in his file and to impose certain restrictions on access to his file.
3. Every time someone consults an individual's file the event is recorded, together with the authorization for the access.
4. If an organization or an individual obtains access to certain information in a file by deceit, this is a crime and a civil wrong. The injured party may sue for invasion of privacy and be awarded damages.

Yet the discussion goes on, for in 1967 the U.S. proposed a Rights to Privacy Act banning wiretapping and electronic eavesdropping but, in 1968 the Safe Streets and Crime Control Bill granted authority for wiretapping and eavesdropping even without a court order for a limited time. Even if the

the government were to pass a law protecting government files, state and private files would still be a problem. Medical records are an especially touchy area. One state (California) has recognized this problem and has declared state files as public records.

1.4 Security and Privacy, Some Differences:

The idea of a right to privacy is closely tied with the traditional concept of information security but, there are important differences. A basic problem in assuring privacy is security. Security is the act of preventing unauthorized access and snooping of sensitive information. This implies the necessity of adequate safeguards built into management, and hardware/software aspects of the system. It would appear that whereas privacy is a social issue, security is a technical and management problem. As indicated previously, privacy is related to a person's personal history and confidences. In the realm of national and international politics, security would imply a link to defense information. This has led to the creation of three security classifications; Confidential, Secret, and Top Secret. As far as information vital to national defense is concerned once a classification for a piece of information is picked, a standard set of rules and procedures are outlined for protection, access, transmitting, and modification of classification or content.

Within the classifications of Confidential, Secret, and Top Secret there are various levels of authority. Security precautions are strengthened with the "need to know" doctrine which states that having the necessary

clearance or level does not automatically grant one authority to access information if that information is not necessary to the success of the objective.

Unfortunately, no such standards exist within the privacy sphere, although larger industrial complexes do have their own privacy classifications which attempt to protect information vital to their economic interests. Certainly the penetration of a large industrial complex could be costly to the company and profitable to the penetrator. The problem of industrial spying and espionage is a continuing crisis. How secure are the lines of communication a company uses? Could a company borrow a chapter from the techniques used by the military and encode their inter-office communications. If the international data banks become a reality it certainly will be important that a foreign company not learn the company secrets of a domestic company, thereby causing the domestic company to lose its ability to compete in an international market.

1.5 Possible Threats to Information Privacy:

There are a number of methods that can be used to intrude on the information stored in a computer. These can best be described by separating them into three categories:

1. Accidental--These are usually the result of user error or system error. The information accidentally received in such a case neverthcless compromises the file.

2. Deliberate, passive--This includes electromagnetic pickup, and of course wiretapping.

3. Deliberate, active--This includes browsing, masquerading as another user, "between lines" entry while user is inactive but, on channel, and "piggy back" entry by interception and transmitting an error to the user. Also, core dumping to get residual information that might possibly of importance. This would be the computer age equivalence of reading a typewriter ribbon.

One of the problems with any sort of digital communication is that many people are lured into a false sense of security because the information being transmitted is in digital form. However, all that is needed to successfully wiretap is a tape recorder and a conversion table to decode the digital patterns. It is also possible to compromise a file by imposing unnecessary safeguards on a piece of information and thereby possibly preventing vital or complimentary information from reaching a policy making board whose decision might be influenced by the availability of certain information. Lastly, but important is the integrity of the personnel who operate the computation utility such as operators, engineers, and management. It has even been suggested that computer scientists and programmers be licensed as a way of insuring their integrity. The problem, with this idea is that programmers tend to be individualists and would reject this an unnecessary regimentation.

PART II

THE FUNCTION OF THE DATA BANK

2.1 Purpose and Scope of the Data Bank:

Some confusion exists as to just what a data bank is and what its purpose is. Added to confusion is the presence of the computer utility. The data bank is viewed as being somewhat different from a computer utility. The utility provides computational power, business, scientific and filing services while the primary purpose of the data bank is the safe keeping of subscriber's files. In otherwords, the data bank is an automated library, which may or may not be tied in with a computer utility or a user's own computer system.

In the case of a library it is relatively simple matter to keep improper or sensitive information out of the hands of unauthorized personnel by denying them physical access to the critical storage area. In a data bank the objectives are similar but, the problems in achieving the desired end are greater. The processing and storing of sensitive information and preventing it from falling into the wrong hands is a technological problem requiring a comprehensive examination. Thus a decision must be made as to what is needed in the way of security. Before this decision can be made an examination of the type of information that a data bank will be required to handle must be made.

2.2 Types of Information Stored in a Data Bank

While the possible uses and the scope of services are numerous it is possible to broadly categorize the type of information to be stored in a data bank into four classes as follows:

1. Private information - to be used by individuals and families in an emergency (perhaps medical information to be relayed to a physician performing emergency medical aid and in need of the patients medical history) or in routine personal business (accounts due, income tax data, bank accounts, etc.). The information is generally not to be shared with outside parties.
2. Private information to be used and shared within a group such as confidential company data, defense information or intelligence information not to be released to anyone outside of the cognizant group.
3. Private information shared at cost to subscribers; e.g., computer utilities sharing procedures for a royalty. Information or service to be sold to the public but, stored in a public data bank. Public libraries might use the data bank as one phase of user services.
4. Shared public information available to everyone at no cost. Census, statistics, stock market quotes, government records and etc.

2.3 Types of User/Subscriber/Clients/Participants of a Data Bank

Invariably a data bank will attract a wide range of users whose intent and goals vary. The types of participants of a data bank are:

a. Individuals or groups who permit information about themselves to be stored for public as well as individual good. Thus an individual might store his medical records or history in a file to make it available in an emergency. Due to the possibly sensitive nature of this data the individuals or groups would probably insist on suggesting a classification and security controls over this information. Certainly they would wish to determine who and under what conditions legitimate parties could access this information, (Medical data, etc.).

b. Individuals who keep information for safekeeping and processing exclusively for their own use. Company policies, trade secrets, bank accounts, statements of loss and earnings are possible examples. The Department of Defense, intelligence agencies, National Security Council and others would want exclusive and absolute rights to certain data. This of course presumes, perhaps unrealistically, that a data bank can be made absolutely secure.

c. Individuals who keep information in a data bank for renting and royalty. To these the data bank serves as a secure storage facility from which computer utilities attached to the data bank rent these procedures as part of their services to their subscribers. In otherwords, the data bank serves as a storage and distribution center from which an accounting of the type and amount of usage of his files can be made so as to bill the utilities attached to the data bank who presumably would bill their customers.

d. Individuals whose apparent desire is to steal unauthorized information. They have no apparent legitimate use of the data bank in mind but, operating under the guise of a legitimate subscriber will try to obtain company secrets or private information by covert means. Obviously, allowance for the detection of this class of user must be made.

The possibly personal, sensitive, or proprietary nature of information makes security in a data bank mandatory. Consequently, operators of data banks whether public or private will be under pressure to guarantee the effectiveness of security procedures. The wide acceptance of data banks will most certainly be delayed until legislation results in the data bank being liable for damages resulting from unauthorized disclosure of a subscribers' file. This is a very cirtical area, if regulations are too strict or the conditions under which damages may be awarded capricious or arbitrary the public data bank

may never be an effective tool for society due to unnecessary harassment. On the other hand if the data banks are misused public distrust will insure their demise.

2.4 Function of a Secure Data Bank

To be a useful tool the data bank must have a certain number of capabilities and/or characteristics. Several of these functions are possibly unique to the data bank, such as information classification and detection and information validation. In a computer utility there are usually no attempts made to insure the authenticity of information inputted to the system. The validation required, is the distinction between the data bank and the computer utility. Management plays an increasingly important role in the data bank operating policy. With this in mind some of the functions of a secure data bank would be:

1. Information classification and detection - When information is received, it must be analyzed and classified according to the sensitivty of its contents. It would be assumed here that what constitutes private or sensitive information is determined by the policy set either by the government or the data bank. The principles of this policy would be used to determine the private nature of the information received. Based on the nature of the information, a security classification would be assigned.

2. Receiving validation and permission to use the information - After recognizing that the received information is sensitive the data bank would next seek the validity (the truthfulness) of the the information from the individual himself as well as request permission to use the information for private and public good. This process allows him to correct any mis-information as well as bestowing upon him the right to refuse the use of the information. The data bank should permit the individual to see his personal information anytime.

3. File update and manipulation functions-The data bank must provide for basic operations permitting the updating and changing of files stored in the system. Retrieval of files and the storage of new files must be simple and efficient.

4. Access control and protection of information - A data bank must also have the capability of specifying and controlling access of information during transmission, manipulation and static storage. Additional protection can be obtained by encryption of the data base.

5. Surveillance, threat monitoring and system recording functions - Overall system security can be considered as part of a general purpose surveillance and monitoring routine where all major components of the system are monitored including the security functions such as threat monitoring, access control, and file processing. All of these functions must of practical necessity operate under the

assumptions: (1) that operators of the system are trustworthy, (2) system personnel have some knowledge of snooping techniques, (3) cost of breaking through security is much greater than the cost of the information stored in the data bank.

A parallel but independent activity will be that of surveillance and threat monitoring function. It is a watch-dog function always "looking" at all the operations taking place at the data bank and recording significant events.

2.5 Representation of Data Flow in a Security Oriented Data Bank

The data bank model outlined in Figure 1 represents the basic structure of a security oriented system. A remote station inputs data to the central data bank site via a decoder/encoder (D/E). The D/E serves to protect information in transit. Information received at the central site through the communications modulator demodulator (modem) may be left in the encoded form by bypassing the decoder.

A security profile for new information is established by a classification phase (Figure 3). This information is used by the access control and management routines to provide necessary safeguards and protection. (Figure 4) After identification and analysis, the access control and management routines are used to authenticate, to monitor accesses, to update and modify file linkages, to create capability lists, and to control processes permitted in the data space. In some cases an extremely sensitive file may be stored in the encrypted form. The file processor is dedicated having only the ability to assign storage space and to insure its efficient utilization. The processor cannot address itself to the contents of any file and merely serves a retrieval and storage function.

Model of a Data Bank.

(a) Equipment Configuration

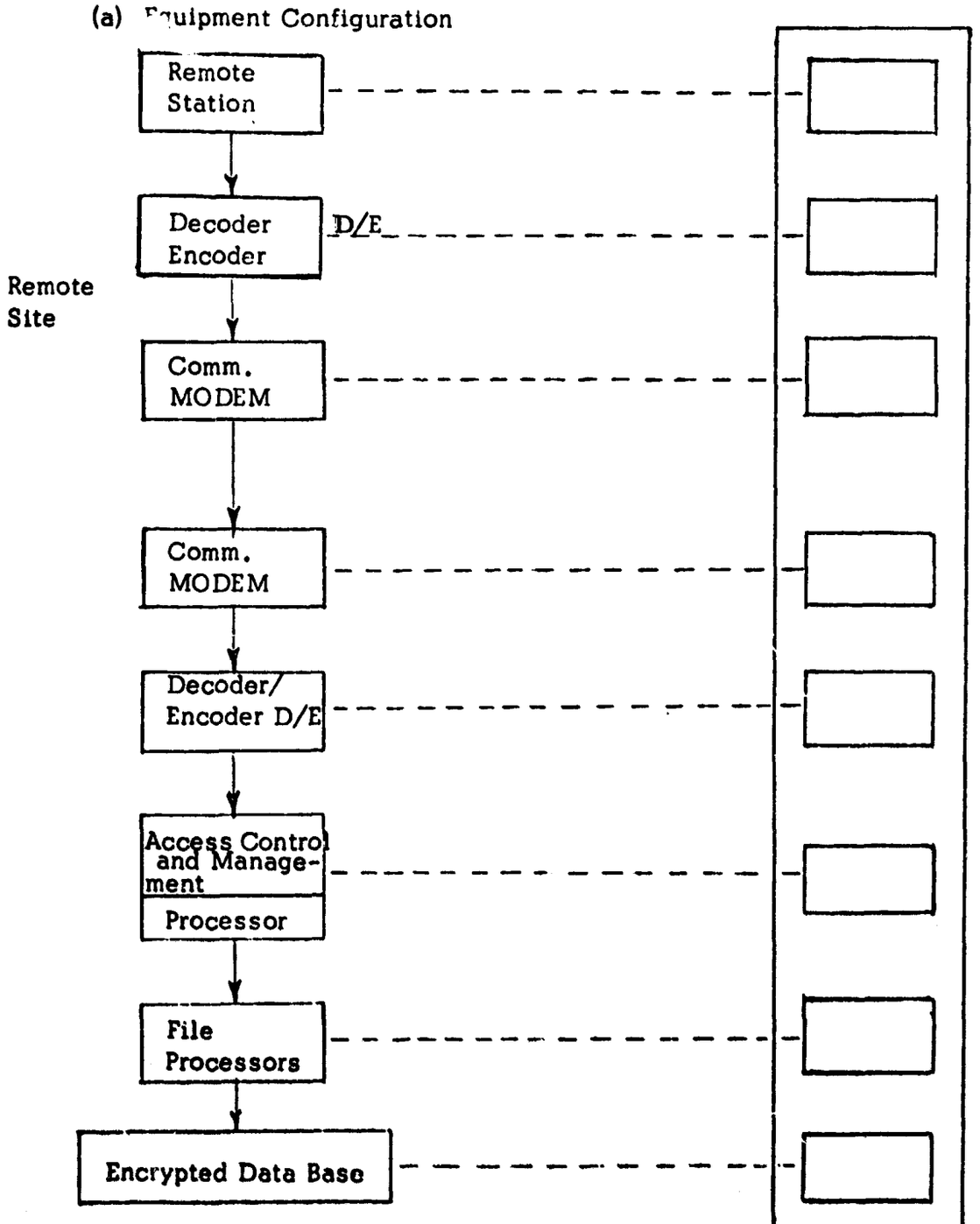


Figure 1

(b) Multiple Remote Attached to Central Data Bank Site

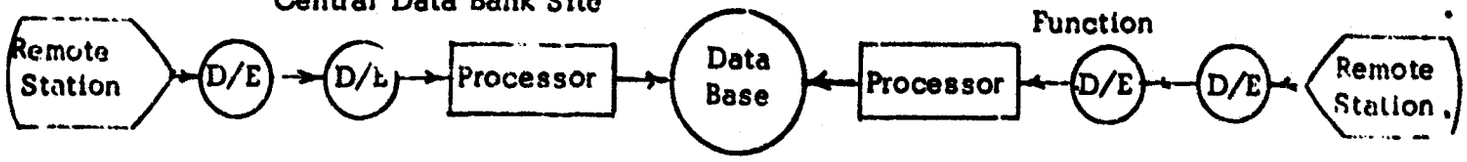


Figure 2

Data Flow. - Data Transformation in a Data Bank

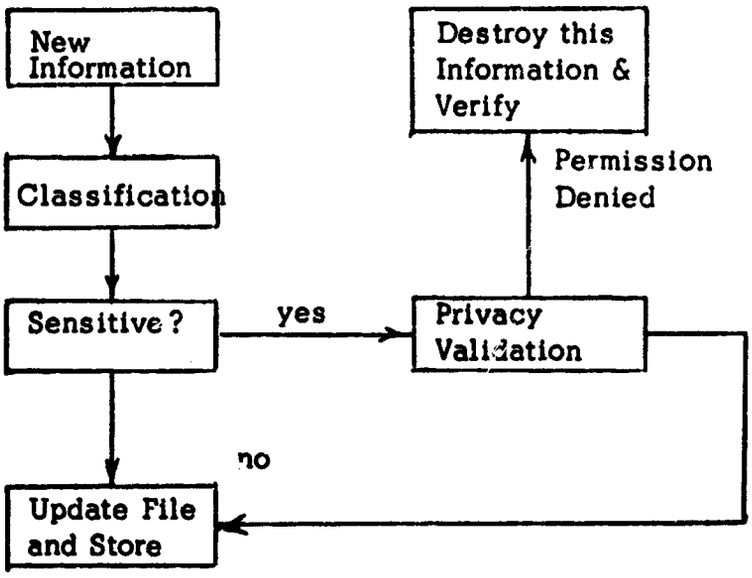


Figure 3

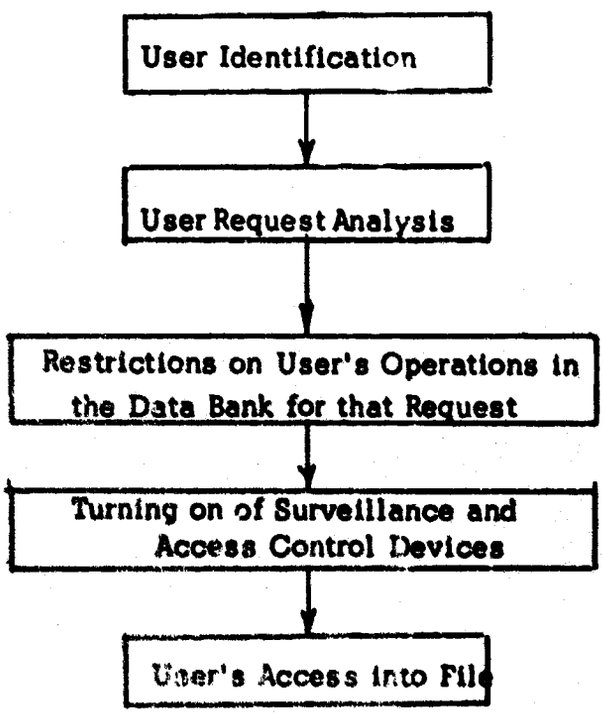


Figure 4

2.6 Information Classification

The first phase of data bank operation from a security point of view is classification of information to be stored in the bank. The information will be either of a sensitive or insensitive nature. Sensitive material will be of the types outlined previously. The classification phase must recognize and validate this information according to a prescribed policy. Certain classes of data will be stored for long periods of time and this information must receive special attention to insure its preservation. This data will be of the type which can be recovered if mutilated or lost, such as bank accounts or personal information. Still other data, such as archives will be non-recoverable if lost.

2.7 Validation Process

The goal of any validation process is to insure the accuracy of information. Validation serves to protect the rights of the individual. A definite problem exists in attempting to protect the individuals rights while maintaining the usefulness of the data bank. Census and medical information require special validation procedures. Presumably, once verified certain classes of information (medical, etc.) may be accessed by a participant at any later time.

2.8 Access Control and Protection of Information

The meaning of access in this context is the display of files stored within this system so as to permit operations like read, write, or update. The reasons for controlling access are obvious, no one wishes another party to initiate activity in his files without permission. The phases of access control are:

Entry & Access (Access Control function)	(1) User identification (2) Analysis of User Request (3) Granting user access to specific files with specific constraints on his activity
Activity monitoring (Surveillance function)	(4) Watch dogging of his file activity
Exit (Access control function)	(5) User's exit from the system - terminates surveillance activity whenever a user leaves the system.

User Identification

The most elemental form of access control is identifying the user as a legitimate subscriber to the data bank. Some way is needed of letting the system know who the user is. This is the objective of the password. The password is a sequence of symbols introduced by the user into the system to be used later by the system in recognizing the user. The types of password schemes that are prevalent [9] are:

1. Single or fixed password schemes
2. Changeable passwords
3. Randomized password. The next password is a function of the current password.
4. Functional passwords which categorize the user regarding his security classifications.
5. Voice recognition scheme.

More elaborate schemes may be desirable since the integrity of passwords is not always assured. One technique to improve this is the use of one-time passwords. Lists of randomly selected passwords would be stored in the computer in a consecutive manner and avilable to the user. After signing in, the user takes the next word on the list transmits it and crosses it off, the processor compares it with its own list and permits access if they agree [9]. Lists are stored internally and are kept in a secure housing and the next password can be queued by a key lock. Another method uses random number generators to accomplish a similar end by compiling a list of one time passwords.

Another possibil ity is that upon logout the user types in a code word that he makes up on the spot. The computer then requests that word the next time the user logs in.

One method that holds promise is that after "signing in" the computer supplies the user with a pseudorandom number. The user performs some unique transformation, preferably a simple algebriac one and sends the answer back to the system. The system then performs the same transformation internally and compares answers. An example of such a transformation would be:

$$P(x) = \sum_{i=1}^{R} \text{odd digits of } x^{3/2} + \text{hour of the day}$$

The advantage here is that a snooper may intercept the argument of the function or answer transmitted to the computer, but he doesn't know the nature of the transformation so the data is useless.

The Dial Up and Call Back option available for the RUSH System [8] is unique in that whenever a sensitive file is required to be accessed, the user's identity is verified by calling him up by telephone and requesting his password to the file. (The user can then modify the password if he did not authorize the access.)

Some passwords may be grouped into functional classes whereby each functional class is associated with a given security profile. Every system has many commands not accessable to every user. Certain passwords might be associated with a restricted subset of these while others encompass a greater instruction repretoire. Additionally, passwords might refer to capability lists, place limits on time of interaction, and limit the number of entries.

Voice recognition schemes are relatively new to this application but a voice pattern might be used as a password or code. At the present state of the art a voice recognition scheme is likely to be expensive and unreliable. Until more reliable techniques are available it should be used to reenforce a more conventional system rather than replace it.

The primary advantage of simple passwords is their convenience to the user and low cost. The more elaborate changeable password schemes may be more expensive, but they are more secure. The primary choice will depend upon the sensitivity of information and possible threats. If the terminal is in same building as the computer then a simple password scheme may be adequate. If large groups share a common terminal more elaborate password schemes may be needed.

Analysis of User Request

The analysis of a user's request involves determining:

1. The users permitted
2. The files requested by the user
3. Interrogating a file owner's list to determine whether the owner has given a particular user permission to access his files.
4. Determining the types of file restrictions required by file owners on their subscribers. These restrictions might be:
   a. Free access (no restrictions)
   b. Paid for access priviledge based on time duration, etc.
   c. Constraints on the use of the file. (See section on Protection Measures)

2.9 Implementing Analysis of User Request

The users request can be analyzed by setting up a user directory which specifies pertinent restrictions. Granting access to files with restrictions involves turning on surveillance functions:

USER DIRECTORY

Password	User's Name	User's File No.	Restrictions on User's Participation	List of Users who are authorized to use his files and constraints imposed	List of Users who have permitted him to access their files and constraints imposed

2.10 Watch Dogging of User's File Activity

Once a user obtains access to a file he is not in unlimited control of that file. He is informed of restrictions imposed upon him by the file's owner. This serves to prevent the user from wasting his time attempting to perform an operation that he has no authority to perform. Whenever restrictions are imposed the system surveillance functions are turned on. These functions enforce the access constraints on all user's. The use has not escaped the surveillance of these watch dog functions even though his access is legitimate. These functions are discussed more fully later.

2.11 User's Exit From System

During the periods of time a user is utilizing the system, he will inevitably create many parallel processes such as arithmetic subroutines filing and searching routines I/O and etc. Only after all user created processes have been completed or ceased should surveillance functions be terminated for that particular user. Obviously printing, plotting, billing of user and activity reports may continue after the user has left the system.

PART III

BASIC PROTECTION METHODS - DYNAMIC PROTECTION

3.1 The nature of the Protection

The type of protection to be provided depends upon the activity in progress. The types of information activities are:

1. Information in transit
2. Information in manipulation
3. Information in storage

Information in all cases is subject to equipment and media failure. During transmission equipment and media failure problems can be minimized by redundancy in the form of error detection and correction codes . Security (protection from snooping) can be provided by the encryption of transmitted data. Similarly protection during manipulative processes can be effected by redundant arithmetic coding and multiple comparisons of arithmetic processes. Security can be provided by having a physically secure processing environment or perhaps even direct manipulation of encrypted data.

3.2 Integrity Management

Control and security of the computational facility itself is one of the most obvious portions of a secure system. Yet, it is the most overlooked or perhaps ignored in a civilian atmosphere where people tend to take for granted any intelligent appearing individual. This is not much of a problem at a small facility as each worker is well known. However, at a large facility it is

relatively easy to get away with something as many people will not question your presence in fear of irritating a supervisor or inspector from another department.

Controls can be placed on engineers and technicians who make modifications or repairs to hardware to insure that a bugging device has not been planted. Measures can be taken to insure that excessive radiation is not being transmitted inadvertently. Insuring that the CRT terminals are secure is important as a lot can be learned from them by a trained observer or operator. Another problem in an over the counter facility such as the one used at the University of Texas is that computer printouts are insecure and it is a trivial matter to steal a computer printout. While this is not normally important in this type of usage it nevertheless could be a problem. Lastly, it is possible for some operator to fool the system by some operation known only to him. The key question is can the file protection schemes be voided by some circuitous manner?

Protection of information in storage is a bit more involved. Stored information is nevertheless subject to the same media and equipment failure problems. The possible protection techniques against system or device failures are:

(a) Redundant coding of stored information

(b) Back-up files - duplicate files

(c) Storage in different media at different locations.

Security protection of stored information can be achieved by:

(a) Control of access to files and media

(b) Encrypting the information stored

(c) Miscellaneous techniques such as parity schemes. (discussed later)

3.3 Protection For Security

As indicated before protection for security can be provided by the following functions: (i) user entry authorization by password schemes(user identification) (ii) Access control restrictions on files and (iii) Security protection of the stored information by encryption. We shall next consider access control protection and static protection of stored information.

3.4 Dynamic Access Protection

This is primarily protection during accessing. All of the standard password schemes may be applied in this phase. Regardless of the password scheme used the essential aspects of all password schemes for an access into the files is the determination of:

(1) Who wants the access

(2) What does he want (which file)

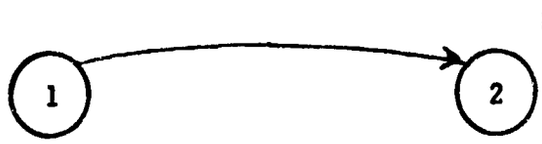
(3) Whose file is requested

(4) Has the owner approved the access, for example:

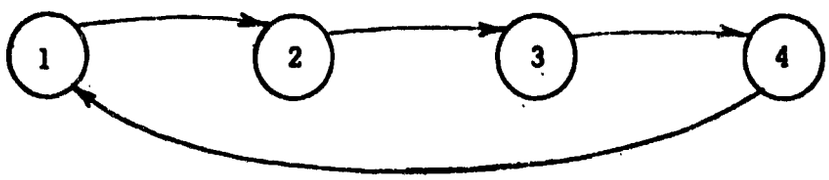
(a) Type of access permitted; READ, WRITE, EXECUTE ONLY, and any special restrictions.

Levels of Access

The simplest type of access is the single level access where only one file is accessed and the file does not initiate any further references to other files as a result of being called. The example below demonstrates this simple case



of a single level access. Multiple level accesses however, may initiate calls to many files as a result of a single call. In the example, procedures and data of file 1 call file 2 which inturn call on file 3, etc. There must be some return link to the calling file. An access path as implied in the



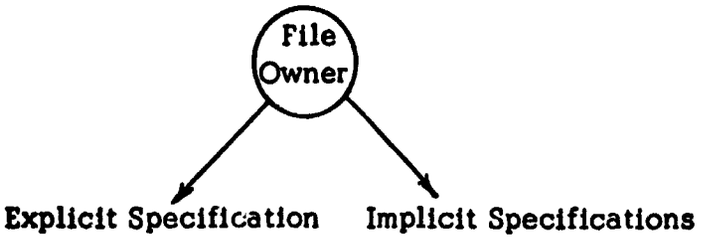
example is in reality an address pair between which a jump is authorized. A gate or entry point becomes an address to which jumps are permitted from specified sources under certain processing restrictions.

In organizing a system it becomes apparent that there is over-head associated with maintaining the user authorizations of the sources and the destinations of many different possible file interactions. This information may be kept either by the operating system or within each file. If it is kept at the system level the cost increases rapidly with the proliferation of users. Thus costs are likely to be distributed over all users rather than those who require complex or special arrangements. Yet, if the system is

small and the file structure is not too complex, distribution of access directories by embedding them within each file may reduce the system burden, yet a centralized file access authorization procedure could be advantageous from the viewpoint of speed and integrity.

3.5 Access Restriction Specification

The purpose of access restrictions specifications are to specify who is to look/see his files and what a user is to be allowed to do. The file owner specifies the user names or file names of those who can access his file. In the course of building a usable file structure two types of accessing specifications modes can be encountered; (1) explicit specification and (2) the implicit specification. In an explicit specification, a file is explicitly accessed by its name etc., and requested to take an action or manipulate data. Implicit access specification results when files themselves initiate further accesses as a result of a request. Such a situation exists when a user requests access to files containing specific procedures. In this procedure "vendors" are polled for the location of the information and if a particular vendor file does not have it, it may refer the request to other files. The explicit access bypasses the searching procedure for the sake of speed, but, results in a complicated file structure while the implicit access slows down response while simplifying communications.



Example of an Explicit Access Specification:

Example of an Implicit Access Specification:

Assume that

U_1 wants income tax procedure file.

U_2 requests U_3 for file but U_3 does not have it.

U_3 requests U_2 for it, U_2 refers to U_j

U_2 requests U_j, U_j has it therefore U_1 gets the file name from U_j.

Some of the questions that may arise when a user accesses a file and finds it locked or in use. Is it desirable to queue a user for access or abort him from the system? The latter can be inconvenient but, suppose the file was being modified for the purpose of excluding that user by removing his authorization? Then truly you cannot now give him access to the file.

Another problem associated with access restrictions is bookkeeping. If the access bookkeeping is kept by the file owner himself, the operating system overhead over the total system will be reduced. However, the supervisor has ultimate responsibility to enforce the file owner's access control policy. Possibly each source file could maintain lists of files it may directly access along with the type of access restriction imposed. The destination file upon receiving an access request, checks the source and operation it is permitted to accept. This could be passed to as many files as was necessary.

3.6 File Restriction

Basic file restriction can be considered to be a function of the protection afforded the memory space. Thus the file owner may specify a number of protection restrictions that may be imposed even if the access is valid.

Read Write Restrictions

1. Read/Write-Examine, copy/update files or write new code, modify old code, etc.
2. Read only (re-entrant) - copy code but cannot change it in any way.
3. Execute only (no copy) - run a program using the procedures in the file without benefit of seeing the code.
   Submit data and receive an answer.
4. Read/Write (single process only) - only one user at a time allowed access.
5. No read, write - No copying or coding.

Boundary or space restrictions

6. Boundary restrictions

Processing Restrictions

7. Modifyable/not modifyable fields in records

8. Specified authorized user interaction i.e., system can modify anything (R/W) but, user can read only. (ROM)

9. Execute only

10. Specified size bounds on new records added etc.

An example of how the file space is organized is illustrated in Figure 5.

File Memory Space

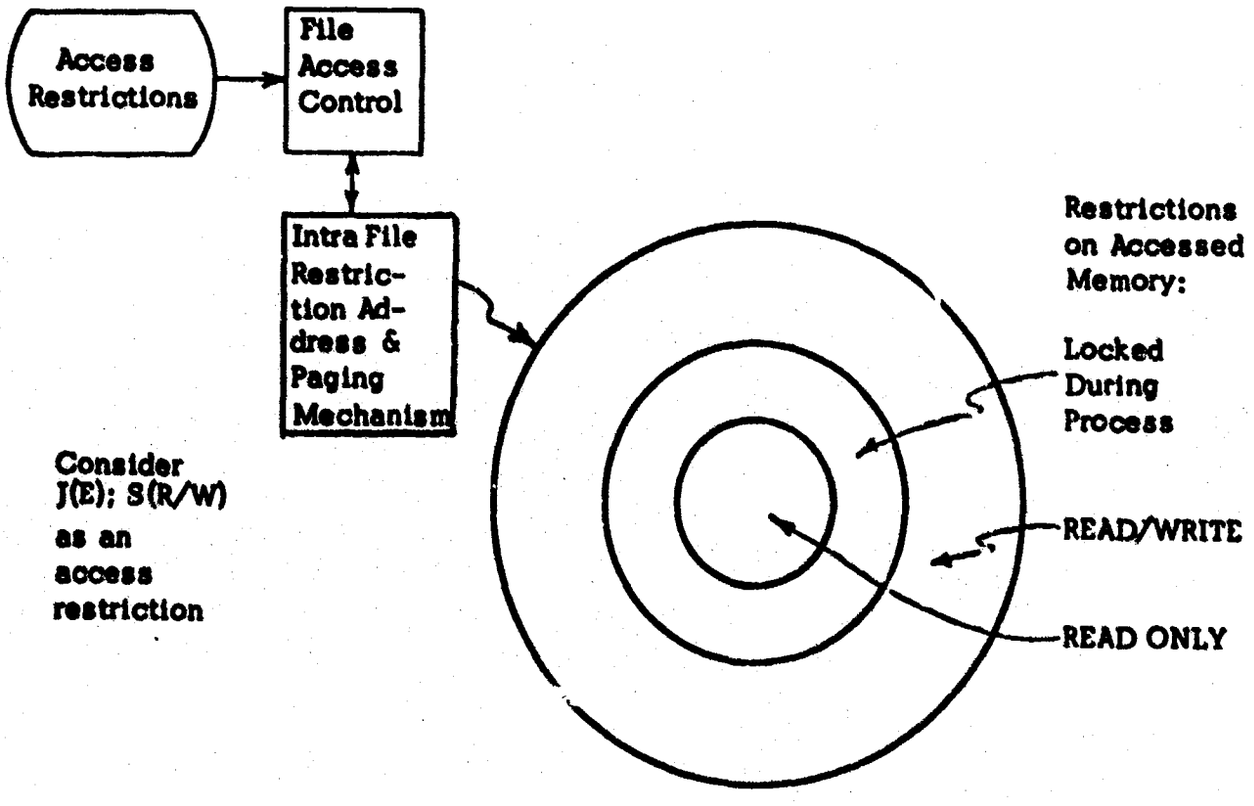


Figure 5

The meaning of the J(E); S(R/W) access restriction would be to limit user J to an execute only mode while he is in the accessed space, while user S is restricted to Read or Write. Thus, the actual restriction on the file for a specific user will be the union of the access and the file restrictions, viz.,

Actual restriction =

Access Restriction ∪ File Restriction

3.7 System Performance, Threat Surveillance Management

Associated with any secure system would be the ability to trace and record all accesses of protected files. Such a recording function would enjoy the same degree of protection as the access control routines. If the recording function is to produce meaningful data it cannot be snooped, modified, or aborted. The recording function could record information such as (1) Access path associated with every successful and unsuccessful access. (2) What did the access change, read, execute, or did it restructure the file or change the access paths.

The justification for implementing such a function go far beyond the idea of merely providing a security trace routine. Consider the large problem of evaluating system performance and efficiency. Most systems are optimized with respect to the manufacturer's generalized view of typical operating conditions. In the field the model doesn't always fit. As a result, many systems undergo evaluation and modification in the field. Essential to these evaluations are statistical studies of such things as (1) File activity (2) Reliability

and malfunction detection (3) Security effectiveness. Thus it is apparent that the recording function could justify itself in many ways.

3.8 Functions of a System Activity Recording Function

(1) This is an independent watch dog of activity always monitoring the processes in parallel.

(2) Measuring and accounting-Billing according to time and processes requested.

(3) Records of activity would also be useful in determining intermittent faults, etc.

(4) It should provide simulated threats and intrusions into system to note the effectiveness of countermeasures.

To carry out these activities would require that the functions have

(1) Parallel processing ability

(2) Access to all registers

(3) Collaborative computation on common data

(4) Ability to freeze system for purpose of taking snapshots of registers, access paths and terminal intersections.

Information gathered must be selective. The advantage to this is that it requires less storage space. The disadvantage of selective data gathering is that useful information may be lost for future analysis.

An example of what a surveillance function could do is the technique of threat monitoring. Threat monitoring is a logical extension of the password idea. This approach relies on the detection of attempted or actual penetration of the system or file to provide real-time responses to the system supervisor.

Monitoring uses cancelling of activities, tracing or post facto analysis to aid in the identification and classification of penetration attempts. A special function records all attempts at entering a file, whether too much time used, etc. and reports this to the file owner and the user so appropriate action can be taken. Suggestions have included mounting of removal files on drives with special disable circuits. Perhaps the disk pack itself could be filed away under lock and key much like a tape. The disk pack could be manufactured in such a way as to preclude its use on any but a specified drive unit. When a certain file is called for, the requester is identified and the owner notified. Furthermore, the disk drive itself might be enabled only with the proper authority. These surveillance functions would insure that working memory space is wiped out after every completed access.

3.9 Threat Monitoring

What follows is an example of a possible threat monitoring scheme as mentioned in the reference [9], preceded by an example of a typical password scheme in common use today.

STANDARD METHOD:

LOGIN, MAN 2793,ACT 5-172

PASSWORD?

PRIVACY3

FILE NAME?

THREAT MONITORING:

(USER'S CONVERSATION)	(MONITORED RECORD)
LOGIN, MAN 2793,ACCT 5-17-2	10:14:08 TERMINALS, 2793, 5-17-2, LOGIN
PASSWORD(5742)=?	
4527s.	10:14:53 TERMINALS, 45273 5742 UNSUCCESSFUL PASSWORD
TRY AGAIN. PASSWORD(9360=?	
69032.	10:14:44 TERMINALS, 69032 9360 PASSWORD OK
FILENAME:	
PRIVACY.	10:18:20 TERMINALS. OPEN-- PRIVACY-FILE
OPERATION:	10:18:20 TERMINALS, SORT RECORDS BY DATE
SORT RECORDS BY DATE	

The secure data bank should continually be tested for weak points. Ideally , the bank would utilize its own personnel to attempt penetrations in order to test the effectiveness of security schemes. File activity studies are essential to the continuing efficiency of the system as inactive files may have to be purged to make room for active users. This periodic purging would be the result of information received from the surveillance function.

3.10 Alternate Protection Schemes

In large time sharing systems such as the CDC 6600 memory must be provided some routine protection. We are not speaking of necessarily thwarting covert attempts to do damage, but it is obvious that in a large utility some method must exist for preventing accidental damage to other programs.

For one thing, a user must not be allowed to interfere with the time sharing monitor of input/output commands, halt commands, etc. The latter capability is obtained by denying the user certain privileged instructions generally reserved for the use of the operating system.

Preventing overwrite and other calamities is generally provided by memory protection schemes such as relocation and bounds registers, segmentation and paging. Memory boundary registers in time-sharing systems like the CDC 6600 prevent this interference between programs. Registers are used to store the upper and lower bounds of a program. If the program attempts to address any program segment outside of its range, an interrupt is generated, and the supervisory program takes control. On the 6600 there are provisions for the handling of seven such programs at one time.

In the 6600 each program is assigned a control point for the execution of that job. The operating system uses control point zero while other programs are assigned to the remaining six. The operating system itself cannot refer to locations outside of its own field length. Only the peripheral processors have such free access of core. A requirement of memory bounds registers, however, is that instructions and data on any one program be contiguous. Other schemes can be used of course and Richards [9] suggests memory protection by two methods:

1. Divide main memory into blocks. Associated with each one is a flip flop which is used as an access flag. All blocks are set to one if they can contain the program.

If program references are made to a block not containing the program an interrupt is generated.

2. Another method attaches extra bits in each word to identify each program. This method appears to be wasteful of bits however, it can be used effectively against someone jumping into a block of memory without proper authority. A variation of this scheme is discussed later under Static Protection.

All of these methods protect contiguous portions of memory (real or virtual) from an alteration by an errant program. They do not provide protection from unauthorized access. This is generally handled by the access control routines mentioned earlier.

Grahamn [7] proposes a protection scheme at the hardware level which affords protection of memory in a more versatile manner than is given by a simple memory bounds register. The key component to Grahamn's approach is the segment, where a segment is defined as a contiguous block of words whose length may vary. Some computers have segment addressing where each word is addressed by an ordered pair of integers (S,W). S is the segment number, and W is the word number within the segment. S ranges from 0 to the maximum allowable value, and W ranges from 0 to the current length of the segment. Associated with a segment is its descriptor which contains the base location of the beginning of the segment, length of the segment, and the access indicator.

Beginning	Length	Access Indicator

DESCRIPTOR

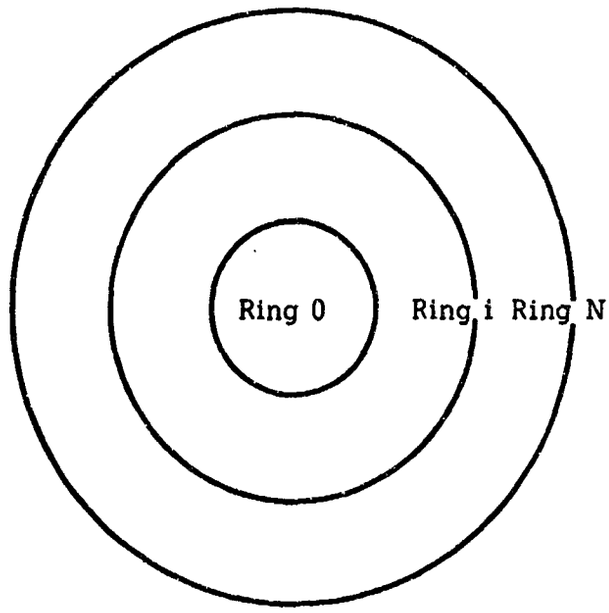
The access indicator indicates the mode of the access, slave mode or master mode. In the slave mode any attempt to execute privileged instructions causes an interrupt. In the master mode any instruction may be executed. If the segment is a procedure the access indicator tells whether it is to be executed in the slave or master mode. Finally it includes a fault bit which when non-zero causes an interrupt on any attempt to reference the segment when operating in the master mode. If the fault bit is non-zero, no access at all is permitted.

For every segment that a process may access or has potential access, the corresponding descriptor resides in a distinguished segment called the descriptor for that segment. All systems will have large numbers of descriptor segments, one for each process. Whenever a process is executing, a register segment for the executing process, called the descriptor base register, indirectly defines the set of segments to which the execution has potential access. To implement layered protection an additional field is added to the descriptor. This number field is called the ring number.

Beginning of Segment	Length	Access Indicator	Ring No.

DESCRIPTOR

The ring is an ordered disjoint set for 0 to some maximum value.



A fault will occur if a procedure executing in ring i tries to execute in ring j where j is less than i. However, a procedure executing in ring i has access to a segment in ring k if k is greater than or equal to i, subject to access restrictions imposed upon it by indicators in its descriptor. Certain classes of sphered service routines can be given a range of rings to operate in so that efficiency may be increased.

Grahamn's scheme has disadvantages in that it rules out memories such as associative memories which are content addressable rather than location addressable. Also, if a data bank has many different data fields with different levels of access, each datum within its 2 or 3 word segment may result in the overhead becoming prohibitive at the present state of the art. Lastly, it imposes a hierarchy on every piece of data that is in the data base and this is not necessarily desirable. Dennis and Vanhorn's [6] scheme attempts to accomplish the same end, but suffers from the first two drawbacks. As mentioned before any technique attaching

authority items to each file suffers from the problem of duplication of pertinent authority items for protected fields in one file.

3.11 The Role of the Status Word In a Secure Time Sharing System

The status word carries the information needed for continuity between programs. The status word however, can carry more than just information required for carrying on the basic program in a time sharing system. In a machine using a table look type of memory addressing scheme (virtual addressing) the basic information required would be the Address Table base entry (Acb), the program number, and a pointer to the register contents. The register contents is the information contained in the operating registers at the time a program was cutoff from execution having exhausted its time slice. We must now hold this information for the next burst of execution the program receives. The information we must keep would be the contents of the A-register, Q-register, Instruction register, memory address register (MAR), call stacks and etc. Since register content information is only a temporary storage and it must be loaded and accessed quickly, some type of active memory is required, which is usually a bank of registers.

There is a lot of information that could be carried in the status word that would make the system more versatile. These include Addressing Mode, Page Size, CPU time used, Pointers to a crypto key, enabling bits for crypto units, and I/O devices, Passwords, and references to a security profile for the user-owner through capability lists.

3.12 Status Word In a MultiProgramming System

Prog no.	Acb	AM	RCP	Password	Page Size	CPU time	I/O L	Directory	Key

Definitions:

Prog no. - supervisor assigned program number

Acb - Primary address table base entry location

AM - Addressing mode of system -- normally set to the relative addressing mode requiring that a virtual address is transformed into a physical address by table lookup. In case of failure of the address table it may be set to the direct addressing mode in order to bypass the address table.

RCP- Register Contents Pointer- sets a pointer to the bank of registers which comprise the Register Contents Table. The RCT holds the contents of registers at the time the program's execution time slice is completed. The operating registers are reloaded from this table when execution of a partially completed program is continued.

Page Size - A system option for varying the page size for a particular operating environment for purposes of optimization.

CPU-time - Accumulative elapsed time clock for accounting and surveillance putposes.

I/O Lock - When enabled, locks out all I/O devices except the ones authorized by the I/O lock field. Each I/O device is assumed to have its own identification key.

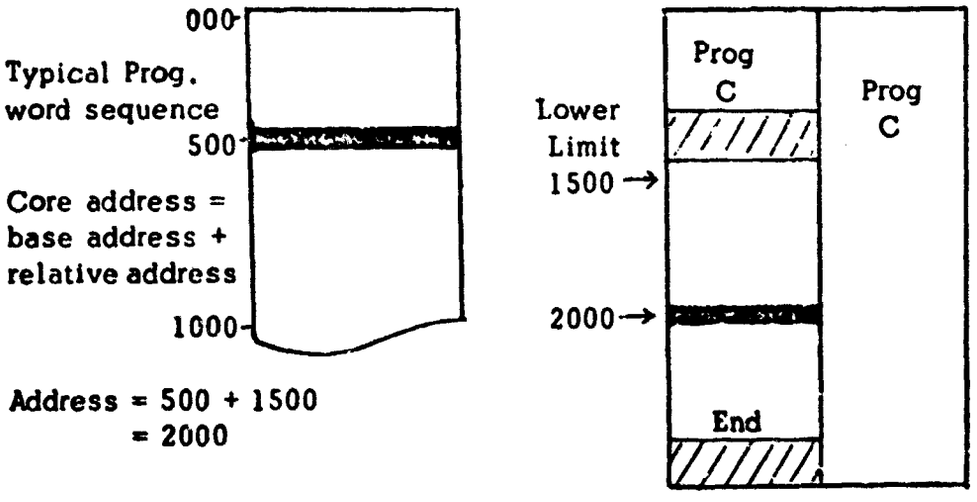
Directory - Pointer - Pointer to the access restriction and file restriction table which contains the permitted modes and paths of access and any file processing restrictions imposed upon the user of the file by the owner. Contains user certification and capabilities. In the case of open or unrestricted files this field would contain a special code.

Key - If equipment has a cryptographic capability for transmission between two points, then a pointer to the key used would be kept here. The location and contents of the key storage is kept secure and cannot be accessed by a user. The keys are normally changed daily or more often on request. This could be accomplished by a computer generated circular list sufficiently large so as to insure that the keys were not used with any predictable regularity. Whenever the proper enabling bit is set in the status word, the supervisor will cause the top member of the key list to be loaded into the cryptographic unit. Associated with each key is the program number to which the key applies. The supervisor meanwhile cues up the proper key at the receiving end unless the information is to reside at a remote location in an encrypted form, in which case the key is stored locally and retrieved whenever that file is fetched from the remote location for local viewing or processing.

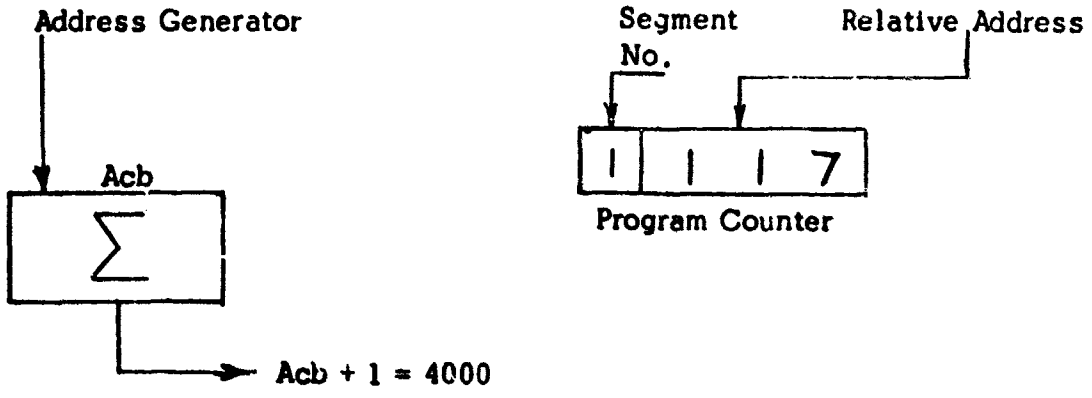
If the only portion of the system to be protected is the conversation between a central site and remote terminals, then once a key is used it is destroyed or discarded. If the terminal is storing information at a remote file, then the key must be stored with the file to facilitate its retreival at a later date, or as mentioned previously, it must be stored locally for purposes of decoding the information. Certainly it is not necessary that all information be encrypted, and not all conversation or storage would utilize this approach to protecting the data.

3.13 Protecting Working Memory by Virtual Addressing

In most systems, a typical program sequence is mapped into core or memory by setting a lower address base limit and an upper address limit between which the instructions are loaded into memory. If the program attempts to jump to an address outside of the limits specified by the supervisor at the time the program was loaded, then an interrupt is generated. This scheme works well and is in common usage today, but it nevertheless has some limitations from the security point of view. It is subject to defaulting due to hardware failure. It does not allow protection of select portions of memory by the programmer. It requires periodic reshuffling of code to make room for entering programs, necessitating housekeeping chores.



In a virtual addressing scheme the programmer does not use an actual core address in writing his program. All addresses are virtual and relative. Although it appears to the programmer that he is directly addressing core, in reality he is being mapped into core by a transformation table. Let us look at how this would appear in its simplest form. Basically the current instruction being processed is sent from the program instruction counter to an address generator. The address generator receives as input the leftmost digit or digits of the instruction identification counter. The leftmost digits thus correspond to the segment being processed. This number could be varied in relation to the page size utilized to obtain the most advantageous page size and segment size ratio. The other input is the address contents base or Acb which is initially assigned by the supervisor on the basis of core requirements.



In the example shown, the address generator has been given the base entry in the address table as Acb + 1. If the instruction counter had read 2111, it would have corresponded to Acb + 2. These values serve as direct table lookup indices. In the example shown.

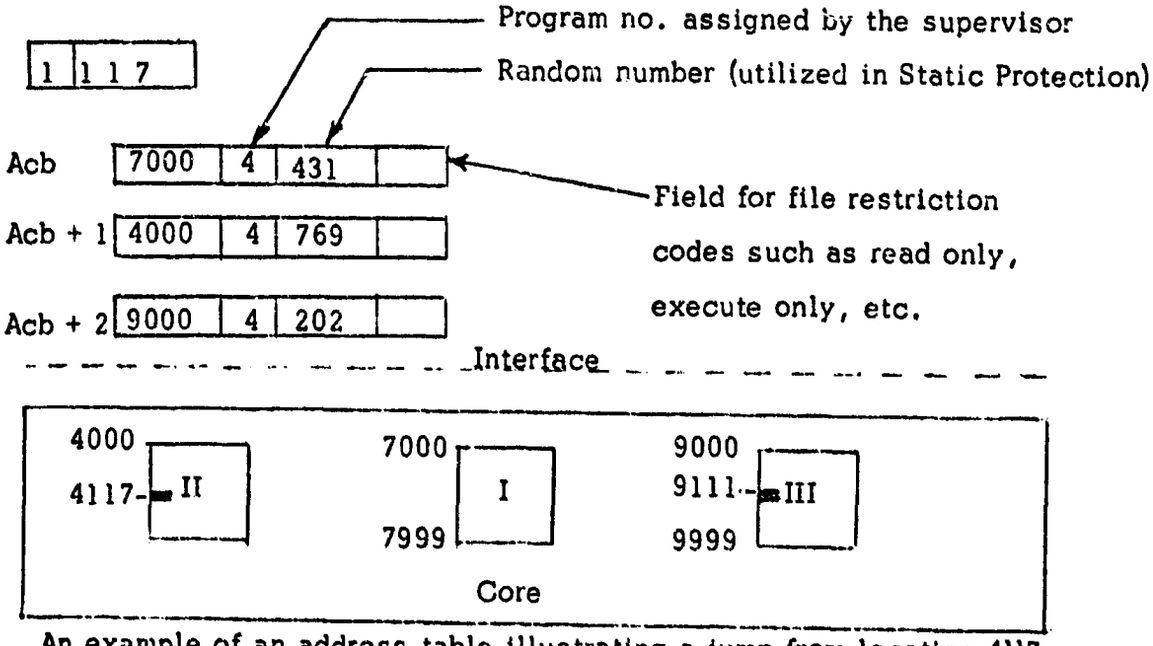
Acb + 1 = 4000

Acb + 2 = 9000

These numbers are the physical addresses in core of the start of pages 1 and 2 of this program. Thus instruction 1117 is found by adding the rightmost number to the table entry after blanking the segment portion. This gives 4000 + 0117 = 4117 as the physical address in core. The programmer is thus unaware of the transformation and core appears to be directly addressable.

The address table also provides a convenient place to store other information important to the program. When the program is initialized and the base entry into the address table is specified, a field in the table is loaded with the program number assigned by the supervisor. A check can be made on the program number before the jump to the physical address is taken. This check provides a way to verify the fact that the program addressing that page in memory is in fact the one to which that page was originally assigned. It is possible to store codes relating to the type of access and file operations to be permitted the user during run time. Thus, once a set of access profiles are established for a user, they are available for the duration of the execution and need not be reinitialized, if the program is waiting for completion as a result of an interrupt.

The address table may also be utilized to store a random number to be used in a Static Storage protection scheme to be discussed in the next chapter.



An example of an address table illustrating a jump from location 4117 to 9111 in physical memory.

PART IV

STATIC PROTECTION

4.1 Privacy Transformations or Encryption:

One obvious but, troublesome way to improve security in a system is to arrange information in such a form that, if it is compromised or stolen, it will be of no use to the party or parties obtaining such information. This means performing some sort of transformation on the data as it is transmitted or as it is loaded into a file. This involves some encoding or encrypting process whereby the subject data bears no resemblance to its original form yet still contains the original information. The successful decoding of this information would be possible only for those having the "key" or inverse transformation algorithms. As a result, wiretapping and other covert methods to obtain privileged information would be minimized as threats. There are numerous cryptographic schemes, and many of them are used in military communications networks. Such schemes have not been used widely in civilian applications because of their inherent high cost. Ideally, this problem could be lessened by designing the cryptography equipment into the basic system design. It is certainly not necessary that all parts of a system be subject to the devices scope. If the crypto device is a part of the hardware, it can be expanded and tailored to fit any requirement.

To date most transformations have been made for the purpose of communications between "secure" points, such as from one terminal to another. Whether a point is secure or not is part of a larger problem, that of total system security as opposed to basic file security. An obvious benefit of a transformation is that the burden of file security is relieved during periods of transmission from one point to another. But, what about security of the facility and of the file itself while it resides in memory, on a disk or tape or while it is in execution. A start might be made by encoding certain classes of data as it is put into the file. This procedure would make the data secure even if access management techniques break down and the file is unintentionally displayed.

If such information residing in a file is encrypted and is accessed by a remote terminal, it is possible to encode the information again prior to transmission. The net result would be a message that is double encrypted requiring two decoding steps at the receiving end. Such a system is illustrated in Figure (6).

4.2 Problems Associated with Privacy Transformations:

There are many problems to be solved in the area of cryptographic design as related to a computer. One yardstick determining the validity of a design is whether or not it can be discussed openly without someone pointing out a weakness. Certainly any procedure must be of a nature that engineers and computer scientists can talk about it freely without in any way affecting its usefulness. Baran [2] points out that one reason why

cryptographic equipment has been so expensive is the insistence that it be absolutely secure. In a computer application, the cryptographic equipment could be something less sophisticated than is required for national defense. It may be wiser to purchase more equipment of lesser sophistication. The philosophy behind this reasoning is that while a determined party might occassionally break a code the net result would be much less information lost over the long haul. Most people would not have the patience or time necessary to decode large streams of information looking for a particular item.

Message interception can be made difficult for someone knowing the code by chopping the message up into a series of segments which are transmitted over different lines together with other traffic and garbage. It then becomes almost impossible to reconstruct the message even if the code is known. Following a technique in long use in military communications, the line can always be filled with redundant traffic or garbage so that there are no apparent periods of inactivity. This procedure makes the beginning and termination of a message difficult to determine.

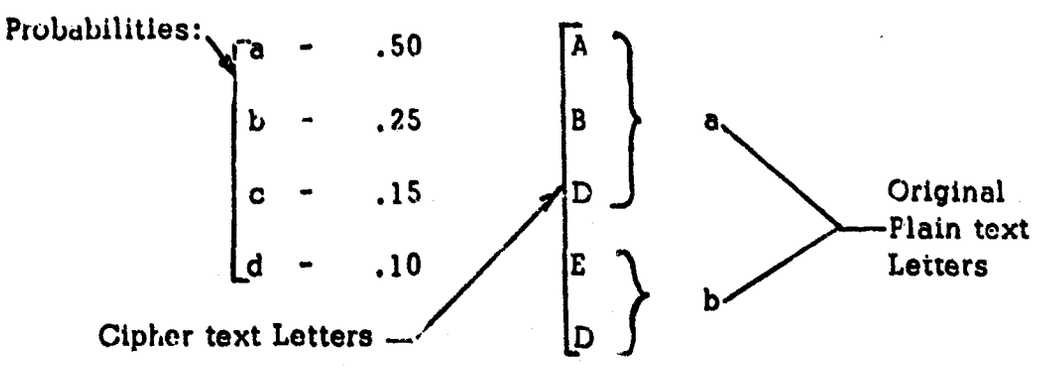
The point is that while the overhead may be great, we can buy any level of security we are willing to pay for. A basic problem is what do you do with a sensitive program when it is in execution. When a program is brought in from an outside source, or filed into central memory for the purpose of execution what do you do with it? Do you leave it in encrypted form while it is in execution or do you transform it back to its

original form. If it is left in encoded form, how do you design equipment that can execute an encrypted program. Ideally, the programmer or operator should not know the particular transformation utilized during the execution period. At the present, however, any program probably would have to be decoded before the arithmetic unit could produce meaningful results. But, what if someone is looking at your arithmetic unit while the program is executing? If he does so and the information is not in an encrypted form the whole purpose of the scheme is defeated.

4.3 Cryptographic Schemes:

Most cryptographic schemes in use attempt to average or transform a text into some hopefully unrecognizable form. This is accomplished by (a) permutation of text symbols (b) reducing the occurrence of high frequency symbols characteristic to the language used (i.e. leveling). Obviously the leveling process requires some substitution or transformation whereby the high frequency characters are substituted for by low frequency symbols. Certain vowel-consonant-consonant-consonant, and vowel-vowel combinations are clues to the word used in a certain context. The example shows how high frequency letters in the plain text can be substituted for by several cipher text letters so as level the text.

Example of Leveling Technique for Given Character Occurrence:



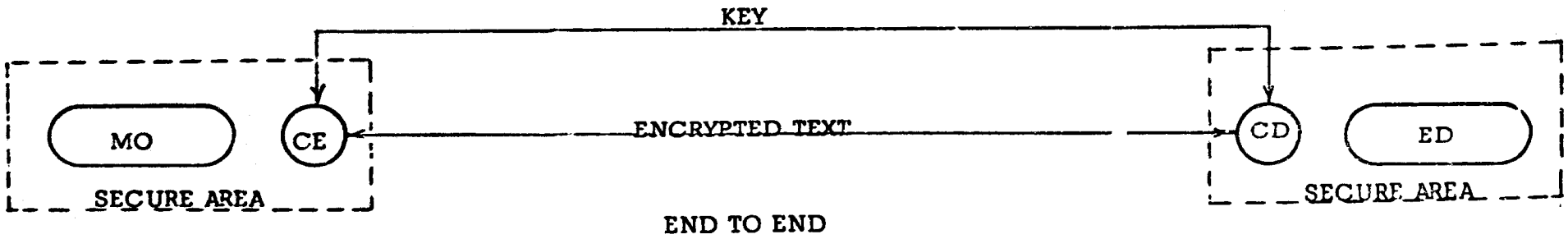
End to end encryption: In this approach a cryptographic device is connected to one end of a line adjacent to the user and a reciprocal unit is placed at the receiving end. Figure (6) shows such a unit. Figure (7) demonstrates a method suggested by Baran [2]. The transformation used in Figure (7) uses two pseudo-random binary streams generated by two "key" generators at each end of the link. The generator at one end generates a long non-periodic digital stream which is combined with the outgoing message by some logical transformation. The resulting combined stream is the encrypted text. This process is a logical proces and is complemented at the other end to decode the message. One problem Baran points out is that the generator must have statistical properties that make it appear as a totally random digital noise generator. Additional facilities must be included for synchronizing the time base clocks at each site. Any transformation used to combine the text and the key must be generated characters of equal probability such as the binary values 1 or 0.

4.4 The Crypto Process:

Examining the algebra of the technique of Figure (7), a "logical-add" circuit is used to perform the equal probability of transformation required to allow reciprocal operation at the receiver. The operation:

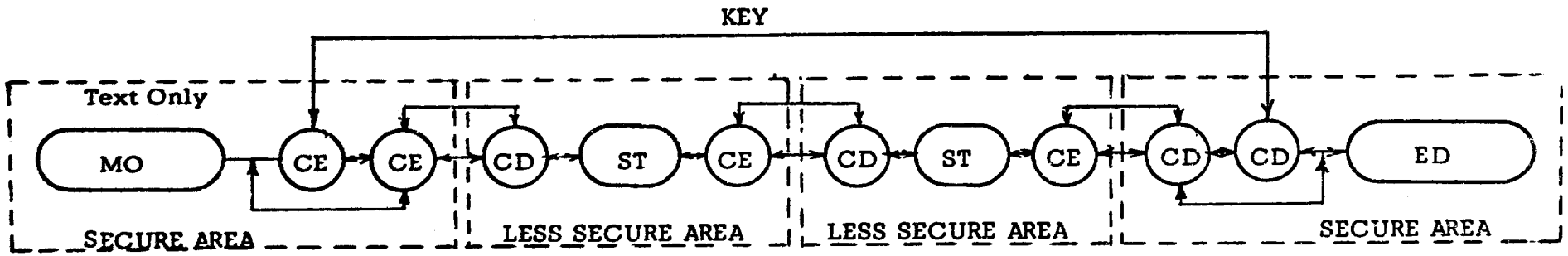
$$M \oplus K = E$$

$$E \oplus K = E$$



END TO END

MO = MESSAGE ORIGINATION
ED = END ADDRESSEE
CT = CLEAR TEXT
CD = CRYPTO DECODER
CE = CRYPTO ENCODER
ST = SWITCH STATION



DOUBLE ENCRYPTION

Figure 6 - Types of Encryption Systems

Where M = the original message

K = the key

E = the encrypted text

$\oplus$ = logical-add-transformation-ex-or

The truth table shows an ex-or or add without carry transformation which has the reciprocal properties required. Of course more complex operators are possible.

Unbreakable ciphers are possible by using an infinitely long non-periodic key. However, it would be necessary using this complementation technique to store the key at both the transmitting and receiving end. Considering the high data rates, the storage requirements for such keys would likely prove prohibitive.

Keys should be chosen that will not reveal any periodicity for the length of time they are used. It would be feasible to change such keys (Figure 8) daily to circumvent this difficulty. It is possible to generate long series of bits from a moderate length key by utilizing all the possible combinations a finite length binary key possesses. The key length that can be generated by a key of length $N = N2^N$, where N is the number of flip-flops or storage elements in the series generator. If $N = 50$, $N2^N = 50,000,000,000,000,000$. Such a key must be chosen carefully however, so as to insure that its statistical properties do not reveal its nature [21].

The complementation technique demonstrated in the previous section is known as the Vernam code and was originally applied to telegraph transmission. This technique could also be useful in static storage of information.

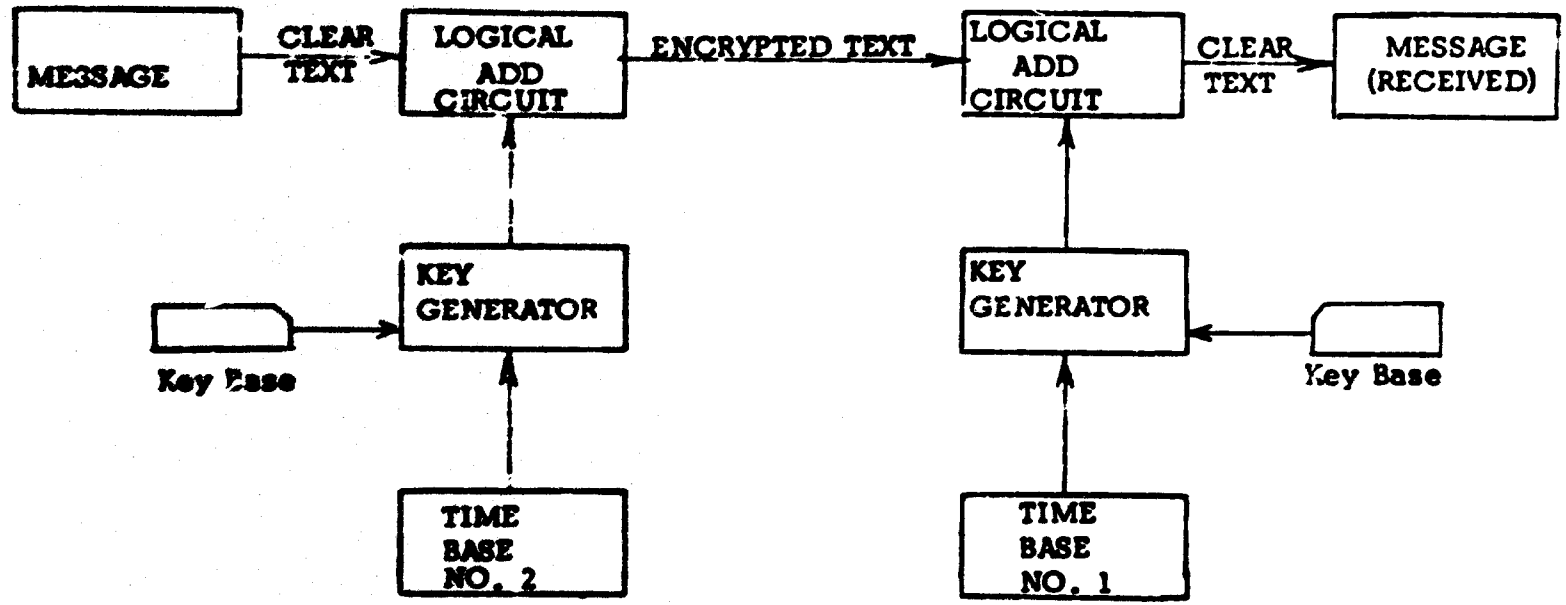


Figure 7 - The Crypto Process

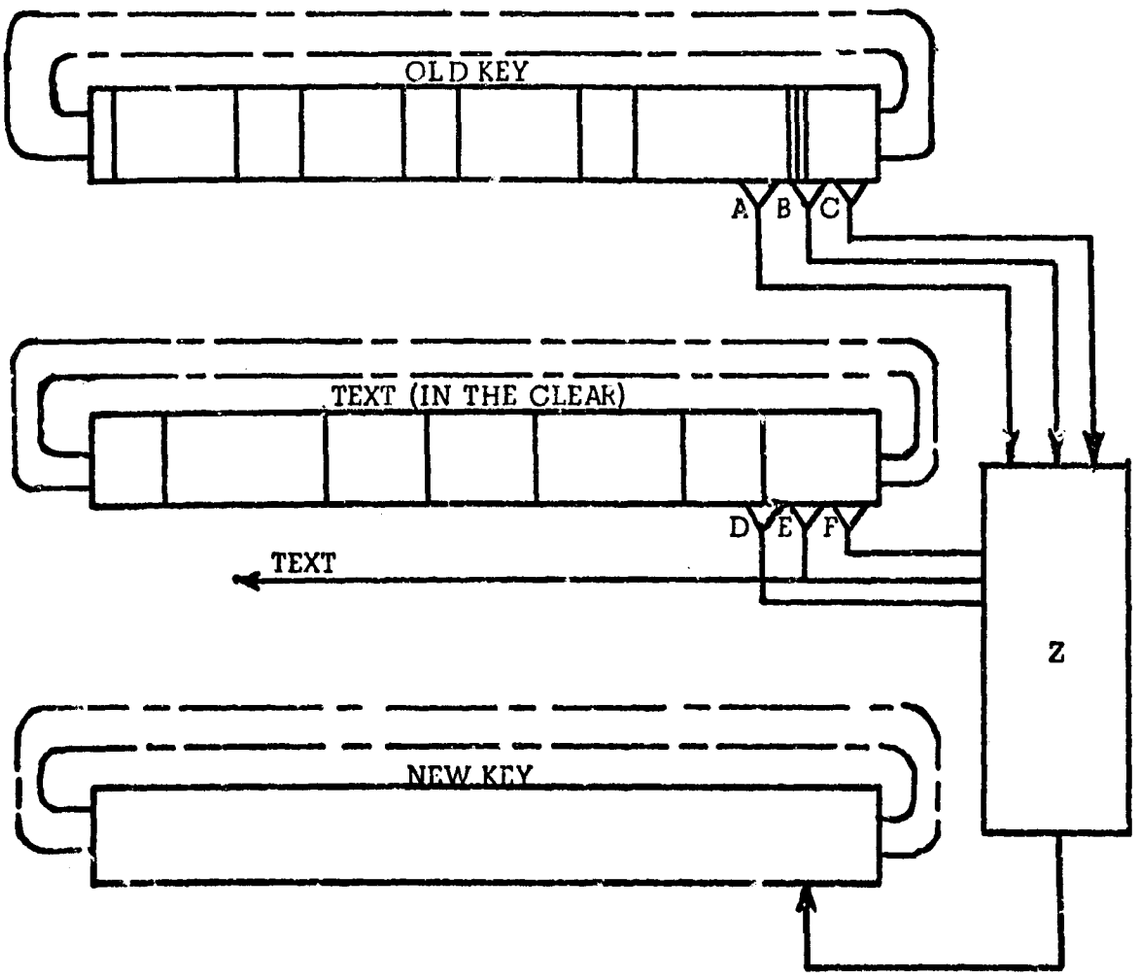
When applied using a non-periodic totally random one-time key the system has been touted as an "unbreakable" system [10].

4.5 The Possibility of More Complex Keys:

If the need is felt for some complex keys, such keys can easily be generated. Previously we were considering only a very simple key. By using a multiplexing drum to store the key and by using multiple magnetic heads to input and output the key, the number of possibilities is extremely large. Figure (8) The three drum bands could serve as follows; the first band is used to store the key used in decrypting the last message, the second drum band stores the deciphered text in the clear that has been derived from the old key. The third band stores the new key generated by the black box Z.

If A, B, & C represent three heads spaced some specified number of bits from each other on the multiplexing drum, then D, E, and F, would represent a second set of heads containing bit patterns from the clear text. These form six separate inputs to the black box Z which can permute these inputs into a number of outputs for the new key which is stored on band three.

Consider six different heads to form six separate inputs. There are many ways six inputs can be arranged to form separate and distinct output functions. Any Boolean function can be expressed as the logical sum of a series of minterms. The minterms being the logical product of the inputs to each head. For a six term, six input function, there are 2^{2^6} or 10^{20}



KEY GENERATOR

Figure 8

allowable combinations, where the number of combinations can be computed from the relation 2^{2^N} if N is the number of variables. Complex logic circuitry may be required in some circumstances but may be simply realized. By using rather simple techniques such as the one discussed, it is possible to add tremendous complexity. There are other approaches to encrypting data. We shall now investigate and evaluate some of them.

4.6 Evaluation of the Vernam Scheme

To study the overhead associated with the Vernam scheme two short computer programs were written which take a message, encode it, and then decode it. Timing evaluation was made of the time required to encode and decode the message. This was done to gain an insight into the overhead associated with using the Vernam method for static storage of information.

The two programs used the Vernam technique, however, the first program uses a periodic key of 60 bits or one computer word, as this is the standard word length on the CDC 6600. This key was used as many times as was necessary to cover the text. The other program used a non-periodic key that was at least as long as the message. This approach helps to alleviate the frequency problem inherent with a periodic key. As will be shown, the one-time key is a superior method from a theoretical point of view but, it is burdened with the overhead problem of the storage of long keys and the generation of such keys having the proper statistical properties. All of these problems increase overhead of the system.

The periodic key demonstrates one of its weaknesses when the standard internal representation is compared to the scrambled text. A ten character word full of blanks is represented internally as 5555555555; whereas, in the scrambled text, the representation is apparently <=) = %% 9 % 7.8. This code could be used as a starting point to find the key. For example, since it is known that we are dealing with a reciprocal process, (assume the snooper knows as much) we can examine the encoding algorithm E $\oplus$ K = M and the decoding algorithm E $\oplus$ K = M, substitute and get E $\oplus$ K $\oplus$ K = E. Thus if the key is applied twice the message is recovered, since M $\oplus$ K = E and in the < in the CDC display code is 72 octal or 111010 binary and a blank is 55 octal or 101101 binary:

111010 = Encrypted letter

$\oplus$ 101101 = Unencrypted blank

010111 = Recovered key letter

This is 27 octal or W in display code. Similarly all other key letters could be found to be WAGARRISON. Knowing the key, nothing is secret.

The problem is complicated in the non-periodic key system as discovered by the Army Signal School cryptologists [10] and commonly referred to as the "one time system". When the basic Vernam system using a random key and the non-repeating key are combined, the result is an impressive cryptographic technique. The system is both theoretically

and practically unbreakable. Even if time is no object, the system is unbreakable. By comparison, some systems are unbreakable only because time is limited while the usefulness of the encoded information can be realized.

The Vernam method does not lend itself to frequency analysis techniques or linguistic trait examinations. There is no way of sorting out any recurring traits because the key is totally random and does not generate any internal signs as indicated in the specific example. Trial and error should eventually lead to the plaintext but while it would bring out the true plaintext, it would also bring out every message of the same length. If every four letter key were tried on the first four letters of an encoded text, the result would merely be a list of all possibly comprehensible four letter words. Time for the encoding/decoding with the one-time key system is the same for all practical purposes. The average encode decode time for the two cases was 2×10^{-3} sec although the first message was 5760 bits long while the second was 480 bits long. Evaluating the cost/bit:

$$\text{Case I (periodic key) } 12 \text{ cards} \times 80 \frac{\text{character}}{\text{card}} \times \frac{6 \text{ bits}}{\text{character}} = 5760 \text{ bits}$$

$$\% \text{ Over head} = \frac{2 \times 10^{-3}}{.304} \times 100\% \approx .658\% \text{ based on } .304 \text{ sec. of central processor (CP) time}$$

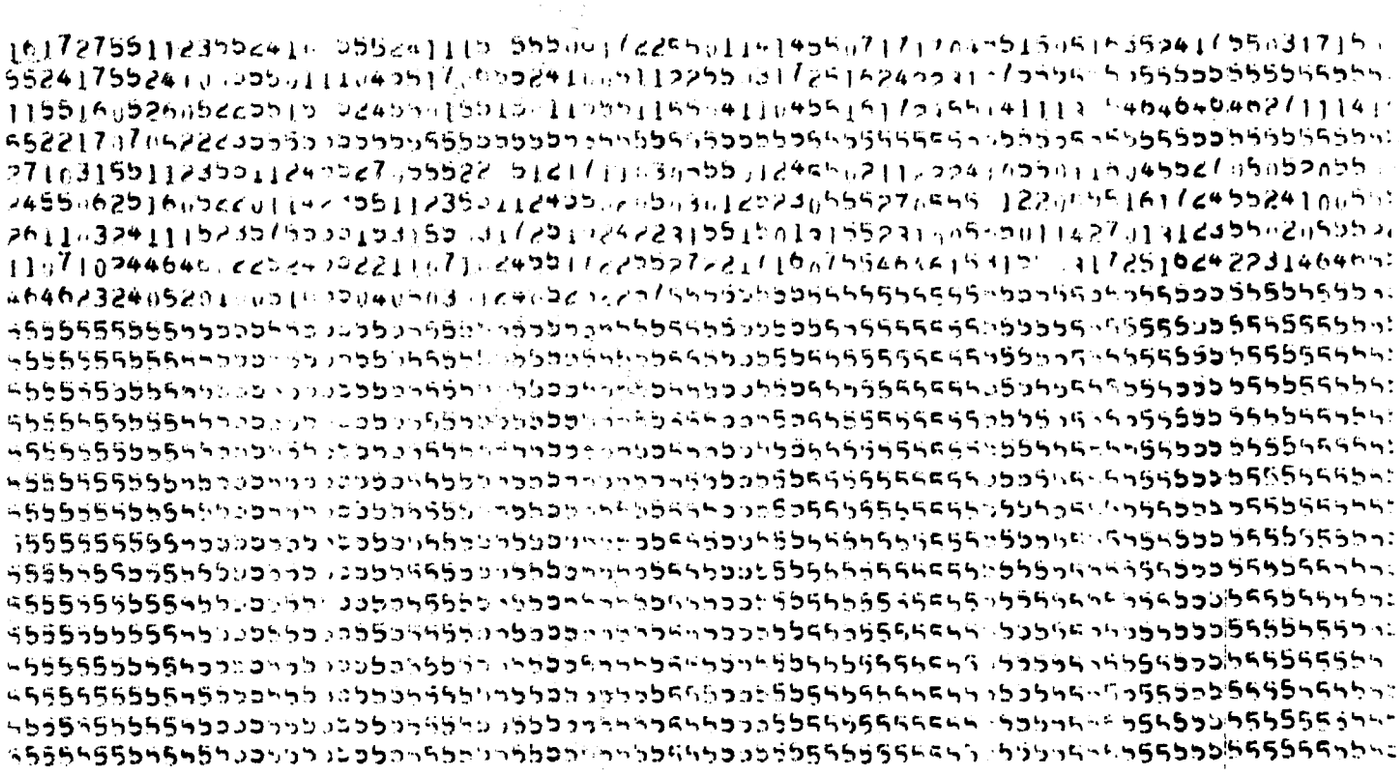
$$\text{Cost/bit} = \frac{.658\%}{5760} = 1.141 \times 10^{-4}\%$$

```
C     ****************************************************************
C     *                                                              *
C     *          VERNAM TECHNIQUE FOR INFORMATION STORAGE            *
C     *                    PERIODIC KEY                              *
C     *                                                              *
C     ****************************************************************
      PROGRAM ENCODE (INPUT, OUTPUT)
      DIMENSION M(100)
      DATA K/100*10 /
C     N = 96 FOR MAXIMUM MESSAGE OF UP TO 12 DATA CARDS AT 8
C     COMPUTER WORDS PER CARD
      READ 104, N
C     INPUT MESSAGE TO BE ENCODED
      READ 105, (M(I), I = 1, N)
C     ECHO PRINT MESSAGE
      PRINT 103
      PRINT 107, (M(I), I = 1, N)
C     PRINT INTERNAL OCTAL FORM OF MESSAGE
      PRINT 102
      PRINT 106, (M(I),I = 1 , N)
      KEY = 10HWAGAMAN
C     SCRAMBLE INTERNAL REPRESENTATION OF MESSAGE
      CALL SECOND (TIME)
      PRINT 108, TIME
      DO 10 I = 1, N
   10 M(I) = (M(I).AND.N.KEY).OR.(KEY.AND.N.M(I))
      CALL SECOND (TIME)
      PRINT 108, TIME
C     SCRAMBLED INTERNAL MESSAGE REPRESENTATION
      PRINT 100
      PRINT 107, (M(I), I = 1, N)
C     UNSCRAMBLE MESSAGE
      CALL SECOND (TIME)
      PRINT 108, TIME
      DO 20 I = 1, N
   20 M(I) = (M(I).AND.N.KEY).OR.(KEY.AND.N.M(I))
      CALL SECOND (TIME)
      PRINT 108, TIME
C     PRINT UNSCRAMBLED MESSAGE
      PRINT 101
      PRINT 107, (M(I), I = 1, N)
  100 FORMAT(12X,*SCRAMBLED INTERNAL REPRESENTATION*,/)
  101 FORMAT(12X,*RECOVERED TEXT AFTER DECODING*,/)
  102 FORMAT(12X,*INTERNAL DISPLAY CODE FORM OF MESSAGE*,/)
  103 FORMAT(12X,*MESSAGE TO BE DECODED*,/)
  104 FORMAT (I2)
  105 FORMAT (8A10)
  106 FORMAT (12X,8O20)
  107 FORMAT (12X,8A10)
  108 FORMAT ( 12X, F5.3)
      END
```

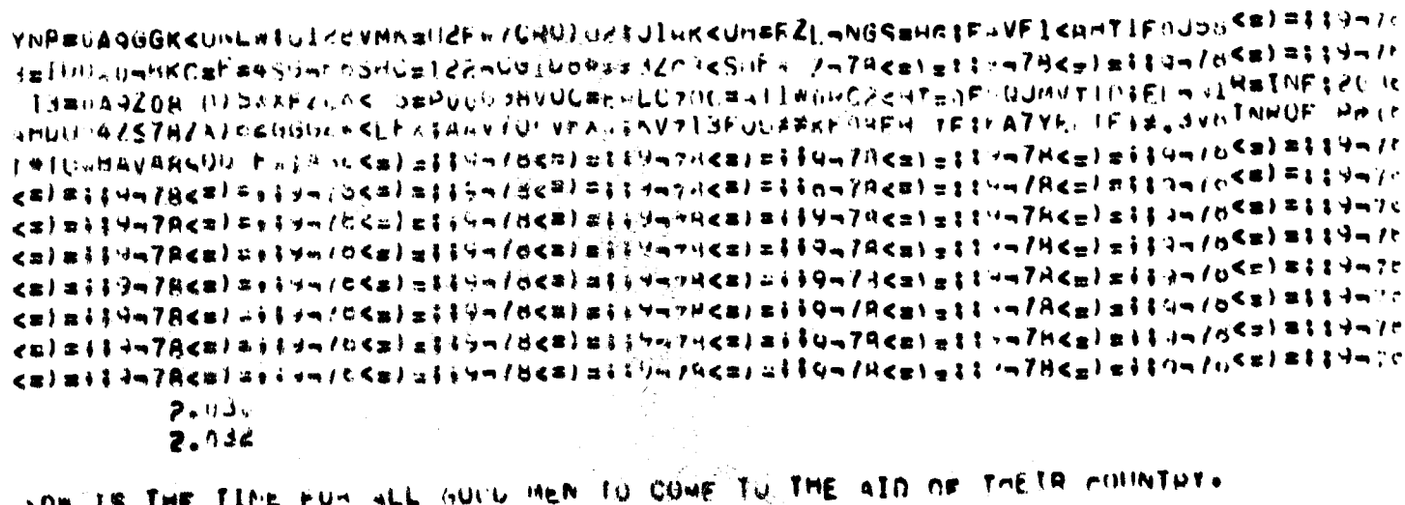
MESSAGE TO BE DECODED

NOW IS THE TIME FOR ALL GOOD MEN TO COME TO THE AID OF THEIR COUNTRY.
I NEVER MET A MAN I DID NOT LIKE----WILL ROGERS
WHY IS IT WE REJOICE AT BIRTH AND WEEP AT FUNERALS IS IT BECAUSE WE ARE VICTIMS. MY COUNTRY MAY SHE ALWAYS BE RIGHT--BUT RIGHT OR WRONG --MY COUNTRY--
--STEPHEN DECATUR.

INTERN L DISPLAY CODE OF MESS GE



2.013
2.014



2.035
2.036

NOW IS THE TIME FOR ALL GOOD MEN TO COME TO THE AID OF THEIR COUNTRY.
I NEVER MET A MAN I DID NOT LIKE---WILL ROGERS
WHY IS IT WE REJOICE AT BIRTH AND KEEP AT FUNERALS IS IT BECAUSE WE ARE NOT THE VICTIMS. MY COUNTRY MAY SHE ALWAYS BE RIGHT--BUT RIGHT OR WRONG--MY COUNTRY--STEPHEN DECATUR.

```
C     ************************************************************************
C     *                                                                      *
C     *       VERNAM ENCRYPTION METHOD FOR A ONE TIME KEY                    *
C     *                                                                      *
C     ************************************************************************
      PROGRAM ENCODE (INPUT, OUTPUT)
      DIMENSION M(100),KEY(100)
      DATA M/100*1H /
C     N = 32 THIS ENCRYPTION USES A ONE TIME KEY WHICH MUST BE AS
C     LONG OR LONGER THAN THE MESSAGE TO BE ENCODED
      READ 104, N
C     READ KEY TO BE USED TO BE USED IN ENCODING MESSAGE
      READ 105, (KEY(I), I = 1, N)
      PRINT 112
      PRINT 107, (KEY(I), I = 1, N)
C     INPUT MESSAGE TO BE ENCODED
      READ 105, (M(I), I = 1, N)
C     ECHO PRINT MESSAGE
      PRINT 103
      PRINT 107, (M(I), I = 1, N)
C     PRINT INTERNAL OCTAL FORM OF MESSAGE
      PRINT 109
      PRINT 106, (M(I), I = 1, N)
C     SCRAMBLE INTERNAL REPRESENTATION
      CALL SECOND (TIME)
      PRINT 108, TIME
      DO 10 I = 1, N
   10 M(I) = (M(I).A..N.KEY(I)).OR.(KEY(I).A..N.M(I))
      CALL SECOND (TIME)
      PRINT 108, TIME
C     SCRAMBLED INTERNAL MESSAGE REPRESENTATION
      PRINT 110
      PRINT 107, (M(I), I = 1, N)
C     UNSCRAMBLE MESSAGE
      CALL SECOND (TIME)
      PRINT 108, TIME
      DO 20 I = 1, N
   20 M(I) = (M(I).A..N.KEY(I)).OR.(KEY(I).A..N.M(I))
      CALL SECOND (TIME)
      PRINT 108, TIME
C     PRINT UNSCRAMBLED MESSAGE
      PRINT 111
      PRINT 107, (M(I), I = 1, N)
  103 FORMAT(//,10X,* MESSAGE TO BE ENCODED*,//)
  104 FORMAT (I2)
  105 FORMAT (8A10)
  106 FORMAT (1X, 4020)
  107 FORMAT (1X, 8A10)
  108 FORMAT ( 10X, F5.3)
  109 FORMAT (//,10X,* OCTAL FORM OF DISPLAY CODE*,//)
  110 FORMAT (//,10X,*SCRAMBLED TEXT OF ENCODED MESSAGE*,//)
  111 FORMAT (//,10X,*RECOVERED TEXT*,//)
  112 FORMAT (10X, *( ~ TIME KEY*,//)
      END
```

ONE TIME KEY

THIS IS THE KEY USED TO ENCODE THIS MESSAGE--IT CONSIOT OF 8 WORDS PER CARD OR 480 BITS PER CARD. SINCE THIS KEY MUST BE AS LONE AS THE MESSAGE WHICH IS 4 CARDS LONG. THIS KEY IS OF NECCESITY AS LONG OR LONGER. THAT MAKES THIS KEY 480 BITS PER CARD X 4 CARDS OR A TOTAL OF 1920 BITS IN ITS ENTIRIETY.

MESSAGE TO BE ENCODED

NOW IS THE TIME FOR ALL GOOD MEN TO COME TO THE AID OF THEIR COUNTRY.
THE EYES OF TEXAS ARE UPON YOU ALL THE LIVE LONG DAY. DO NOT
THINK YOU CAN ESCAPE THEM FROM NIGHT TILL EARLY IN THE MORN.
THE EYES OF TEXAS ARE UPON UPON YOU TILL GABRIEL BLOWS HIS HORN.

OCTAL FORM OF DISPLAY CODE

```
16172755112355241005552411150555061722550114145507171706455150516552417550317150 5
5524175524100555011104551706552410051122550317251624223157555555555555555555555555
5555555555552410055505310523551706552405300123550122055525201716553117255501141 4
5524100555141126055514171607550401315755041755161724555555555555555555555555555555
2410111613553117255503011655052303012005552410051555062217155516110710245524111 4
1455050122143155111655241005555151722165755555555555555555555555555555555555555555
2410055505310523551706552405300123550122055525201716552520171655311725552411141 4
5507010222110514550214172723551011235510172216575555555555555555555555555555555555
        1.996
        1.998
```

SCRAMBLED TEXT OF ENCODED MESSAGE

```
ZG3¬9Z¬↓1M/↓BH1 S1W(=XC BALK(H/Z+21 NJ3V=SJK]A0 RFJ¬F2↓↓GC9[ T GJG!I)! ,=![ 7!
]Nv .9 0/≥ K/P=2BB↓VQOP/=FM9F≥DK≠≠R ¬U6N/ ^V  FX  M1RS+A=T)¬WN)K7C+9,+ 9¬ ]
WI0JX U 0)=≡Z+L ,JU1 ∠0/R$$1JN,KZN1M UZ6 ?KF!CK EA8SMWR57FF,↓ 5=-/¬ ↓+9¬
4M1 Z<3¬.FR¬↓U2S¬,  A M≥P8,TRK? V2+=↓2CX=K=MT9Y/≡Y6M3G¬+ 2 A0A8)8↓9!9/↓≠8
        2.005
        2.007
```

RECOVERED TEXT

NOW IS THE TIME FOR ALL GOOD MEN TO COME TO THE AID OF THEIR COUNTRY.
THE EYES OF TEXAS ARE UPON YOU ALL THE LIVE LONG DAY. DO NOT
THINK YOU CAN ESCAPE THEM FROM NIGHT TILL EARLY IN THE MORN.
THE EYES OF TEXAS ARE UPON UPON YOU TILL GABRIEL BLOWS HIS HORN.

Case II (one-line key) 4 cards X 80 $\frac{\text{characters}}{\text{card}}$ X $\frac{6 \text{ bits}}{\text{character}}$ = 480 bits.

% overhead = $\frac{2 \times 10^{-3}}{.304}$ X 100% = .658% based on .304 sec of CP time.

Cost/bit = $\frac{.658\%}{480}$ = 1.370×10^{-3}%

Several things should be kept in mind at this point. When Case I is less expensive than Case II in terms of processing time, it is less secure. Secondly for a 100,000 word file in a 60 bit machine, overhead could be as high as 75%. Core requirements for the periodic key technique are minimal; however, the one-time key requires twice as much core as there is information to be stored, since the key is as long as the message. This objection can be minimized by using the key for many different files requiring addition storage only as great as the largest expected file. It is also possible to generate this key externally loading it only when it is required.

4.7 Evaluation of the Vignerian Tableau Method

The origin of this scheme is attributed to Blaise de Vignere and is probably the most famous encryption method of all time. The system goes back to 1585 when Vignere published a book on cryptography. The method was forgotten, revived in the 19th Century and finally buried at the turn of the century. The system is breakable, especially if the original word divisions are kept. While not suitable for diplomatic and military purposes it is a reasonably secure system and lends itself to implementation in software. A similar system, which we will call the

modulo arithmetic scheme,lends itself to hardware implementation plus several other refinements.

The Vignere tableau,as it is called,is a polyalphabetic system (two or more cipher alphabets are employed in some prearranged pattern). The system essentially consists of an alphabetic table in which the first letter of successive rows have been shifted end around one letter from the previous. Indeed,it is not necessary that they be shifted just one letter. As long as the shift relation between successive rows is known any arrangement is suitable. An abbreviated example is given below:

	A	B	C	D	E	F	G
A	a	b	c	d	e	f	g
B	b	c	d	e	f	g	a
C	c	d	e	f	g	a	b
D	d	e	f	g	a	b	c
E	e	f	g	a	b	c	d
F	f	g	a	b	c	d	e
G	g	a	b	c	d	e	f

Vignere tableau

To encipher a message with the tableau, a key is chosen, say BAD. The message to be encoded will be CABBAGE. A normal alphabet at the top is used for the plaintext and another normal alphabet vertically is used for the key. The key is written above the message repeating it as much as is necessary to cover the message. The intersecting rows

and columns of the two letters in the tableau represent the transposition.

Example:

Key: B A D B A D B

Plain: C A B B A G E

Cipher: d a e c a c e

To decipher, one enters from the side with the lead key letter, goes across until he finds the letter, and then up until he reaches the normal alphabet at the top. The scheme can be improved by the use of random keys, one time keys, variable shifting of the alphabet, etc.

The program written to evaluate the technique uses a tableau based around CDC 6600 display code which is defined conveniently to expedite processing of the tableau. Each character is represented by an octal number from 1 octal to 77 octal, A being 1 octal B being 2 octal, etc. The tableau consists of a 63X63 matrix of 3969 characters; however, it is not necessary to store the complete matrix as not all of it is used in any given message. It is only necessary to generate those rows starting with the key letters and to use these rows to encode the message. This cuts down on the core memory required to load the program. The Vignere' program requires more core than the Vernam system, as a transposition table must be stored; whereas, only a mathematical algorithm is necessary in the Vernam system.

Time studies of the technique show an eightfold increase in encode/decode time as well as an increase in central processor time.

Exp. Encode

$$68 \text{ character} \times \frac{60 \text{ bits}}{\text{character}} = 4080 \text{ bits}$$

$$\% \text{ overhead} = \frac{1.010^{-3} \text{ sec}}{.392} \times 100\% = .255\% \text{ based on } .392$$

of CP time.

$$\text{Cost/bit} = \frac{.255\%}{4080} = 6.25 \times 10^{-5} \%$$

Decode

$$\% \text{ overhead} = \frac{13 \times 10^{-3}}{.392} \times 100\% = 3.32\% \text{ based on } .392 \text{ sec}$$

of CP time.

$$\text{Cost/bit} = \frac{3.32\%}{4080} = 8.12 \times 10^{-4} \%$$

Decoding time was greater than encoding time for the message sample. More memory was used in this program than was necessary since only six bits of sixty were used in each word, as each character occupied a whole word instead of the usual ten characters per word. This facilitated processing by cutting down the time required, but increased memory requirements tenfold. These costs may not necessarily be representative since some installations charge more for central processing time than they do for extra memory during execution. Another interesting point is that while average encode/decode time is greater for the Vignere method, cost/bit is less.

```
      C     ********************************************************************
      C     *                                                                  *
      C     *          ENCRYPTION BY VIGNERIAN TABLEAU                         *
      C     *                                                                  *
      C     ********************************************************************
            PROGRAM CIPHER (INPUT, OUTPUT)
000002      COMMON/1/ TAB (44,64), KEY (64), NX (100), NY (100), NLTKEY,NLTMSG
           1,NZ(100)
000002      INTEGER TAB
      C     READ NUMBER OF LETTERS IN KEY AND IN MESSAGE
000002      READ 100, NLTKEY, NLTMSG,NLTAB
000014      PRINT 100, NLTKEY, NLTMSG, NLTAB
      C     READ KEY
000026      READ 101, (KEY(I), I = 1, NLTKEY)
      C     READ MESSAGE
000035      READ 101, (NX(I), I =1, NLTMSG)
      C     ECHO PRINT KEY
000044      PRINT 102, (KEY(I), I = 1, NLTKEY)
      C     PRINT MESSAGE
000053      PRINT 106
000057      PRINT 105, (NX(I), I = 1, NLTMSG)
      C     CREATE VIGNERIAN TABLEAU
000066      DO 10  J =1, NLTKEY
000070      ISTART = KEY(J)
000072      DO 10 I = 1, NLTAB
000101      TAB(J,I) = ISTART
000102      IF(ISTART.EQ.NLTAB) ISTART = 0
000103   10 ISTART = ISTART + 1
000110      PRINT 109
000113      PRINT 103, ((TAB(J,I),I =1, NLTAB), J = 1, NLTKEY)
      C     ENCODE MESSAGE
000133      CALL SECOND (TIME)
000135      PRINT 104, TIME
000143      DO 20 I =1, NLTMSG, NLTKEY
000145      DO 20 J = 1, NLTKEY
000152      M = I + J - 1
000154      IF(M.GT.NLTMSG) GO TO 30
000157      K = NX(M)
000163      NY(M) = TAB(J, K)
000167   20 CONTINUE
000167   30 CONTINUE
000167      CALL SECOND (TIME)
000171      PRINT 104, TIME
      C     PRINT CIPHER MESSAGE
000177      PRINT 107
000203      PRINT 105, (NY(I), I = 1, NLTMSG)
      C     DECODE CIPHER MESSAGE
000212      CALL SECOND (TIME)
000214      PRINT 104, TIME
000222      DO 50 K=1, NLTMSG,NLTKEY
000224      DO 50 J=1, NLTKEY
000225      M = K + J - 1
000227      IF(M.GT.NLTMSG) GO TO 60
000232      DO 40 I = 1, NLTAB
000233      IF(TAB(J,I).NE.NY(M)) GO TO 40
000240      NZ(M) = I
000241      GO TO 50
000241   40 CONTINUE
```

```
 50 CONTINUE
 60 CONTINUE
    CALL SECOND (TIME)
    PRINT 104, TIME
    PRINT 110
    PRINT 105, (NZ(I), I = 1, NLTMSG)
100 FORMAT (I2,I3,I3)
101 FORMAT (A0R1)
102 FORMAT(1X, 6HKEY = 8A10,/)
103 FORMAT (1X,63R1,/)
104 FORMAT(10X, F5.3)
105 FORMAT (1X, 40R2,/)
106 FORMAT(1X,*MESSAGE *,/)
107 FORMAT(1X,*CIPHER MESSAGE*,/)
109 FORMAT (/,1X, *VIGNERIAN TABLEAU*,/)
110 FORMAT (1X, *DECODED MESSAGE*,/)
    END
```

```
KEY =          T          Y          P          E
MESSAGE

 N O W   I S   T H E   T I M E   F O R   A L L   G O O D   M E N   T O   C O M E

   T O   T H E   A I D   O F   T H E I R   C O U N T R Y

VIGNERIAN TABLEAU

TUVWXYZ0123456789+-*/()$= ,.≡[]:≠≠∨∧↑↓<>≤≥¬!ABCDEFGHIJKLMNOPQRS

YZ0123456789+-*/()$= ,.≡[]:≠≠∨∧↑↓<>≤≥¬!ABCDEFGHIJKLMNOPQRSTUVWX

PQRSTUVWXYZ0123456789+-*/()$= ,.≡[]:≠≠∨∧↑↓<>≤≥¬!ABCDEFGHIJKLMNO

EFGHIJKLMNOPQRSTUVWXYZ0123456789+-*/()$= ,.≡[]:≠≠∨∧↑↓<>≤≥¬!ABCD

         2.070
         2.071
CIPHER MESSAGE

 6 * - [ 1 $ ≤ X 0 2 ≤ X 1 + T [ Y * 6 [ T 9 0 [ Z * 3 H A + T R A = 3 [ V * 1 I

 A = 3 [ * 5 T [ T 6 S [ 7 3 ≤ X 0 2 X V A 0 3 Y 6 = 6 2
         2.078
         2.091
DECODED MESSAGE

 N O W   I S   T H E   T I M E   F O R   A L L   G O O D   M E N   T O   C O M E

   T O   T H E   A I D   O F   T H E I R   C O U N T R Y
```

4.8 Modulo Encryption

The Modulo method is a modification of the Vignerian method. The basic difference is that no table is generated for the encryption decryption cycle , saving some core memory. However, time required for encode/decode is greater. The scheme lends itself more to hardware implementation rather than to software like the Vignere' scheme. A hardware unit could probably be built that would perform the function much faster.

The theory of encryption is simply to add the value of the display code of the key letter and the text to yield the cipher letter. This is effectively what is done in the Vignere' technique as programmed previously. Modulo 63 arithmetic is required, since there are 63 display code characters in the CDC 6600. Since the last 10 - 15 characters in display code are specialized and would not appear in most texts,it would be possible to vary the modulus by 10 - 15 without losing any information. This would add some security over the Vignere' scheme. Another arbitrary number can be added in to change the output cipher. This number can be changed arbitrarily by the programmer before or during the program adding to the security of the scheme. Those two factors make it more secure than the Vignerian scheme. The encode functions are:

```
ISUM I  =  NX(M) + KEY(I) + IFOOLU - 1

NY(M)   =  MOD(ISUMI, 63)
```

Here IFOOLU is the complicating factor that may be introduced by the programmer. To decode the reverse process is:

```
      I SUM2  =  NY(M) - KEY(J) - IFOOLU+1
50    IF (ISUM2.LE.0) ISUM2 = ISUM2 + 63
      IF (ISUM2.LE.0) GO to 50
      NZ(M) = MOD(ISUM2, 63)
```

An additional check is required to check to see if ISUM 2 is negative. If so, 63 is added repeatedly until ISUM2 is positive. This accounts for the increase in time required for decoding.

Timing studies for the modulo method reveal the following for a 480 character, 28800 bit message.

Encode

$$\% \text{ overhead} = \frac{8 \times 10^{-3}}{.412} \times 100\% = 1.94\% \text{ based on}$$

.412 sec CP Time.

$$\text{Cost/bit} = \frac{1.94\%}{28800} = 6.75 \times 10^{-5}\%$$

Decode

$$\% \text{ overhead} = \frac{18 \times 10^{-3}}{.412} \times 100\% = 4.38\%$$

$$\text{Cost/bit} = \frac{4.38\%}{28800} = 1.52 \times 10^{-4}\%$$

It should be noted that,in the Vignerian method and the modulo method,the extra memory required brought the overhead per bit down. This can be misleading; for,as this program is written, a full 60 bit word is required for each character encoded. Thus there is a tradeoff between memory and time as brought out by the Modulo program. The word packing program does the same thing as the Modulo program except that by

word packing, memory requirements for the encoded message are cut by a factor of ten. However, the saving in memory is lost by a significant increase in encode time and decode time required. Overhead increased from 1.94% on .412 sec. of CP time to 13.02% on .461 sec. of CP time. The cost/bit shot from $1.52X10^{-4}$% to $2.71X10^{-2}$%. This points out the disadvantage of performing encoding and decoding in software.

```
C     ****************************************************************
C     *                                                              *
C     *                      MODULO ENCRYPTION                       *
C     *                                                              *
C     ****************************************************************
      PROGRAM CIPHER (INPUT, OUTPUT)
      DIMENSION KEY(20), NX(500), NY(500), NZ(500)
C     READ NUMBER OF LETTERS IN KEY AND IN MESSAGE
      READ 100, NLTKEY, NLTMSG,NLTAB
C     READ KEY
      READ 101, (KEY(I), I = 1, NLTKEY)
C     READ MESSAGE
      READ 101, (NX(I), I =1, NLTMSG)
C     ECHO PRINT KEY
      PRINT 102, (KEY(I), I = 1, NLTKEY)
C     PRINT MESSAGE
      PRINT 106
      PRINT 105, (NX(I), I = 1, NLTMSG)
      IFOOLU = 43
C     ENCODE MESSAGE
      CALL SECOND (TIME)
      PRINT 104, TIME
      DO 10 I = 1, NLTMSG, NLTKEY
      DO 10 J = 1, NLTKEY
      M = I + J - 1
      IF(M.GT.NLTMSG) GO TO 20
      ISUM1 = NX(M) + KEY(J) + IFOOLU - 1
      NY(M) = MOD(ISUM1, NLTAB)
   10 CONTINUE
   20 CONTINUE
      CALL SECOND (TIME)
      PRINT 104, TIME
      PRINT 107
      PRINT 105, (NY(I), I = 1, NLTMSG)
C     DECODE CIPHER MESSAGE
      CALL SECOND (TIME)
      PRINT 104, TIME
      DO 30 K = 1, NLTMSG, NLTKEY
      DO 30 J = 1, NLTKEY
      M= K + J - 1
      IF(M.GT.NLTMSG) GO TO 40
      ISUM2 = NY(M) - KEY(J) - IFOOLU + 1
   50 IF(ISUM2.LE.0) ISUM2 = ISUM2 + NLTAB
      IF(ISUM2.LE.0) GO TO 50
      NZ(M) = MOD(ISUM2, NLTAB)
   30 CONTINUE
   40 CONTINUE
      CALL SECOND (TIME)
      PRINT 104, TIME
      PRINT 110
      PRINT 105, (NZ(I), I = 1, NLTMSG)
  100 FORMAT (I2,I3,I3)
  101 FORMAT (80R1)
  102 FORMAT(1X, 6HKEY = 20R4,/)
  104 FORMAT(10X, F5.3)
  105 FORMAT (1X, 65R1,/)
  106 FORMAT(1X,*MESSAGE *,/)
  107 FORMAT(1X,*CIPHER MESSAGE*,/)
```

```
000227      110 FORMAT (1X, *DECODED MESSAGE*,/)
000227          END
005600
KEY =    K   I   T   T   Y       W   A   W   K
MESSAGE

      NAME:  JOHN Q. PUBLIC          AGE:    32  INCOME:  6500

                    CREDIT STATUS: GOOD- REFUSED TO PAY LATE CHA

RGES ON LOAN                                          FROM EASY-GO A

PPLIANCE STORE ON 2 JAN 1965                          ARREST RECORD:

 ARRESTED -MINOR IN POSSESSION 12 MAY 1957

 POLITICAL PARTY: REPUBLICAN   HEALTH: FAIR- CONTACTED

               INFECTITIOUS HEPATITIOUS-- JULY 1957  COMMENTS: N

ONE
         2.044
         2.052
CIPHER MESSAGE

86==[FA=O<(6=IS5AYS+8DTAP6+Y.886==[F5=I<(6==[VPY.¬DVNLYL5Y8VQO==[

F5Y.886==[F5Y.886==[F+2G+¬H=RXYGA!!(8<NNH¬5≥G>KQDC[=RYRV06K X25.JV

H<OR[*AYNFV8==[F5Y.886=[F5Y.886==[F5Y.886==[F5Y.88+QNOFA=UO1<N=E

/CAKVDVU=W=8≥G8ER=1[7!+.RZU4=[F5Y.886==[F5Y.88A==[F!≥T<IH=QIQR≥F(

8#QQIS(>ΞFA1AHMS)5≠P8FCHRISF≠QDAP]=QYLY3ZVV==[F5Y.886==[F5Y.8A6==[

FC<N¬J<8 PFC=TJO*=QI/H N¬↑≠M=[F<ΞCRJ>]=JY>≥/8+CMSE0GΞFA86==[F5Y.8

86==[F5Y.8A6==[F>+H<↑HMSM+H¬.≥<D SM=><WI1Z=IY9LY3ZVV==Q* +GDJ8]=R

*AΞ.886==[F5Y.886==[F5Y.8
         2.076
         2.094
DECODED MESSAGE

      NAME:  JOHN Q. PUBLIC          AGE:    32  INCOME:  6500

                    CREDIT STATUS: GOOD- REFUSED TO PAY LATE CHA

RGES ON LOAN                                          FROM EASY-GO A

PPLIANCE STORE ON 2 JAN 1965                          ARREST RECORD:

 ARRESTED -MINOR IN POSSESSION 12 MAY 1957

 POLITICAL PARTY: REPUBLICAN   HEALTH: FAIR- CONTACTED

               INFECTITIOUS HEPATITIOUS-- JULY 1957  COMMENTS: N

ONE
```

```
C     ********************************************************************
C     *                                                                  *
C     *              MODULO ENCRYPTION WITH WORD PACKING                 *
C     *                                                                  *
C     ********************************************************************
      PROGRAM MODULO (INPUT,OUTPUT)
      COMMON /1/  KEY(20),NX(500),NY(500),NZ(500),
     1J,IFOOLU,NWORDS,NLTKEY,NLTAB,INDEX
C     READ NUMBER OF LETTERS IN KEY AND NUMBER OF WORDS IN MESSAGE
      READ 100, NLTKEY,NWORDS,NLTAB
C     READ KEY
      READ 101, (KEY(I), I = 1, NLTKEY)
C     READ MESSAGE
      READ 102, (NX(I), I = 1, NWORDS)
C     PRINT KEY
      PRINT 103, (KEY(I), I = 1, NLTKEY)
C     PRINT MESSAGE
      PRINT 104
      PRINT 105, (NX(I), I = 1, NWORDS)
      IFOOLU = 43
      MASK = 77000000000000000008
C     ENCODE MESSAGE
      CALL SECOND (TIME)
      PRINT 106, TIME
      INDEX = 1
      DO 20 J = 1, NWORDS
      IHOLD = NX(J)
      NY(J) = 0
      DO 10 I = 1, 10
      ITEMP= IHOLD.AND.MASK
      ITEMP = LSHIFT(ITEMP,6)
      CALL ENCODE(ITEMP)
      NY(J) = LSHIFT(NY(J),6)
      NY(J) = NY(J).OR.ITEMP
      IHOLD = LSHIFT(IHOLD,6)
   10 CONTINUE
   20 CONTINUE
      CALL SECOND (TIME)
      PRINT 106, TIME
      PRINT 107
      PRINT 105, (NY(I), I = 1, NWORDS)
C     DECODE CIPHER MESSAGE
      CALL SECOND (TIME)
      PRINT 106, TIME
      INDEX = 1
      DO 40 J = 1, NWORDS
      IHOLD = NY(J)
      NZ(J) = 0
      DO 30 I = 1, 10
      ITEMP = IHOLD.AND.MASK
      ITEMP = LSHIFT(ITEMP,6)
      CALL DECODE(ITEMP)
      NZ(J) = LSHIFT(NZ(J),6)
      NZ(J) = NZ(J).OR.ITEMP
      IHOLD = LSHIFT(IHOLD,6)
   30 CONTINUE
   40 CONTINUE
      CALL SECOND (TIME)
```

```
      PRINT 106, TIME
      PRINT 108
      PRINT 105, (NZ(I), I = 1, NWORDS)
  100 FORMAT(I2,I3,I3)
  101 FORMAT(80R1)
  102 FORMAT(8R10)
  103 FORMAT(12X,6HKEY = 20R4,/)
  104 FORMAT(12X,*MESSAGE*,/)
  105 FORMAT(12X,8A10,/)
  106 FORMAT(12X,F5.3)
  107 FORMAT(12X,*CIPHER MESSAGE*,/)
  108 FORMAT(12X,*DECODED MESSAGE*,/)
      END
```

```
      SUBROUTINE ENCODE(ITEMP)
      COMMON /1/  KEY(20),NX(500),NY(500),NZ(500),
     1J,IFOOLU,NWORDS,NLTKEY,NLTAB,INDEX
      IF(INDEX.GT.NLTKEY) INDEX = 1
      ISTART = KEY(INDEX)
      ITEMP = ITEMP + ISTART + IFOOLU - 1
      ITEMP = MOD(ITEMP, NLTAB)
      INDEX = INDEX + 1
      RETURN
      END
```

```
      SUBROUTINE DECODE(ITEMP)
      COMMON /1/  KEY(20),NX(500),NY(500),NZ(500),
     1J,IFOOLU,NWORDS,NLTKEY,NLTAB,INDEX
      IF(INDEX.GT.NLTKEY) INDEX = 1
      ISTART = KEY(INDEX)
      ITEMP = ITEMP - ISTART - IFOOLU + 1
   50 IF(ITEMP.LE.0) ITEMP=ITEMP + NLTAB
      IF(ITEMP.LE.0) GO TO 50
      ITEMP = MOD(ITEMP, NLTAB)
      INDEX = INDEX + 1
      RETURN
      END
```

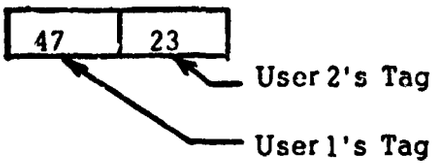
```
KEY =    K    I    T    T    Y         H    A    W    K
MESSAGE

NOW IS THE TIME FOR ALL GOOD MEN TO COME TO THE AID OF THEIR COUNTRY.

2.432
2.438
CIPHER MESSAGE

DCV=MS5 J<BHHLIF+<TBV K=K*R..C<B=SSF-<O<RHN=X5AYC++6NF(=<=KHRVHT-=FF(H)<=
=(F5Y.,
2.441
2.448
DECODED MESSAGE

NOW IS THE TIME FOR ALL GOOD MEN TO COME TO THE AID OF THEIR COUNTRY.
```

```
  UNIVERSITY OF TEXAS 6600    UT 1
UBTE249.  200TE READ.
UBTE249. MODULO,,7,47000,17,PHAU0007,GARRISON.
UBTE249. RUN(SX)
UBTE249.   CTIME 000.923 SEC.  RUN LEVEL 60+
UBTE249. LGO.
UBTE249.  LOADER UNUSED STORAGE 032030.
UBTE249.  END - MODULO
UBTE249. CP  000.461 SEC.
UBTE249. PP  003.171 SEC.
UBTE249. TM  002.673 SEC.        3 (OCTAL)
```

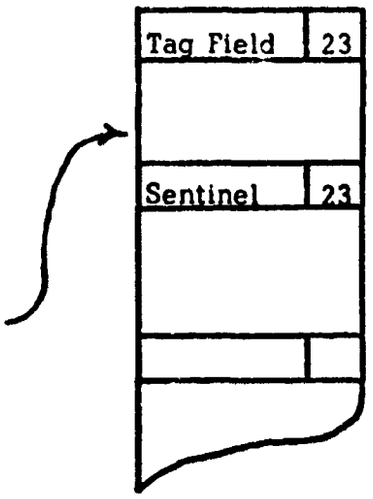
4.9 Other Techniques to Insure Static Protection

In addition to cryptographic methods there are other techniques possible. It would be possible to store supervisory assigned tags in each data word or block. If accessed, this tags alert the computer of an illegal jump into a page or block of memory. Normally all accesses to the memory space must be made through an entry point. If anyone jumps into the memory space in an execution mode and avoids the normal entry point, he will be detected. One possible way to implement this is to load a header word on each page with a supervisory assigned tag, which serves as the entry point. The header word then arms the sentinal word with the value of the tag given it by the supervisor. The sentinel words are placed at various points in the page. Upon reaching one of the sentinel or check points the supervisor would request that the user identify himself. If the user's supervisory assigned tag does not agree, he would be put into a loop or some delaying routine, until he could be traced, or his accesses would be aborted. This technique provides protection against parties entering an unauthorized data or command space without going through the proper entry points. Encryption does not protect against illegal entry but attempts to render useless, information so obtained.

Sentinel Technique:



Header Word User 2's normal entry point for this block or page

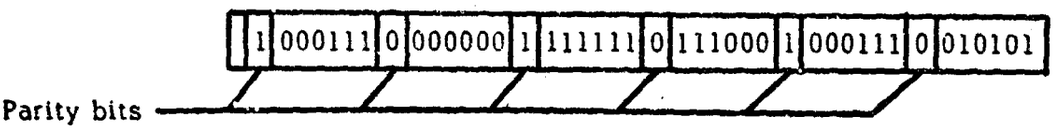


If user attempts entry here he will be detected upon reaching sentinel

Another possibility, less wasteful of bits, is to allow the user to assign the parity scheme to be used within each data word within his file. Thus, anyone using the program who does not know the scheme will be detected as a non-user since parity errors will result. To demonstrate the point consider a 48 bit data word of six 8-bit bytes. Attached to each byte is an extra bit or parity bit. This bit is always such that the number of 1's is either odd or even in each byte. Thus, including bits, there are 54 bits in each data word. If an even parity check is used for the word, the scheme would require that all six bytes be even parity. It would thus be possible to specify the parity scheme within each word. This parity scheme for each word could be varied from program to program. This parity scheme could be specified by a file owner in advance and thus would be unknown to anyone else.

This scheme could be very easily implemented and would not require any special equipment.

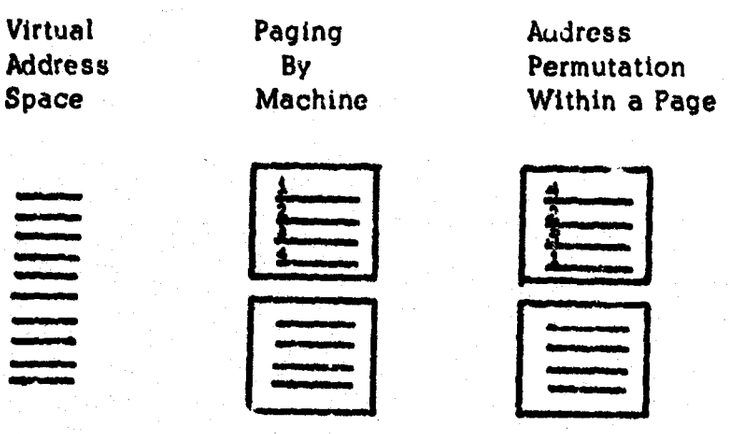
Parity bit scheme within a word:



Address Permutation Within a Page:

This method extends the virtual address scheme described earlier to permutation of addresses within a page. This scheme thus scrambles addresses within a page so as to make sequential execution or copying meaningless. The scrambler could operate effectively without the user's knowledge, requiring only that the unscrambling be effected when the proper user is using the file. This requires storing the transformation key for quick access.

Inter-Page Address Permutation



Exp: Address Permutation Key: 4231

Bit Permutation Within A Word

Bit permutation is similar to the Vernam system except we do not perform a complementative transformation but, rather rearrange bits within a word according to some prearranged key. The example shows how this might appear.

1 2 3 4 5 6 7 8 9	Data word for manipulation
3 6 9 7 4 2 1 8 5	Word in Storage

[Key: (3,6 9 7 4 2 1 8 5)]

The permutation technique is best performed at the bit level, rather than at the character level or byte level, since it would be easy to unscramble characters making up a common occuring instruction. Consider the permutation A X C I which would be recognizable as C I A X. At the bit level this situation would not be so obvious. When the Vernam technique is combined with permutations of bits, the possible combinations get large quickly.

Consider N words per page $(N = 2^{10})$

No. of possible permutations = $(n!)$

No. of possible complementations in a word $\Big\} = 2^n$

Of course, associated with this technique are the needs for simple permutation and depermutation algorithms and a simple method for permuting 2^{10} numbers.

COMPARISON OF TECHNIQUES

SCHEME

Method	Vernam	Vernam	Vignere'	Modulo
Size of key	Periodic key Variable, as discussed 60 bits	One-time key Variable-as long as the message	Variable, as discussed 24 bits	Variable, as discussed 60 bits
Additional Memory Required	Only that required for message	Twice that required for message-key as long as message	Some storage required for tableau as many words as letters in text	As many words as letters in text.
Decoding Overhead	Low-encode decode algorithm identical-fast if key is known	Low-encode decode algorithm identical-fast if key is known	Fair - limited number of trials if key is known	Fair - limited number of trials if key is known
Security-difficulty level in breaking code	Very good if long random key used	Unbreakable in theory unless key is known	Good	Good
Cost/bit for decode-encode (Average)	1.141×10^{-4} %	13.7×10^{-4} %	4.373×10^{-4} %	1.098×10^{-4} %

PART V

EXPLANATION OF COMPONENTS AND DATA FLOW OF A SECURE DATA BANK

5.1 Privacy Recognizer and Classifier

The operation of this phase depends upon the policy established for system operations. These policies would be highly organized and closely regulated, Data normally comes into the system through a recognizer and classifier, (Figure 10) where it is determined if information is of a sensitive nature. If not sensitive, it is sent to the access control portion of the system for storage. If it is sensitive, check is made to allow them to determine for themselves if the information is valid and if it can be used. Meanwhile, the data in question is stored very securely. If storage permission is obtained, a security classification is suggested, and the information is retrieved for classification and then submitted to the system. If permission to store the data is denied the information in equestion is purged from all files. This is verified by a search made through the system for all copies of the data, which include those stored in the backup files, change files, etc. All copies are deleted and the indiv.. is notified.

5.2 Access Control and Management

This unit is responsible for implementing the access control and file manipulation restrictions outlined in the section on protection mechanisms.

Secure Data Bank Model

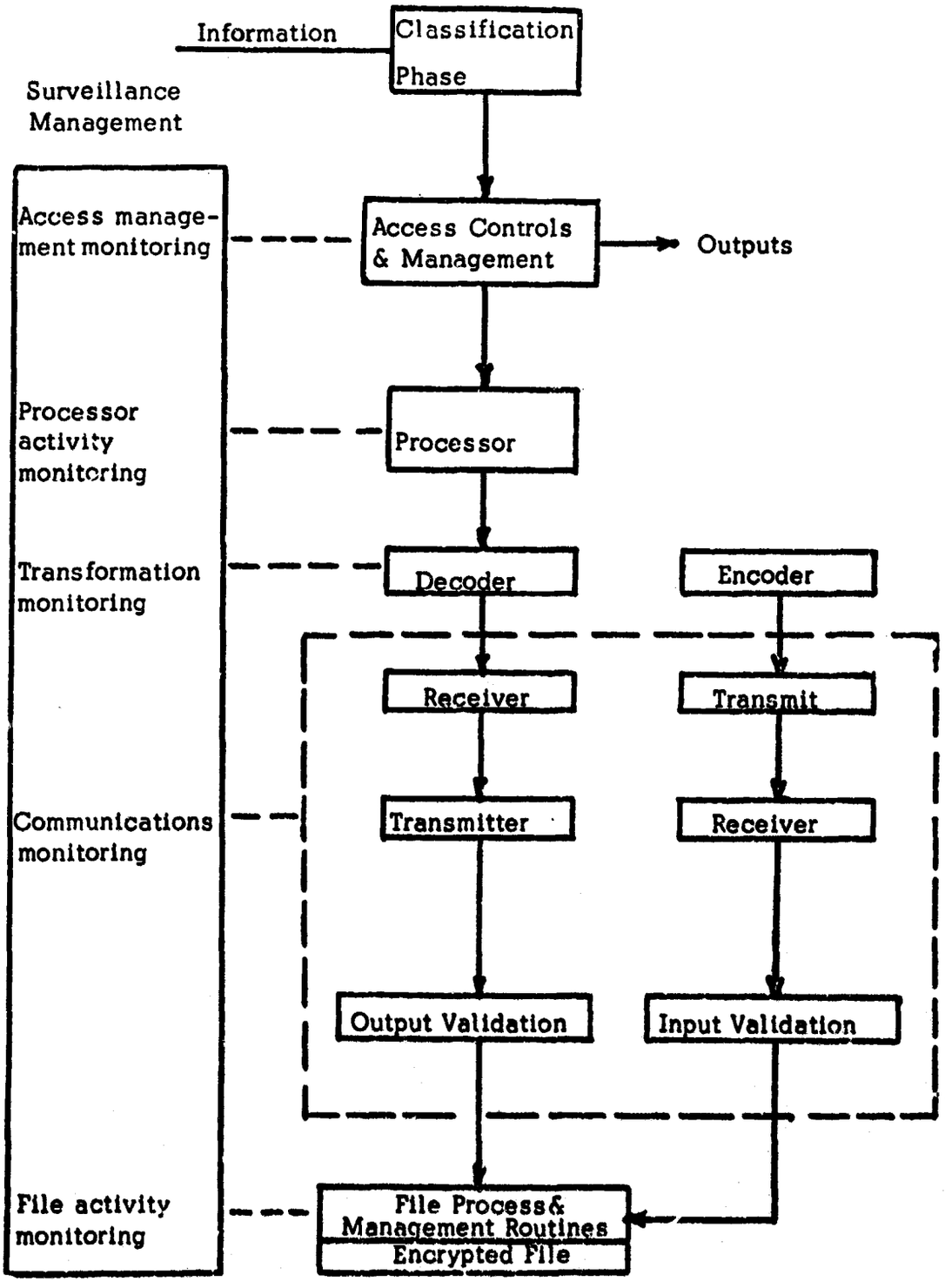


Figure 9

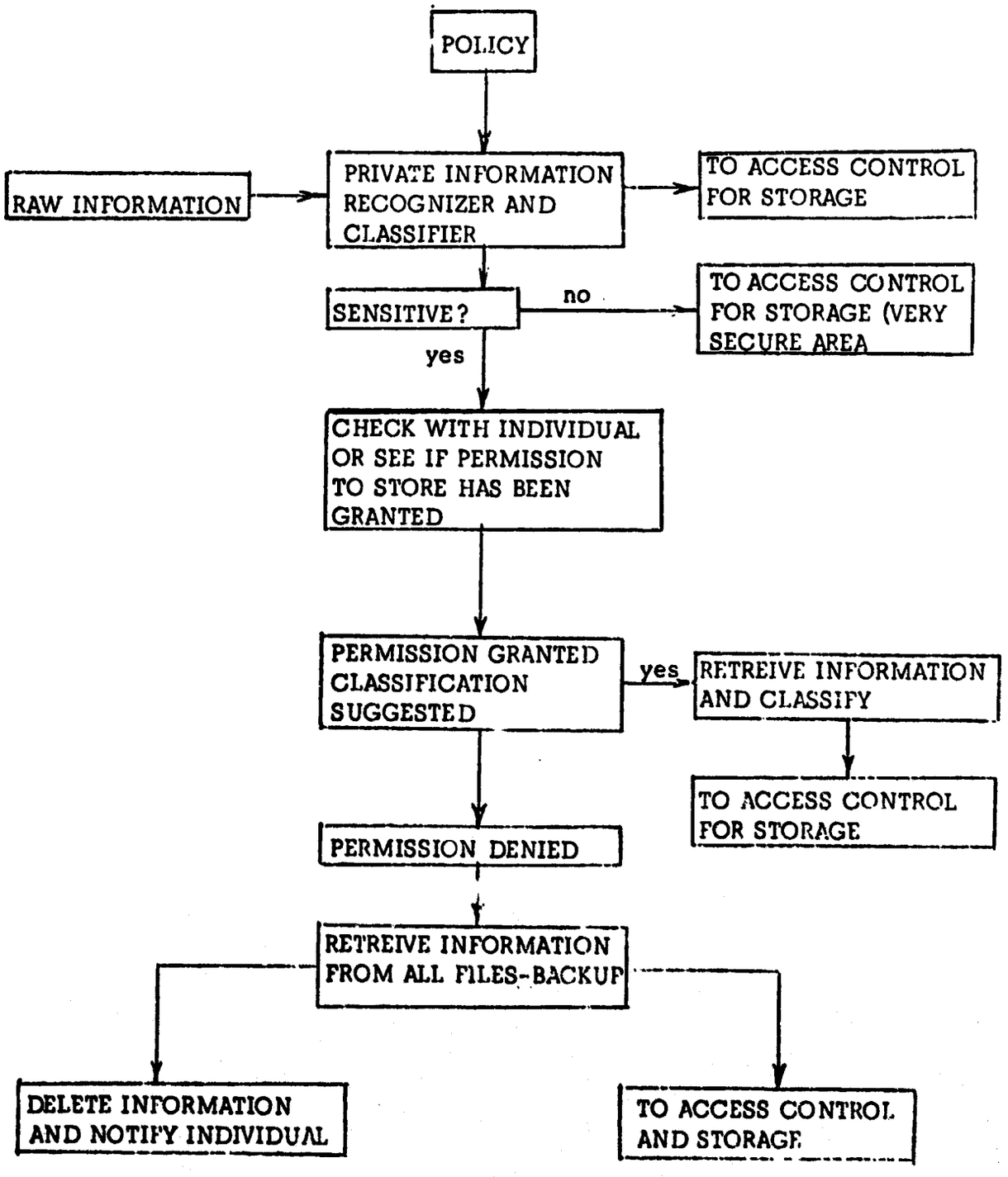


Figure 10 - INFORMATION CLASSIFICATION PHASE

A logical choice of what access control capabilities a data bank would require would be implemented and inserted at this point.

5.3 The Processor

The processor functions in a physically secure environment so as to insure its integrity for execution. The processor also initiates action by the encoder.

5.4 The Encoder

The encoder performs some logical transformation on the data before it is stored in the file. The topic of privacy transformation and encryption was discussed previously.

5.5 Communications Loop

The communications loop represents a portion of the system that would be required if the processor and mass storage file are physically separated. If they are in the same room, the loop is obviously not required. The encoder is still desirable to provide protection if access management and file restrictions fail to protect the data. Input and output validation requires certification of input and output data streams. This implies rechecking authorization by use of an alternate communications line or by retransmission on an alternate loop.

5.6 File Processor and Management Routine.

File processor and management routines are visualized as merely cataloging and filing routines. These routines manage use of storage

space and fetch and store encrypted records. The file processor functions independently of the main processor. The main processor addresses the file processor and cannot bypass it to fetch records. The file processor is charged with file storage management and protection or protection of static storage. This storage protection is largely concerned with protecting the files from security violations and system failure. In the case of a delete or change request this system insures that all copies of record regardless of the file are changed or destroyed. Depending upon the value of the stored data, the data may be given several types of protection. Possible options on the storage schemes would be

1. No protection
2. Main File, Backup file
3. Main File, Backup file, and change file for recording changes during transactions.
4. Encrypted File
5. Encrypted File and options, 3

If all the information is not critical, then we may not require protection at all. If it is lost, it can easily be replaced at a later time. The protection level can be increased by providing backups and a change file capability so that insurance against physical destruction of a file is available. The encrypted file and combination of encryption, backup and

change file offer the highest level of protection but require extensive overhead.

5.7 Decoder

The decoder performs the reverse transformation on data enabling the processor to operate on the data fetched from storage. The inverse of the key used to encode the data is used to decode the data.

5.8 System Monitor

All flow of data through the system is monitored by the various monitoring functions indicated. Activities of the access control, processor, encoder-decoder, communications loop and file are monitored whenever data is being moved within the system.

PART 6

CURRENT STATUS OF SECURITY ORIENTED SYSTEMS

Several systems have been designed to increase security of stored data. Multics, RUSH, Cambridge, ADEPT-50 and the BCC Model I are all working systems. While these systems or any system is not the ultimate, they represent a significant attempt toward implementation of a protected system. The Cambridge University File, RUSH, BCC Model I and the Adept-50 are described as they represent interesting approaches to system organization. The basic concept behind the Multics system is discussed and commented upon.

6.1 Cambridge University File Protection System

This system is based around a Titan computer. Associated with each file is a privacy arrangement. File activities are divided into two classes:

> Class I: This class permits execution, read, delete, update, and change of status of a file.
>
> Class II: This class permits several activities of which two are file creation, modification, or deletion of a privacy arrangement.

Class I activities are represented by a 4x5 matrix of 1 bit elements. The activity is represented by the "ith" column of the array and contains a 1 if that activity is permitted, otherwise, it is zero. Each row corresponds to a category of users. Category 1 is file owner and

category 4 is used in connection with "public files." Whenever a file is created the user declares activities permitted for each file category by entering four alphanumeric characters.

Example:

F R N N Where:

F = free access to all category 1 users

R = read and execute permitted by all category 2 users

N = no access permitted

The matrix associated with this arrangement is:

	a	b	c	d	e	
Category 1	1	1	1	1	1	a = Execute
Category 2	1	1	0	0	0	b = Read
Category 3	0	0	0	0	0	c = Update
Category 4	0	0	0	0	0	d = Change of Status

Nonowners may use other files if arrangements have been made with the owner. This comes under category 2 manipulations. Additional identification is required by the system in this mode and the password used must satisfy some requirement set by the owner of the file to be accessed.

The advantage of this scheme is that it allows a very precise and understandable device for setting up the access profiles for all users. The matrix can be examined once, and those files, where no access restrictions are imposed, can be allowed maximum freedom, thus reducing

the necessity of validating an access everytime. Whenever a restricted file is accessed, it may be easily authenticated. The matrices needed are small and can be easily stored.

The primary disadvantage of this approach is that it is dependent on the integrity of the underlying software, admittedly not a unique problem. While the Cambridge system gives the user assistance in setting up his files, it is only as good as the protection structures used at the file level.

6.2 . Dynamic Protection Structures and the Berkeley Computer Corp. Model I

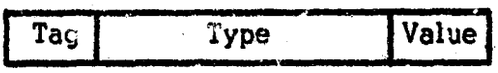
The capability list scheme can be extended beyond that described by Dennis and Van Horn and is so discussed by Lampson [9]. Three fundamental ideas are essential to this approach.

1. Objects are given unique unalterable names called Capabilities. The possession of a capability is considered the right to access the object of that capability.
2. Capabilities are grouped into objects called domains. Whenever control passes from one domain to another, capabilities change.
3. Within each domain are special capabilities called access keys. These keys are the authorization to the domain to grant capabilities within itself. Each domain thus maintains its own access keys and identification.

This scheme allows relative freedom in writing programs and debugging. It insures that elaborate pre-arrangements are not necessary for integrating programs. Additionally it maintains a degree of flexibility in implementation so that modifications can be made at a later date.

The capabilities mentioned can be any word protected by the supervisor or tagged. That word must not be alterable by anyone except the supervisor. A program running in a domain refers to a capability by using an unprotected name. A check is made to authenticate the name of the capability to see if it is contained within the domain.

Capability:



Tag = read only, except to supervisor

Type = File

Value = disk address of index

These are capabilities for various objects in the system such as files, pages, processes, domains, interrupts, terminals, and access keys.

Arrangement of memory is closely related to the nature of the domain. If the mapping hardware is viewed as part of the capability scheme, accessing segments through capabilities implies that a domain is just a collection of capabilities.

Generally the hardware in the system does not allow direct addressing of address spaces between domains. To circumvent this problem either a complex communication routine must be built in or the hardware must be orgainzed so that the address space of one domain is a subset of another. This restricts versatility but, the ring like structure is often satisfactory.

The problems of domains and processes also depend on the system. The capabilities in the domain are used to control processes. Thus, processes can be restricted to running in specified domains. Calls and transfers of control are handled by the use of gates which can be passed as a capability. Gates are used to control entry points with entry usually allowed only at one point. A call stack is used to insure that control is returned to the proper point. Interrupts and traps can cause a transfer between domains when there is a need to force an action on the system.

The use of capabilities and domains provides a basic framework in which program protection can be achieved and security realized. However, there are complications. If the structure is a ringed one, the problems are easily handled since it is assumed that any access to an inner ring is generated by the proper calling program. In the case of a complicated overlapping structure of domains, however, where all the domains not subsets of others, then additional information has to be kept as to the source and destination of every call. Checks have to be made when-

ever an attempt to access a file is made as to whether the call came from a legitimate source.

6.3 The RUSH Time Sharing System

The privacy measures in the RUSH time-sharing system (Remote users of shared hardware) utilizes 80 modules of processors operating in a time sharing mode on an IBM 360 model 50. A monitoring executive controls 60 remote terminal users. The programs are stored in 2,097,152 bytes directly addressable, Rush coexists with the 360 option 2 (multi-programming fixed tasks) and uses IBM's file management schemes.

The user of RUSH converses with an IBM 2741 Selectric or a Teletype Mod 28, 32, 33, 35, or 37. Rush is business or scientific. The only conversational language offered is a problem oriented version of of PL1. Software protection is provided for the RUSH monitor since IBM has none. The principle devices used are:

I. A. Blocking user sign-on, LOGIN contains three levels of protection.

1. Master Account Identification
2. Sub account
3. Protection key on (2)

B. A second try at a valid master account causes a disconnect

If the protection key, after it is entered, is improper, a disconnect will result.

II. Protecting User Program and Data Files

A. Object of a LOAD/SAVE statement is 1 or 2 parameters. (Name, Key)

1. User supplies the name of the file and may optionally add a key.

2. First four characters of the Key are for "read only" requests, and a full six characters are required in order to write over the name on the disk.

3. LOAD/SAVE area is under master account name. This area cannot be executed in a loop, and it is difficult to figure out the key by trial.

III. The remote job entry mode (RJE) allows building a job stream in a separate partition of memory. The customer's identification and file name must agree with his master account in the LOGIN statement. The user cannot read or write on files other than his own.

The RUSH uses full system 360 memory protection features. Memory can be protected in blocks. Only legal blocks for a particular active user can be processed by the executive. Furthermore, the executive is operated under a key that disallows all "store commands" within itself; thus it cannot destroy itself.

IV. There is no protection in the RUSH system against hardware penetration such as wiretaps. Attempted hardware penetrations are logged after forcing a disconnect. A call back system is available, whereby a user may invalidate an attempted access. The system is fully interpretative and snooping can be analyzed. Typewriters with no printout and change of account numbers on a rotating basis help insure greater security.

The security procedures in the MULTICS SYSTEM are basically divided into two areas (1) Compartmentalization - which is the idea of separating various activities of the supervisor to minimize abuse of each one. (2) Auditability - that is to say, high level trace and analysis routines, which are used to spot unauthorized terminals and to verify that the terminal supposedly running is, in fact, the one that was originally authorized.

6.4 Adept-50 Time Sharing System

The Adept-50 Time Sharing System uses software for implementing its security wall. In this system, four security objects are used, the user, terminal, file, and the job. Each group of objects are identified and are further enhanced by a security profile triplet, Authority, Franchise, (Need to Know) and Category (e.g., Eyes Only, Crypto).

Whenever a user successfully logs in, his security profile is retrieved and dynamically derived for the user as a function of his job and the terminal. These profiles act as keys for using Adept-Files.

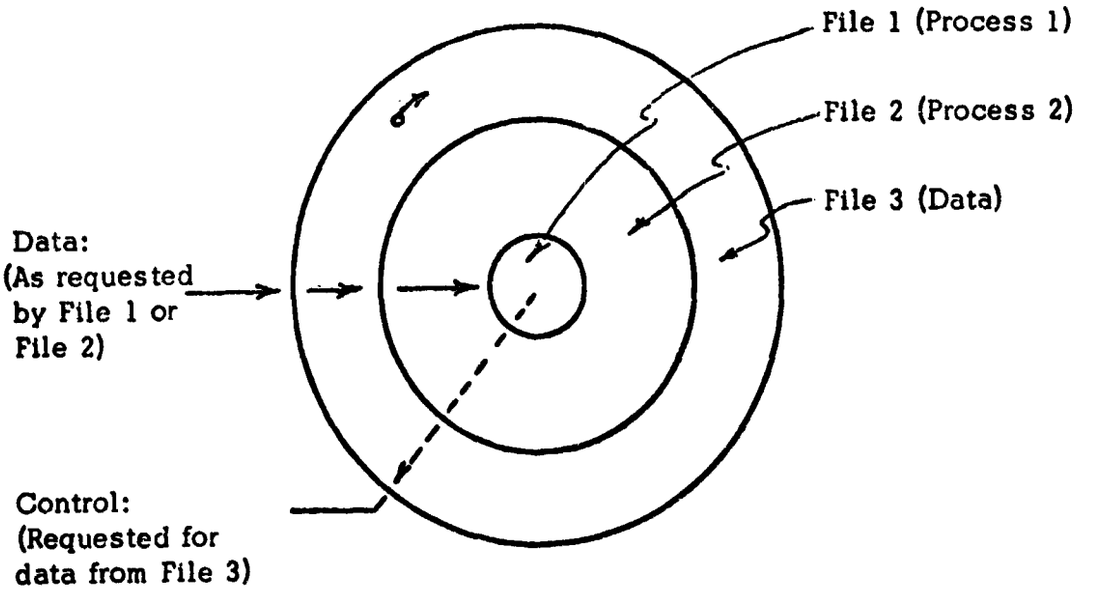
Through the use of Create, command, the file owner can establish authorizations to access and the type of access permitted for his users. A Change command permits modification of properties.

The advantages of this system lie in the fact that it is based on a well formed model of a realistic security structure. It enables a file owner to establish a security profile for the users of his files. The system also has the capability of remembering the security history of previously created files. The system is flexible and easily applied.

The primary disadvantages, as noted by the designer, is in the amount of critical coding, the dispersal of programs, and data in memory which degrade confidence in the system. The system also needs more security compartments or classifications and error detection of security profile data to increase user confidence. A primary disadvantage of this system and most security systems is that they are all based around computers which are not security orientated at all levels. As such most of the protection is based in software, which makes it highly operator dependent. The best solution, it would seem, would be to provide the security at the hardware level with complimentary software used to supervise the overall system.

If one is forced to choose any of the schemes as preferable, the capability list approach is probably superior to the ringed structure as implemented in Multics. The capability scheme allows distribution of control and cost throughout the files. The ringed structure can quickly become expensive. Consider the structure on the following page.

MULTICS RINGED STRUCTURE:



In the ringed structure,data flows inward from file 3 to file 1; but data cannot flow the other way. In many cases,it would be possible to ruin a file by passing bad data to an inner ring necessitating the keeping of backup files in case correction is needed. Validation of data may be required before data is accepted. As long as we stay within a ring there are no problems. As indicated, control passes from the inner ring to the outer rings necessitating validation of control each time an inner file requests data from an outer file. If there are many rings, the system quickly deteriorates in efficiency.

In any system which attaches authority items to each file such as the capability list. There is the problem of duplication of pertinent authority. items for protected fields in one file. Consider J users, K private fields, in each of L files, and if each user has access to the files of S users,

then SxKxL entries are to be made in each authority item for the protection of the user. If there are J users, then there are T = JxSxKxL entries to be maintained. If J = 200, K = 4, L = 2, S = 10 then T = 16,000. Considering storage used per entry, storage and maintenance may be too high. As J approaches J-1 the system becomes inefficient for it is maintaining large lists of unauthorized users. Thus, it will be necessary to periodically purge the system of inactive users placing them on some type of inactive capability list.

6.5 Locating the Security Wall

There are problems associated with deciding where to build the security wall. By security wall we mean the point at which primary system security is implemented. There are two points where the problem can be tackled: (1) at the access managements level, where most efforts are usually directed and (2) at the data base. If we make a system impregnable, there are overhead and customer convenience problems. Alternately, we may desire to have a very open system with easy access and low level trace and suditability. Yet if we desire an open syste, and we must still provide security since we must insure that any accidental or deliberate access of material doc s not compromise a file. This leads us to the possibility of encoding the data base. Figure (11) shows the four possible situations involved in planning a system. The horizontal axis represents a security capability at the data base while the vertical axis represents increasing diligence of access authentication and increasing

process restrictions. As towhich quadrant a system is to operate in depends upon the environment and to a large extent on cost and use of the system. A dedicated military system operating in a potentially hostile environment would be best operated in the first quadrant.

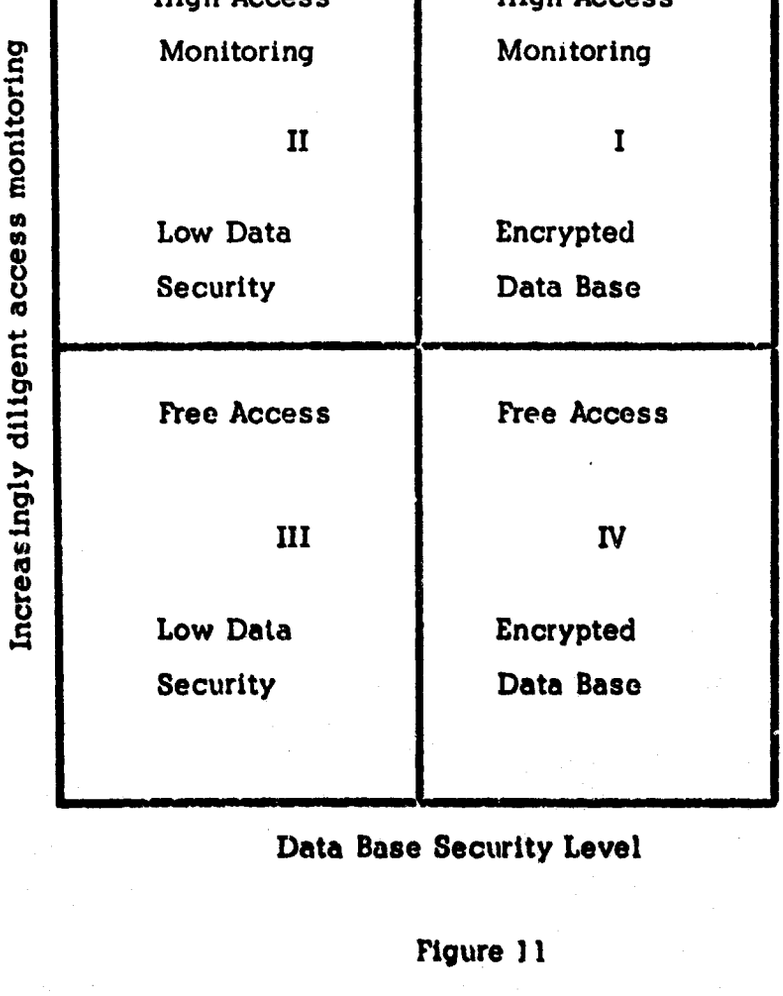


Figure 11

It is not hard to visualize cases where second or fourth quadrant systems might be practical. Realistically speaking most systems operate in the second or third quadrant for reasons of cost. Significantly, an encrypted data base is novel and somewhat impractical for a large class of circumstances. It is certainly conceivable that a system might operate on the axis or combine all ideas in greater or lesser degrees.

PART 7

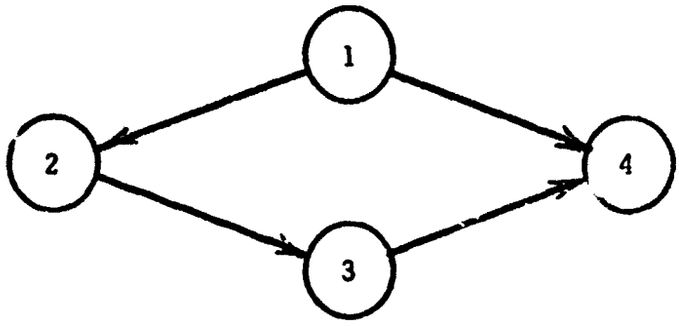
FUTURE WORK

The future of the data bank lies in designing a system with adequate protection yet not so complex or expensive as to discourage its use. The easiest way to kill an idea is to implement it in a way as to make it too troublesome for the majority to use. The "why bother attitude" is the major bottleneck. The development of new schemes for access control and file management are thus indicated.

Additionally there is room for improvement in the technique used to protect memory in many computers especially the large time sharing machines as these will be part of any data bank or computer utility. The paging computer using a virtual addressing scheme offers an excellent opportunity for improving the security of the memory itself.

7.1 Directed Graph Organization

One technique worthy of study from a cost efficiency standpoint is the organization of a system by the representation of files and processes as nodes of a directed graph. If this were done it would be possible to maintain a connectivity matrix [C] specifying the accesses permitted starting at a given entry point. In a connectivity matrix the presence of a one in the intersection of any row and column implies a path exists from a source (the number of the row) to a destination (the number of the column). A zero implies no path. Thus there is an implied direction in the directed graph and the matrix:



Connectivity [C] matrix:

	1	2	3	4
1	0	1	0	1
2	0	0	1	0
3	0	0	0	1
4	0	0	0	0

Whenever a file is added the connectivity matrix is updated to reflect the new connection. More useful is the Reachability matrix (R-matrix) derivable from the C matrix. This gives the files that could be reached from any one point assuming no restrictions within the files. For example the R-matrix reveals what is obvious from the graph, that file one can reach any file while file four can reach none . The fact that

	1	2	3	4
1	1	1	1	1
2	0	0	1	1
3	0	0	0	1
4	0	0	0	0

a file cannot reach itself implies no loops. It is also worthwhile to note that, if we remove path 1 → 4 changing the C - matrix, the R - matrix does not change. Reachability studies can reveal much about the nature of the file structure.

This sort of top down organization is not practical in most cases due to its lack of flexibility. However, if the files are static in number and are not subject to frequent reorganization, this approach might be practical. This end would be aided by the acceptance and creation of the so called data descriptive languages, which would attempt to standardize internal representation of stored data.

7.2 Conclusions

The role of the designers in planning a public data bank will be to develop the techniques necessary to insure that privacy does not become impractical or too costly. The right to privacy is "sacred" and it would be a shame to see it forsaken just for the sake of simplifying the design of a data bank.

In apprasing the effect that large data banks will have upon society one must first consider what direction the managerial policies of the bank will take. If the policy makers are allowed to go unchecked without thorough and carefully formulated guidelines, the public will suffer unwarranted intrusion into its affairs by public and private agencies with disastrous consequences.

Recognizing the problem is unfortunately, only the beginning. The nature of the threat and the most effective countermeasure must be set forth. The countermeasures one employs in assuring privacy of data bank files or computer integrity are varied, and very little analysis of their effectiveness has been made. One can appreicate the problem having heard the story of six computer experts in California last year who set out to see how much trouble it would be to rig vote counting computers. In spite of efforts by fraud detection experts they succeeded in rigging the computers in two out of every three tries.

It is evident that more study into the organization of computers, data banks, and information retrieval systems from a security point of view is required if confidence in a public data bank is to be realized. All considered, the public data bank could be useful, but until effective policy controls are innaguarated, and security technology catches up with the threat, the public data bank is likely to be one of those "good" ideas that never get off the ground.

REFERENCES

1. Baran, P. Communication, Computers and People. AFIPS 1965 Fall Joint Computer Conf. Vol. 27. Pt. 2. Thompson Book Co. Washington, D. C.

2. Baran, P. On Distributed Communications: IX. Security, And Tamper-Free Considerations. Memorandum RM-3765-PR. The Rand Corporation. Santa Monica, California

3. Baran, P. Remarks On the Question of Privacy Raised by The Automation of Mental Health Records. P-3523. The Rand Corporation. Santa Monica, California.

4. Control Data Corp. Control Data 6400/6600 Computing Systems Reference Manual. Pub. N. 60100000, Control Data Corp., St. Paul Minn., 1966

5. Corbato, F. J. and Vyssotsky, V. A. Introduction and Overview of the Multics System. Proc, AFIPS 1965 Fall Joint Computer Conf. Vol N. 27. Pt. 1. Spartan Books, New York, 11 185-196.

6. Dennis, J. B. and Vanhorn, E. C., Programming Semantics For Multiprogrammed Computations. Comm. ACM 9,3 (Mar 1966), 143-155.

7. Grahamn, R. M., Protection in an Information Processing Utility. Comm. ACM 11, 5 (May 1968). 365-369.

8. Glaser, E. L., "A Brief Description of Privacy Measures In the Multics Operating System" AFIPS, Vol. 30, Proc. SJCC, Thompson Books: Washington, D. C., 1967, pp.303-304.

9. Hoffman, Lance J. Computers and Privacy: A Survey: Computing Surveys. ACM June 1969, Vol. I, No. 2.

10. Kahn, David. The Codebreakers, Macmillan, New York, 1969.

11. Lampson, B. W. Dynamic Protection Structures, AFIPS, Fall Joint Computer Conference. Vol. 35.

12. Lickson, Charles P. Attorney at Law. Privacy and The Computer Age. IEEE Spectrum Vol. 5. No. 10. Oct 1968. 345 East 97th Street, New York, N. Y. 10017.

13. Richards, R. K., Electronic Digital Systems, John Wiley and Sons, Inc., New York, 1966.

14. Weissman, C. and Linde R. R., The Adept-50 Time Sharing System, AFIPS 1969 Fall Joint Computer Conference, Vol. 35.

15. Westin, Alan F., Privacy and Freedom, Athenum, New York, 1967.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R&D
(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY (Corporate author) The University of Texas at Austin Electronics Research Center Austin, Texas 78712	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b GROUP

3 REPORT TITLE
PRIVACY AND SECURITY IN DATA BANKS

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Scientific Interim

5. AUTHOR(S) (Last name, first name, initial)
W. A. Garrison
C. V. Ramamoorthy

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
2 November 1970	120	15

8a. CONTRACT OR GRANT NO. AFOSR 69-1792 b. PROJECT NO. 4751 c. 61102F d. 681305	9a. ORIGINATOR'S REPORT NUMBER(S) JSEP, Technical Memorandum No. 24 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) AFOSR-70-2628TR

10. AVAILABILITY/LIMITATION NOTICES
1. This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES TECH, OTHER	12. SPONSORING MILITARY ACTIVITY JSEP through AF Office of Scientific Research (NE) 1400 Wilson Boulevard Arlington, Virginia 22209

13. ABSTRACT

The problems and implications of privacy in today's computer oriented society are many and diverse. The right to privacy is not a formal right guaranteed by the Bill of Rights. How are we to insure that privacy will not become a non-existant commodity? Will the creation of large data banks containing personal information result in "automated blackmail"? Will the end result be a police type dossier on every citizen in the country?

Will the day come when we will evolve into "cashless" society, where all financial transactions are recorded by a computer? How can we accrue the benefits that a large data bank can bring about without the fear that "Big Brother" is watching us.

Some of the possible solutions both legal and technical are discussed along with some current schemes being employed to insure the privacy and security of information. Additionally, some proposals are discussed which might be used to guarantee file integrity. The application of cryptography to the security problem is also discussed. Classification of the various levels of protection is made with suggested environments in which they might be applicable. A suggestion is made as to how overall system performance might be monitored as a result of implementing high level security and auditing routines.

DD FORM 1473 JAN 64

UNCLASSIFIED
Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
DATA BANKS-PRIVACY						
DATA BANKS-SECURITY						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through _____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through _____."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through _____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS), (S), (C), or (U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.