Privacy transformations for databank systems*

by REIN TURN

The Rand Corporation
Santa Monica, California

INTRODUCTION

The term *databank* implies a centralized collection of data to which a number of users have access. A computerized *databank system* consists of the data files, the associated computer facility, a management structure, and a user community. Several classes of databank systems can be defined on the basis of the nature of the organization supported by the databank, and its activity; the nature of the data and its uses; and the structure of the associated computer facility. Such classifications have been discussed in detail elsewhere.[1]

The recent years have seen a steady increase in the establishment of databanks in all sectors of our society in the United States,[2] as well as in other countries:[3,4] in the federal, state and local governments for administrative, law enforcement, education, social welfare, health care purposes; in business and industry for supporting management, planning, marketing, manufacturing and research; in universities for administrative purposes and for supporting social research projects; and the like.

The information maintained in such databank systems includes proprietary data on the operations of industrial concerns, sales data of business establishments, and large collections of personal information on individuals. In all databank systems there is a need to control the access to the data, if for no other purpose than, at least, to assure the *integrity* of the data—that they will not be accidentally modified or erased. In many databanks containing proprietary business information, classified defense information, or confidential personal information on individuals, there is a requirement for *data security*—protection against accidental or deliberate destruction, and unauthorized access, modification or dissemination of the data.

In databanks maintaining personal information on individuals, often collected without the consent or knowledge of the persons concerned, the questions of potential violations of an individual's *right of privacy*—his right to determine for himself what personal information to share with others, arise. These, however, relate to what personal information is gathered in the first place, and thus are legal, political and ethical questions, rather than the technical questions of data security which are addressed in this paper.

Privacy transformations** represent one technique for providing data security—the mathematical/logical transformation of the protected data into forms which are unintelligible to all but the holders of the "keys" to the transformations, i.e., those who know what *inverse* transformations to apply. This capability of privacy transformations is very useful for providing data protection beyond the more conventional access control mechanisms, such as passwords in their various forms, which can be circumvented or nullified through flaws in software, wiretapping, or outright physical theft of data-carrying, demountable storage media.[5]

This paper will first briefly review the relevant characteristics of several classes of privacy transformations, then present a set of suitability criteria for databank applications, and conclude with a discussion of implementation and operational considerations.

PRIVACY AND TRANSFORMATIONS

Historically, there has always existed a requirement to prevent access to information in a message when outside of the physical control of either the originator or the intended receiver, i.e., when the message is in some communication channel. Indeed, certain classes of messages have always been subject to interception, copying, and attempts to uncover the information they contain.[6] In the computer age this threat is also extended to stored messages and data.

Shannon[7] refers to the methods of protecting information in messages and data as *secrecy systems*. There are two kinds:

- *Concealment systems* where the existence of a message is hidden, such as in the case of using invisible ink, or mixing a message with other, unrelated text.

* The research reported in this paper was supported by the National Science Foundation Grant No. GI-29943. However, any views or conclusions contained in this paper should not be interpreted as representing the official position of the National Science Foundation or The Rand Corporation.

** The term "privacy transformation" is synonymous with "cryptographic transformation". It was coined in the early days of computer security research[5] to distinguish the use of cryptographic techniques in civilian and commercial systems from their use for protecting classified national defense information.

- "True" *secrecy systems* where the existence of a message is not hidden, but its *meaning* is concealed by the use of privacy transformations—encryption techniques.

In the following, only the "true" secrecy systems are considered, since concealment systems are not applicable to computerized databank systems—no one can assert that there are no data in such systems, although whether or not there is information worth protecting may be debatable.

A privacy transformation is a mapping $T(K)$, from the space $RS(A)$ of all possible records of finite length which are composed of symbols from a finite alphabet, A, according to the vocabulary, syntax, and grammar of a natural or artificial language, L, into the space $ES(B)$ of strings of characters from an alphabet B. The original, untransformed record, R, is called the "plaintext" and its equivalent transformed character string, E, the "ciphertext" or a "cryptogram." The transformation, $T(K)$, is usually a member of a large space, TS, of similar transformations. The set of parameters, K, of the transformation $T(K)$ are called the "key" which selects $T(K)$ out of the space TS.

Several classes of privacy transformations exist and are in use. A major classification criterion is the nature of the mapping T itself: it may be *irreversible* (i.e., many-to-one) mapping of records into ciphertext strings, or a one-to-one mapping with a unique inverse, T^{-1}. Both classes of privacy transformations find applications in protecting confidentiality and security in databank systems.

Irreversible privacy transformations

A many-to-one privacy transformation, T, when applied to record space $RS(A)$, may convert more than one record into the same ciphertext string E in the space $ES(B)$. That is, given E and the knowledge of the exact transformation used, an uncertainty remains which of the possible records was transformed into E. Unless the intended receivers possess additional contextual information for resolving the uncertainty, many-to-one transformations are inappropriate for precise communication of storage of information.

However, there are situations in databank systems where the maintenance of the original level of information content is not required or could be reduced in the interest of protecting the confidentiality of the information. For example, statistical databank systems, such as the U.S. Bureau of Census and various social sciences research projects, collect data on individuals under the authority of a law or with the individuals' voluntary participation. The data, especially in certain social sciences research projects, may be very sensitive and may lead to considerable harm to some individuals if disclosed.[8] The threats to the confidentiality of such data include legal means—subpoenae issued by courts, grand juries, and investigative committees with subpoena power.[9]

Irreversible privacy transformations can be used in such databank systems to hide personal characteristics of individuals in the group characteristics, and by reducing the credibility of the information. The following can be used:[10,11,12]

- *Aggregation.* The irreversible transformation T applied to a group of data records computes the averages of various data elements in the records and, in each record of the group, replaces the original data elements with the group averages. As the size of the aggregated group of records is increased, the transformation increases the uncertainty about the original information in the records.
- *Random modification.* The transformation consists of adding a randomly varying component to the original information in the records, thereby introducing errors. If the random variables are produced by a process whose statistical characteristics are properly chosen, the statistical value of the modified records are not altered, but credibility of each individual record is now reduced and along with this, the value of such record as incriminating evidence against an individual.

A prerequisite for effective use of the above classes of irreversible privacy transformation is, of course, the original, untransformed records be totally removed from the databank. The price paid for increased data confidentiality is, however, a reduction of the future statistical utility of the data—it will not be possible to make new, precise correlation analyses between various characteristics of individuals (these have been aggregated or innoculated with errors) or to make longitudinal analyses—studies of changes in persons' characteristics or attitudes over periods of time. The confidentiality protection vs. data utility tradeoff is an important question which is still being studied.

Reversible privacy transformations

Transformations in this class are those which are usually discussed as "cryptographic transformations"—the one-to-one mappings from the record space $R(A)$ into the ciphertext space $ES(B)$ which have unique inverses. The protection provided to the data rests in keeping the key, K, of the transformation $T(K)$ from falling into unauthorized hands, and in the expectation that the recovery of original records or the key from the ciphertext forms is a task beyond the resources and know-how of the potential interceptors.

Further classification of reversible privacy transformations, henceforth simply "privacy transformations," can be made on the basis of the mathematical or logical operations involved in applying the transformation. Four principal classes of privacy transformations used in databank systems—coding, compression, substitution and transposition, are briefly discussed below. More detailed discussions can be found in the literature.[6,13,14]

Coding

Coding is a transformation where an entire record, parts of it, words, or syllables of the language L_i used in the record space *RS(A)* are replaced with words or groups of characters of some other (usually artificial) language L_j.[6,15] A coding transformation and its inverse are usually applied with the help of a coding dictionary (code book) or by using table look-up methods. The protection afforded depends on maintaining control over the code books and in frequent changes of codes. Besides providing confidentiality protection, coding can also provide a considerable degree of data compression in transmission or storage. The resulting economy is a main reason for the widespread use of codes in computer files.

Compression

Data compression transformations are used to reduce the redundancy in stored or transmitted data by removing repeated consecutive characters—blanks or alphanumerics, from the records. Other types of data compression transformations attempt to achieve more compact storage of records by "packing" more characters into the storage space normally occupied by a single character. The resultant, compacted data files contain records which have been distorted by the compression algorithms and which will be largely unintelligible when accessed with normal utility programs in the databank. For correct retrieval, decompression algorithms must be applied. Even though data compression is applied mainly to achieve storage or transmission time economies, the associated confidentiality protection may also be sufficient in mild threat environments.

Substitution

Substitution transformations replace single characters or groups of characters of the alphabet A_i of language L used in the record space $RS(A_i)$, with characters or groups of characters of some other alphabet B (or set of alphabets $B_1, \cdots, B_M$). That is, the transformed record is still-composed in language L, but transmitted or stored using alphabet B. Replacement of characters of English alphabet with six-bit binary codes is a very simple substitution transformation. The key K of the transformation $T(K)$ specifies a particular substitution correspondence. The protection obtained depends, in addition to protecting the key, on the number of possible substitution correspondences between alphabets A_i and B (i.e., the size of the key space) and the nature of the language L.

Substitution transformations can be subclassified as *monoalphabetic* and *polyalphabetic*. Each of these could be *monographic* and *polygraphic*. The latter classification refers to number of characters that are being substituted as a group: in monographic substitutions, single characters are substituted (independently of each other and the context of the message) with single characters (or groups of characters). In polygraphic substitutions groups of two or more characters are substituted by similar (or larger) groups.

- *Monoalphabetic substitution.* An alphabet B is chosen to correspond with the original alphabet A such that to each character in A corresponds a unique character (group of characters) in B. As will be discussed later, monoalphabetic substitutions leave the basic language statistics (average character frequency, average polygram frequencies) invariant and, thus, remain susceptible to basic cryptanalytic techniques.
- *Polyalphabetic substitution.* Here the alphabet B is actually a set of alphabets $B_1, B_2, \cdots, B_M$ which are used cyclically with period M. For example, in a monographic M-alphabetic substitution, the first character, r_1, of record R is substituted with a character of alphabet B_1, the second with a character from B_2, the M-th with a character from B_M, and the next character again from B_1. The effect of a polyalphabetic substitution is to hide the original characteristics of the language L, since a given character of alphabet A may now be transformed into M different characters of alphabets $B_1, \cdots, B_M$.

It is common to derive the alphabet B from alphabet A by making a permutation of the characters of A to correspond with the original characters. The simplest such permutation is a cyclic shift of the characters of A by a fixed number of characters, μ. This class of substitution transformations is called "Caesar ciphers." They are extremely simple to solve as, in the case of the English alphabets, a maximum of 25 trials are required to discover the "key," the number of characters that alphabet A was shifted to obtain alphabet B.

A polyalphabetic substitution transformation using M Caesar ciphers as the alphabets $B_1, \cdots, B_M$ (with repetition allowed, i.e., $B_i = B_j$, for some i and j, for several such pairs) is called a "Vigenere cipher." The key is now a set of M numbers which specify the shifts used to generate from alphabet A the alphabets $B_1, \cdots, B_M$. A special case of the Vigenere transformation is the situation where the number of alphabets, M, is larger than the number of characters in a set of records to be transformed. This transformation is called the "Vernam cipher" and it can provide a very high level of protection.[7]

Substitution transformations may be implemented in several ways. Table look-up operations are used for substitutions with alphabets B_i that are arbitrary permutations of the alphabet A. Certain *algebraic* operations, however, permit relatively simple computation of the required substitutions.[14,16,17,18,19]

In algebraic substitutions, the N_A characters of the alphabet A are set in a correspondence with the positive integers $0,1, \cdots, N_A - 1$ (for example, $a=0$, $b \leftrightarrows 1, \cdots$, $y=25$ in the English alphabet). These form an algebraic ring under the operations of addition module (N_A) and

subtraction module (N_A). Then, choosing an integer k in the range 0 to N_A-1 specifies a particular substitution transformation of characters r_i of records R into characters e_i of the transformed version, E, of R:

$$e_i=r_i+k(\text{mod } N_A),$$

and the inverse transformation

$$r_i=e_i-k \text{ (mod } N_A).$$

For polyalphabetic substitutions, a sequence of integers $k, k_0, \cdots, k_{M-1}$, are used cyclically:

$$e_i=r_i+k_j \text{ (mod } N_A), j=i \text{ (mod } M)$$

Polygraphic substitutions of n-character groups (n-grams) by other n-grams can be represented as sets of simultaneous linear congruences and computed by matrix operations.[16,17]

$$e_i=\sum_{j=1}^{n} c_{ij}r_j, i=1, \cdots, n$$

where the elements c_{ij} of the matrix C are selected among integers in the range $0, \cdots, N_A-1$, such that the matrix C has an inverse. If the matrix C is fixed, the substitution is monoalphabetic (in terms of n-grams). Polyalphabetic n-gram substitutions are obtained by introducing a cyclically varying parameter, t, in the matrix C.[18] The matrix $C(t)$ must have the property that its determinant is independent of the parameter t, and is a prime number modulo (N_A).

Transposition

Privacy transformations that permute the ordering of characters in the original message are called transposition transformations. The transformation may be applied to the entire message all at once, or on a block-by-block basis. The alphabet of the message remains unchanged. A common method for implementing a transposition is to write the block to be transformed in a matrix form following some rule and then rewrite in linear form using a different rule. For example, the message may be written first as rows of the matrix and then transcribed by taking the column of the matrix in some specified order.

Transposition transformations retain the character frequency statistics of the language but destroy the higher order statistics (polygram frequencies).

Composite transformations

The effectiveness of privacy transformations can be increased (although not always) by applying a sequence of transformations, $T_1(K_1), T_2(K_2), \cdots, T_S(K_S)$, such that $E=RT_1T_2\cdots T_S$. Typically, the transformation T_i are either all substitutions, all transpositions, or a mix of these.

The case where all transformations are substitutions is called an S-loop substitution transformation:[20]

$$e_i=r_i+k_{j_1}+k_{j_2}+\cdots+K_{jS} \text{ (modulo } N_A)$$

where $j_g=1$ (mod M_g), $g=1, \cdots, S$. If the periods of the polyalphabetic transformations $T_1, \cdots, T_S$, are mutually prime, the period of the composite transformation $T=T_1\cdots T_S$ is the product of periods $M_1, \cdots, M_S$ of the component transformations.

A particularly effective composite transformation suggested by Shannon[7] is a "mixing transformation" which may consist of a sequence of n-gram substitutions and transpositions. Such mixing transformations can be highly effective in hiding the language characteristics, as well as possibly information in the ciphertext E of the nature of privacy transformations used.

SUITABILITY CRITERIA

Among a set of requirements stated by Kerckhoffs some seventy years ago[7] are:

- The cryptographic transformations used should be, if not theoretically unbreakable, unbreakable in practice;
- A knowledge by enemy of system's hardware should not compromise the protection provided to the messages;
- The key should be able to provide all the protection, it should be easily changeable;
- The application of the transformation should be simple, requiring neither complicated rules nor mental strain.

Kerckhoffs' requirements were derived for manually operated communication systems, but are also applicable in modern communication systems and computerized databanks.

The suitability of a particular class of privacy transformations for application in a communication network or in the files of a databank depends on: (1) the relevant characteristics of the particular application, (2) the inherent characteristics of the class of privacy transformations used, and (3) the technical aspects of the system that implements the application and the privacy transformation. Although the principal purpose of using privacy transformations is to provide security to information in transit or in storage, the effects of application of transformation to the utility of the system are equally important —a system may be designed to provide excellent security, but at such a cost in loss of performance that it may become useless.

Application characteristics

The characteristics that affect the effectiveness of a candidate class of privacy transformations in protecting information include the following:

a. *Value of the information.* Whether or not the value of information can be determined adequately depends largely on the nature of information involved. The most difficult to assess is personal information, the easiest to assess is business information. Information affecting national security is usually treated as invaluable and any cost in its protection is considered justifiable. Important is also the time dependency of the assessed value, and this has a direct bearing on the suitability of a class of privacy transformations. For example, if the transformations can resist a cryptoanalytic effort of reasonable intensity for T hours, and the value of the protected information is expected to decrease below a critical threshold in less than this time, the transformation will provide sufficient protection. Determination of the value of information is discussed in more detail in Section V of this progress report.
b. *Language(s) used.* The information to be protected by a privacy transformation is carried in the words of the message (or computer record) and is inextricably identified with these words and the language that provides the vocabulary, grammar, and syntax for embedding the information into the message. In natural languages the vocabulary, grammar, and syntax have evolved over periods of time with no regard to the possible application of privacy transformations. In artificial languages the need to provide protection through the use of privacy transformations can be taken into account already in the language design phase.
c. *Dimensions of the application.* The static and dynamic aspects of the application—the ranges of volumes of messages or data to be stored, processed, and/or transmitted; the required rates and maximum allowed time for operating on a message or data record; and the nature of the processing (sequential, random access, concurrent, etc.), establish criteria which must be satisfied in implementing the privacy transformation.
d. *The personnel characteristics* that affect their role in the application, control, safeguarding of the privacy transformation system: level of expertise, integrity, discipline, etc. Errors made will require repetition of processing or transmissions in providing more intercepted material for the cryptanalyst.

Inherent characteristics of privacy transformations

The most important overall criterion in selecting a privacy transformation is the *amount of security* that it can provide. In general, security appears similar to reliability—both are concerned with techniques for assuring proper operation of systems, and both require *a priori* prediction of the probability of proper operation. From the point of view of a particular implementation, however, reliable operation is a prerequisite of secure operation.

The inherent characteristics of privacy transformations which affect the amount of security provided, and the effective operation of the application process, include:

a. *Size (cardinality) of the key space.* The protection provided by privacy transformations depends on the intruder's uncertainty concerning the transformation used. In general, it must be assumed that the intruder knows the particular class of transformations, but does not know the specific set of key parameters employed. For example, the transformation may be a monoalphabetic substitution, but which one? There are 26! possible permutations of the English alphabet (although not all are permissible, such as the permutation that changes only two letters and leaves the rest the same). A large space of *permissible* keys, each selected with the same *a priori* probability is a prerequisite for any effective secrecy system.
b. *Effect on language.* A privacy transformation provides protection by drastically altering the appearance of the plaintext record (or computer record). Ideally, all the characteristics of the source language (the plaintext language) are altered and made unrecognizable. The extent to which this is achieved is one measure of the suitability of the transformations. For example, a simple (monoalphabetic) substitution is not very effective for languages that have prominent differences in the average frequencies of characters. On the other hand, simple substitution may be quite effective in enciphering numeric-data where all numerals are essentially equally likely.
c. *Complexity.* The complexity of the privacy transformation may contribute to the amount of security by providing more complete scrambling of the language characteristics, but it also contributes to the cost in its application: in computer data banks where transformations are applied by software techniques (programs) complexity translated directly into computer time used for nonproductive (from the point of view of the application) operations.
d. *Effects on dimensions.* Certain privacy transformations involving substitution of characters or polygrams with higher order polygrams (e.g., every character replaced by a pair of characters; a digram replaced by a trigram) increase the length of the ciphertext message compared to the plaintext message. This increases the transmission time or storage space required. Coding transformations, however, can be designed to reduce these requirements.
e. *Error susceptibility.* Compound transformations and super encryptions that involve several transformations applied sequentially may have very undesirable error propagation properties. Lease susceptible are monographic substitutions where error in applying the transformation to a character affects

only that character and does not propagate. However, substitutions that use the ciphertext itself as the key (with appropriate translation by a few characters) are extremely susceptible to error propagation.

f. *Length of the key.* The concept of a "key" to a privacy transformation T is often used in two senses. In the case of polyalphabetic substitutions, for example, the sequence of numbers k_i added to the corresponding message characters, n_i, is called the "key." The length, N, of this sequence $k_1, \cdots, k_N$ is the "key length"; it corresponds to the period in the use of different alphabets. For the cryptanalyst who attempts to discover the key by studying intercepted ciphertext messages, longer keys mean more unknowns that must be determined, hence, providing more protection to the information. In certain implementations, however, the key sequence is produced by a computational process which is specified by only a few parameters. Here the "key" that selects the privacy transformation (i.e., the production of the sequence applied to the plaintext) is the set of parameters, rather than the sequence produced. If the cryptanalyst can attempt to solve for the parameters of this process, rather than the entire sequence produced by the process, his number of unknowns is greatly reduced. An example of this is the generation of random numbers $X_i = AX_{i-1} + B \pmod N$. A large number of X_i are produced, but there are only three unknowns: A, B, and X_0.

The various characteristics listed above are evaluated for the different classes of privacy transformations in a following section.

System implementation characteristics

The third set of characteristics that determines the suitability of a particular class of privacy transformations is associated with the system implementation of the application.

a. *Processing capability.* The processing speed of the system and the storage capacity. Availability of instructions for easy application of the privacy transformations. Availability of hardware devices or software programs. Capability to use suitable file structures.
b. *Error environment.* The error characteristics of the communication channel, or the storage medium. The availability of error detecting/correcting codes.
c. *Security environment.* The capability to provide for the security of the keys for the privacy transformations, and to protect the information in the enciphering and deciphering processes.
d. *System personnel.* These may be the same as the applications personnel. Also included are the operators, programmers, maintenance engineers of the system. Their expertise in operating the system, as well as their integrity has an important role in making the use of privacy transformations a success.

Language characteristics

As stated previously, information is communicated by using a language. Concealment of the information in a written record (on any medium such as paper, magnetic surface, electronic circuitry, etc.) through the use of privacy transformations requires that the message is transformed in such a way that any resemblance with the original form is obliterated.

Natural Languages. Investigations of the structures of natural languages[21,22] have shown that there are a number of structural and statistical characteristics of their vocabularies that, in normal usage, are relatively insensitive of the context and can be used to identify the particular language used:

a. *Single character (monograph) frequency distribution*—there is a large difference in the usage of letters in the vocabularies of natural languages. For example, on the average the letter "e" appears 100 times more often than the letter "q"; in French the letter "q" occurs 11 times as often as in English. Special vocabularies, such as family names of persons or tactical orders.
b. *Polygram frequency distribution.* The data here is normally limited to pairs of characters (diagrams) which show transitions of letters to other letters in the word structure, and triplets of characters (trigrams). For example, the two most frequent diagrams in English are "th" and "he," but "es" and "en" in French and Spanish. The two most frequent English trigrams are "the" and "ing," in French they are "ent" and "que."
c. *Starting and terminal letter frequencies.* These differ sharply from the general letter frequency distribution. For example, the letter "e" (most frequent in the general distribution) ranks 14 as a starting letter, and first as a terminal letter. The letters "v," "q," and "j" have extremely low frequencies as terminal letters. Proper names, in general, have different starting and terminal letter frequencies.
d. *Word usage frequencies.* Word frequency distributions are much more dependent on the particular application areas than the various polygraph frequencies. The first ranking words, however, tend to be prepositions and connectives which are used in the same manner in all application areas. For example, the first nine are: the, of, and, to, a, in, that, is, was. The word frequency distributions form the basis of the so-called "probable word" method of cryptanalysis.

TABLE I—Effects of Classes of Privacy Transformations on Language Characteristics

Characteristic	Substitutions				Transposition	Composite
	Monoalphabetic		Polyalphabetic			
	Simple	m-graphic	Simple	k-graphic		
Single character frequency	Invariant (changed alphabet)	Changed	Changed*	Changed*	Invariant	Changed**
k-gram frequency distribution	Invariant	Changed***	Changed	Changed	Invariant	Changed
Word frequency distribution	Invariant (within the new alphabet)		Changed	Changed	Changed	Changed**
Pattern word structures	Invariant	Changed	Changed	Changed	Changed	Changed
Syntactic structure	Invariant	Partly Changed	Invariant	Partly Changed	Changed	Changed

* Within the period of applying alphabetic transformation; over large numbers of periods, some of the characteristics may show invariance.
** Assuming a composite of transpositions and polyalphabetic substitutions.
*** Changed for certain values of k in k-grams (e.g., for k not a divisor of m).

e. *Word structure patterns* (isomorphisms). There are groups of words which have similar patterns of letter occurrences in the word (e.g., aDDeD, sEEmEd, have the pattern -xx-x-). This structural information can be used to place words in "congruence" classes, and the classes can be used in cryptanalysis.
f. *Word length frequencies*. This information also characterizes different languages and, on occasion, application areas. For example, the mean word length in English is 4.5, but 5.9 in German.

Various other statistics about word structure, word-to-word transitions, etc. can be derived. Their utility from cryptanalytic point of view depends on the specific application area. Table I presents an assessment of the effects of privacy transformations on language characteristics.

The statistical structure of a language provides a certain degree of predictability in constructing words in that language. This predictability can be measured in terms of *redundancy*—the inefficiency in the use of the available character sequences from a given alphabet as words of the language. For example, a redundancy of .75 indicates that 75 percent of the possible character-sequences (up to some relatively small length) are not used as words. In general, languages with high redundancy require more complex privacy transformations than those with low redundancy.

The sentence structure and the rules of proper usage, *syntax and grammar*, likewise, place constraints in the formation of strings of words as sentences in the message. The more rigid the syntactical and grammatical requirements imposed on the message source, the more complex privacy transformations are required to effectively diffuse the structure and increase the uncertainty of the cryptanalyst.

Artificial Languages. Application of privacy transformations to information in computerized retrieval systems involves working with so-called artificial languages (e.g., codes, query languages, and programming languages) and data. These differ significantly from the natural languages and can be expected to influence the protective effectiveness of privacy transformations in different ways.

Four levels of artificial languages can be recognized. Starting with the level most similar to a natural language there are:

a. *Query languages*. These are languages designed for user interaction with the retrieval system—to request information, choose processing options, etc. For easy interaction with the system the vocabulary of a query language statement available to the user is usually a restricted subset of natural language words, arranged with precisely specified structure in natural language sentences. For example, a request may be stated RETRIEVE ALL NAMES (ENGINEER, CALF, AGE: 30-50). Many query languages provide menus of operations that are allowed. Here the wording of the choices is, likewise, kept relatively brief. Query language statements are used mainly in communication channels linking terminals with the retrieval system computers.
b. *Higher Order Programming Languages*. Programs written in higher order languages such as FORTRAN, PL/1, ALGOL, etc. may require privacy transformations if they are considered sufficiently valuable (such as certain proprietary programs) and stored in computer accessible form. Programming languages have a *fixed vocabulary* of words selected from the natural language to specify the program structure and designate dataprocessing operations (e.g., EQUIVALENCE, DIMENSION, DO, READ, WRITE, etc.) and an open-ended *variable vocabulary* specified by the programmer for variable names, numerical values, arithmetic logi-

cal processing statements, and such. The choice of some of these words is subjective with the programmer. Often these are similar to words in the natural language (e.g., ICOUNT, JSET, II, AVALUE=BVALUE(J)+INDEX, etc.). The *character set* of a typical higher-order programming language includes many special characters (PL/1, for example, has a 60-character alphabet). The *syntactical and grammatical* rules are very rigid and must be precisely followed.

c. *Assembly languages*. An intermediate step from the higher order language designed for increasing programming ease to efficient computer-executable form—the "machine language," is an assembly language. It is obtained from a higher-order language through a compilation process. The vocabulary of an assembly language consists of mnemonic names for the instruction set of the computer (usually two- or three-letter groups) and the variable names specified in the higher order language. The format is quite rigid. Programs are sometimes stored in the assembly language form.
d. *Machine language*. A machine language program is composed of instruction codes, constant numerical values, and addresses. All of these are coded as binary numbers. The instruction words are divided into fixed length fields that contain the different codes. The sets of allowed code numbers for the various fixed fields may have different cardinalities. No resemblance with a natural language is left. The alphabet consists of binary numbers with the ranges of values specified by the field lengths. Operating programs are usually stored in the machine language form.
e. *Interpretive languages*. In some interactive computer systems programs are stored in a higher-order language only in the execution phase (e.g., the JOSS language). For this, a dictionary and various analysis programs are maintained and used. The characteristics of higher-order languages discussed above are also typical of interpretive languages.

The statistics of higher order artificial languages tend to reflect the statistics of the underlying natural language, but this similarity decreases in assembly languages, and is essentially nonexistent in machine language.

The overall effect of the limited fixed vocabulary, large character set, rigid structure and lack of syntactic ambiguity is a reduction of the effectiveness of applying privacy transformation. On the other hand, the availability of the variable vocabulary can be used to change the statistical characteristics of the language almost at will.

Data. The principal use of privacy transformations in retrieval systems can be expected to center about protection of data, both in storage and in transit. Certain categories of personal information, in particular, require a degree of confidentiality sufficiently high to warrant the use of privacy transformation.

In general, personal information records consist of the following parts:

a. Person's name, address, and other identifying characteristics. In some data files the name and address may be replaced by a code number, where the name/address and code number correspondences are maintained in some dictionary. The name and address, if included, can be expected to be in the natural language. Other characteristics may be coded.
b. General descriptive information, a mixture of proper names (e.g., the birth place, parents), codes, and numeric information.
c. Narrative information. A mixture of natural language sentences, abbreviations, and codes (e.g., the description of a person's criminal history).

The inclusion of names, abbreviations, and numerical codes can be expected to considerably change the statistics of personal data as compared with the natural language text. In particular, it may be expected that the occurrence of proper names which have no identical natural language words will tend to "flatten out" the single letter and polygram frequencies.

In records with fixed formats (i.e., where fixed length fields are provided for names, addresses, etc.) the "blank" characters will have a relative high frequency of occurrence (just as in numeric data, zeroes will be the most frequent numerals). Sorting of the files into alphabetic or numerical order, likewise, in a structural feature of data files that can weaken the effectiveness of privacy transformations.

EFFECTIVENESS AND COST

The two most important considerations in selecting a class of privacy transformations for implementation in a databank system are the *effectiveness* of the transformations in providing data security and the initial and recurring *costs* of providing this protection. These must be weighed against the estimated *value* of the protected information in order to implement a *rational* protection system—one that provides a level of data security warranted by the value of the protected information.[1]

Effectiveness measures

The effectiveness of privacy transformations is usually discussed in terms of the resources and expertise required by the "enemy" cryptanalyst to "break" the privacy transformation used, i.e., to discover the key. The following assumptions about the intruder cryptanalyst must be made:

- He knows in detail the class of transformations being used; the language (vocabulary, syntax, grammar)

used in the records or programs; the general subject matter of the data. He does not know the specific key of the privacy transformations used or the exact contents of protected records, although he may know some words that are highly likely to occur.
- He is knowledgeable in computer technology, operation and use; knowledgeable in the operational procedures of the target databank; and has a digital computer at his disposal.

A necessary prerequisite for attempting to break a privacy transformation system is the availability of a sufficient amount of ciphertext. The minimum amount required for unique recovery of the record or message is called the *unicity distance* by Shannon.[7] It is a function of the size of the key space, the redundancy of the language, and the number of alphabets (key period) used in polyalphabetic substitutions, or the period of transposition transformations. For example, the unicity distance for M-alphabetic substitution transformation is $53M$ and for transposition of period M (i.e., character permutations take place in M-character groups) the unicity distance is 1.7 log $M!$. In general, it may be expected that in databank applications there will be large amounts of ciphertext available to the intruders. Note, however, that the ciphertext available must be longer than the key period, i.e., a key is used more than once to transform records or messages. In databank systems this can be expected to be the situation, as using nonrepeated keys to transform large amounts of data will be impractical from the point of key management for permitting information retrieval, and for providing security to the keys themselves.

Other information that helps the cryptanalyst includes:

- A number of different records known to be transformed with the same key—these can be used for simultaneous solution and checking of trial solutions.
- Fragments of plaintext corresponding to the available ciphertext, or paraphrased messages or records that are in the available ciphertext. These are very useful for generating trial solutions.
- Knowledge of the probable words in the records or knowledge of the key selection habits of the target databank—if keys are short, they may be *coherent*—are words of natural language or generated by some algorithmic process.
- As much knowledge of the statistical characteristics of the language used in the plaintext as possible.

Again, a great deal of this information, including plaintext fragments, must be expected to become available to the intruder. The ability of a privacy transformation system to withstand a cryptanalytic attack for sufficiently long (i.e., for the information to lose its value, or for the data to be retransformed) can be regarded as a measure of effectiveness of the transformations. There are two kinds of measures of effectiveness: information-theoretic measures and pragmatic "work-factor" measures.

Information-theoretic measures

These measures assess the theoretical effectiveness of a secrecy system against cryptanalysis where the intruder has unlimited resources and expertise available. Shannon[7] modeled the situation as follows: each message, R, and each choice of a privacy transformation key, K, has, from the point of view of the cryptanalyst, *a priori* probability associated with it. These are $p(R)$ and $p(K)$, respectively, and they represent the crytanalyst's knowledge of the situation before the message is transmitted.

After he intercepts and analyzes an intercepted ciphertext, E, he can calculate *a posteriori* probabilities of the various messages and keys, $p_E(R)$ and $p_E(K)$, respectively, that could have produced the intercepted ciphertext. *Perfect secrecy* is obtained if the $p_E(R)=p(R)$ and $p_E(K)$, i.e., the cryptanalyst has obtained no information at all from the intercepted ciphertext. Shannon shows that in order to have perfect secrecy, the number of keys must be at least as great as the number of possible messages.

As a measure of the theoretical amount of secrecy, Shannon defined *equivocation*—a statistical measure of how near to solution is an average cryptogram E of N characters. There are two equivocations, that of the key, $H_E(K,N)$, and the message equivocation, $H_E(R,N)$, where

$$H_E(K,N)=\sum_{E,K} p(E,K)\log p_E(K)$$

$$H_C(R,N)=\sum_{E,R} p(E,R)\log p_E(R)$$

where $p(E,K)$ and $p(E,R)$ are the *a priori* probabilities of cryptogram E and key K, and cryptogram E and message R, respectively. The summation is over all possible cryptograms of N letters and all keys or messages.

The equivocation functions for the key of the privacy transformation, $H_E(K,N)$, has the following properties:

- Key equivocation is an non-increasing function of N.
- For perfect systems, key equivocation remains constant at its initial value (when $N=0$).
- For non-perfect systems, the decrease in key equivocation is no more than the amount of redundancy in the N letters of the language L used in the plaintext.
- For most of the simple types of privacy transformations, equivocation becomes zero after the number of intercepted characters exceeds the unicity distance. After that point, a unique solution is theoretically possible.
- For certain privacy transformation systems, called *ideal* secrecy system, equivocation remains non-zero no matter how much ciphertext is intercepted.

These properties point out the importance of the *redundancy* in the language used in the plaintext records or language. If there is no redundancy at all, i.e., if all words are of equal length, say N, and if any combination of N characters of the alphabet used is a meaningful word of the language, the secrecy of the system will be

perfect. Such properties do not exist in natural languages, but can be designed into artificial languages. However, they tend to be in conflict with present trends of making artificial languages as close to natural languages as possible.

All presently existing large databank systems have redundancy in the stored data or programs. Large amounts of ciphertext, fragments of plaintext, etc. are likely to be readily available. Although exact evaluation of initial equivocation for such databank is a complex problem and has not been attempted, it is clear that simple privacy transformation systems applied here are theoretically solvable. Nevertheless, p privacy transformation systems for databank applications can be devised to have sufficiently high levels of *practical* security, i.e., sufficiently high *work factors* for the intruders, to discourage attempts to break these systems through cryptanalysis.

Work factor measures

On the practical side, an assessment of the effectiveness of privacy transformations can be attempted in terms of the effort and resources required to break the system through cryptanalysis. Such a measurement has been called the intruder's "work factor." The units of measurement can be the expected number of logical/mathematical operations. These can be converted into units of time and, subsequently, into dollars by specifying a computing capability which the intruder is expected to have available.

Several authors have examined computer-aided cryptanalysis and the effort involved.[20,23,24] Tuckerman,[20,25] in particular, has probed the computational effort involved in breaking of polyalphabetic single-loop and 2-loop substitutions under several assumptions of availability of plaintext fragments:

- For the simplest monoalphabetic substitution, the Caesar cipher, a single subtraction is sufficient if a fragment of the corresponding plaintext is available. If not, the "running down the alphabet" method can be used to generate N_A trial solutions (corresponding to the N_A characters in the alphabet) and examined for plausible plaintext. Alternately, character frequency distributions can be computed for the ciphertext and matched with the known frequencies of the language to produce solution candidates for examination. The time required on a moderately fast computer would be a few minutes at the most.
- A single-loop polyalphabetic (Vigenere) substitution of period M, can be reduced to M Caesar ciphers by a statistical analysis of the ciphertext. At least $20M$ characters of ciphertext are required. Considerable computation may be required to estimate the correct period—candidate periods are proposed, character frequency distributions computed, and correlation tests made. On a computer, however, the work is again measured in minutes or a few tens of minutes.

The 2-loop polyalphabetic substitution transformations can likewise be solved by conversion into single-loop cases and, subsequently, into Caesar ciphers. The computation required is more extensive but, by no means prohibitive. A larger hurdle to the would-be intruder is the development of programs that are needed.

Transposition transformations are solved by similar methods—by generating trial solutions, performing statistican analyses on n-grams, and using "heuristic" techniques to reduce the search space. Computational tasks, again, are not prohibitive. However, it is possible to construct complex composite transformations which require hours of computing time for their solution.

In general, the availability of digital computers and sophisticated computational algorithms has greatly reduced the protection provided in the paper-and-pencil days by the polyalphabetic and substitution transformations. Whether or not this protection is adequate in a given databank system depends on the value of protected information both to the intruder and to the owners.

Costs

The use of privacy transformations involves the initial costs of the necessary hardware or software, and the recurring costs of additional processing required and maintenance of the integrity of the privacy transformation system used.

Hardware costs are involved, in particular, in application of privacy transformations to terminal-computer communication links. Here the enciphering/deciphering device at the terminals is likely to be a hardware device. However, the logic circuitry involved is not necessarily excessive or costly since the integrated circuit prices are steadily falling. For example, the hardware involved in one, rather sophisticated ciphering/deciphering unit[26] for transforming 16-byte blocks consists of 162 TTL logic modules which could be placed on four LSI chips at a density of 280 circuits per chip. Transformation of one block requires 165 microseconds.

Software requirements, likewise, are not necessarily expensive. Programming of the mentioned transformation required some 1300 bytes of storage of 9 ms. on the IBM 360/67 computer.[26]

Other experimental data on the cost of applying privacy transformations yields similar results. The application of privacy transformations to 10-bit characters in a CDC-6600 computer[27] has shown the following percentages of processing time required for the transformations:

- Vernam type polyalphabetic substitution transformation (with one-time-only key): .66 percent to encode, .66 percent to decode.
- Polyalphabetic substitution with a short, periodic key (using table look-up technique): .25 percent to encode, 3.32 percent to decode.
- Polyalphabetic substitution with short, periodic key, using modular arithmetic for transformation: 1.94 percent to encode, 4.38 percent to decode.

As usual, there are the memory space vs. execution time overhead tradeoffs that can be applied.

The above cost figures are quite sensitive to the type of application, the computer system, and specific implementation of the transformations, and they represent only isolated data points. Estimates of decreased functional capability of a databank system due to the use of privacy transformations, as well as costs of maintaining the secrecy system integrity through providing key security, key changes, and the like, are even less available.

IMPLEMENTATION IN DATABANK SYSTEMS

Privacy transformations can be used in databank systems for protecting communications between the computer and remotely located terminals, the data stored in the files, or both. The suitability criteria for implementing privacy transformations in databanks—processing capability, error environment, security environment, and system personnel expertise—have already been discussed. These, and the specific application characteristics of the databank, provide the general criteria for selecting the type of privacy transformations to be used.

A privacy transformation system can be implemented by using hardware devices, software, or both.[26,28] Software implementation is more attractive for performing the transformations in the computer processor unit, while hardware devices appear more suitable for implementation in the remote terminals. However, the decreasing cost of hardware is making feasible the use of special privacy transformation modules also in the central processors.[26]

Application communication links and data files

The major differences in the application of privacy transformations in communication links (usually hardware switched or dedicated telephone circuits) and in data files include the following:

- In communication systems the encoding and decoding operations are done at two different locations and two copies of the key are required, while in file system application these operations are performed at the same location and only one copy of the key is needed.
- A specific communication usually involves one user, while a file may be shared between many users with different access and processing authorizations.
- In communication links, the message remains transformed for a very short time interval since encoding and decoding operations are performed almost simultaneously, in data file application this time interval may be days or months.
- The transformed records in files may be subject to selective changes at unpredictable time intervals and at unpredictable frequencies, while the message in communication link is not changed in transit.
- A change of the privacy transformation keys in a communication application is a simple replacement of the old key with the new one. In data files, this entails reprocessing the entire file of records, or maintaining an archival file of previously used keys and associated indices.
- A common-carrier communication link normally uses certain signal patterns for internal switching control. These should not appear in the ciphertext form of messages. There is no such problem in files as the control data parts.
- Communication links have higher error rates while errors in the file system are more amenable to detection and control.
- There is much less processing capability available at terminals than in the central processor.

Several other differences emerge when the implementation of one or the other of the three main classes of privacy transformations—substitutions, transpositions, composite transformations—are considered below.

Key management

The differences in the nature of communication systems and data files impact the choice of the type of privacy transformations and, in particular, the requirements for key generation, storage, logistics, and safeguarding. For example, in communication systems totally random keys can be used only once and then discarded, but in the file system they must be stored or means provided for their generation later, thus reducing the level of security such systems usually offer.

The need to store transformation keys in the data file application sets up different requirements for key possession and control than in communication links. In the latter case, individual users could be in possession of their own keys and copies stored in the processor. For file use, however, this is not desirable. The entire file, or the various classes of records in the file, should be transformed with the same key, but the keys need not be revealed to the users. Rather, the access to the file would depend on a different set of identification-authentication procedures which establish the authorization of a user to access the file and give him access to routines that retrieve or store the involved records. If the privacy transformations used have a high work factor and the key security is also high, reprocessing of the file for changing of the key need not be very frequent.

Key generation

A long key, i.e., the specification of different alphabets and their sequence, is necessary in substitution transformations where the transformation is performed by using modular arithmetic—modulo A_N addition of the key characters to the plaintext characters. As discussed previously, approximately $20M$ characters of ciphertext is needed for computer-aided solution of these transformations. If the key is sufficiently long such that this amount of ciphertext is not produced, a high degree of security is

achieved (although fragments of plaintext and the language characteristics may still make breaking of the key practical). Since storage in the computer memory of very long keys is not economical, various algorithms are used to *generate* the key as required. Computation of random numbers and feedback shift-register sequence generators are among the standard key generator techniques.[29,30] A drawback of algorithmic key generation approach is that now the *real* key is not the pseudo-random sequence added to the plaintext, but the much shorter set of parameters (an initial state and a few constants) that are used to specify a particular version of the key generation algorithms. For example, only $2n$ properly selected bits are needed in a linear feedback shift-register to produce a nonrepeating sequence of 2^n bits. Also, it is possible to recover the parameters and, hence, the key by analyzing fragments of the key stream.

Another problem with key stream generators in the communication links is the need for *synchronization* of the encoding key stream at the transmitting end with the decoding key stream at the receiving end. Such synchronization may be hard to achieve and maintain in noisy communication links. Self-synchronization is definitely a property which key stream generators should possess.[31]

Transposition and composite-privacy transformations are usually called *block transformations* as they are applied to a block of plaintext simultaneously. Very complex transformations with high work factors can be obtained.[26,32] The required keys can be stored in blocks of the main memory, special read-only memories, or generated algorithmically. Assemblying of the key at the transformation application time for several independent "subkeys" can provide additional protection against key compromises. A sophisticated block transformation can be expected to involve several sequentially applied transformations on the entire block or various subblocks. Since the computation time can be substantial for the software implementation in the processor, special purpose hardware may turn out to be more economical. In terminals, the hardware implementation is the only alternative.

CONCLUDING REMARKS

The need for data security in computerized databank systems is increasing. Privacy transformations can provide protection against a variety of threats—wiretapping to obtain transmitted information or system access control information, active entry into the system through illicit terminals, disruption through insertion of illegitimate information in the communication channel, snooping in the files, theft of removable storage devices, and the like. Their use in the databank systems, both in communication links connecting remote terminals to the processor and in data files, is now economically feasible.

On the other hand, digital computers greatly simplify the cryptanalytic tasks of the would-be intruders who must be expected to have available the necessary resources and expertise. It is important, therefore, for those charged with the design of data security mechanisms in databank systems to understand the capabilities and shortcomings of privacy transformations, and to be aware of the criteria which must be applied in their selection. This paper has strived to contribute to such understanding and awareness.

However, privacy transformations are only one facet of the general problem of access control and data security. The design of a data security system providing protection commensurate with the value of protected information requires consideration of all types of available data security mechanisms, their relative advantages and disadvantages, cost-effectiveness, and the structure and operation of the databank system. The measures of the amount of security provided by different mechanisms, measures of the value of information, and the tools for tradeoff analysis, are now beginning to crystalize into a discipline of "data security engineering." It is likely that in a few years the design of data security systems will be much less an art than it is today.

REFERENCES

1. Turn, R., Shaprio, N. Z., *"Privacy and Security in Databank Systems—Measures of Effectiveness, Costs, and Protector-Intruder Interactions,"* AFIPS Conference Proceedings, FJCC, Vol. 41, pp. 435-444, 1972.
2. Westin, A. F., Baker, M. A., *Databanks in a Free Society—Computers, Record-Keeping and Privacy,* Quadrangle Books, New York.
3. Younger, K., *Report of the Committee on Privacy,* Her Majesty's Stationery Office, London, July 1972.
4. Carroll, J. M. "Snapshot 1971—How Canada Organizes Information About People," *AFIPS Conference Proceedings,* FJCC, Vol. 41, pp. 445-452, 1972.
5. Petersen, H. E., Turn, R., "System Implications of Information Privacy," *AFIPS Conference Proceedings,* SJCC, Vol. 30, pp. 291-300, 1967.
6. Kahn, D., *The Codebreakers,* Macmillan, New York, 1967.
7. Shannon, C., "Communication Theory of Secrecy Systems," *Bell System Technical Journal,* Vol. 28, pp. 654-715, 1949.
8. "ACE Study of Campus Unrest—Questions for Behavioral Scientists, *Science,* Vol. 165, July 1969.
9. Nejelshi, P., Lerman, L. M., "A Researcher-Subject Testimonial Privilege—What to do Before the Subpoena Arrives," *Wisconsin Law Review,* No. 4, Fall, pp. 1085-1148, 1971.
10. Hansen, M. H., "Insuring Confidentiality of Individual Records in Data Retrieval and Storage and Retrieval for Statistical Purposes," *AFIPS Conference Proceedings,* Vol. 39, FJCC, pp. 579-585, 1971.
11. Fellegi, I., "On the Question of Statistical Confidentiality," *Journal of the American Statistical Association,* pp. 7-18, March 1972.
12. Boruch, R. F., "Strategies for Eliciting and Merging Confidential Social Research Data," *Policy Sciences,* Vol. 3, pp. 375-397, 1972.
13. Gaines, H. F., *Cryptanalysis,* Dover Publications, Inc., New York, 1956.
14. Sinkov, A., *Elementary Cryptanalysis—A Mathematical Approach,* Random House, New York, 1968.
15. Friedman, W. F., Mendelsohn, C. J., "Notes on Code Words," *American Mathematical Monthly,* pp. 394-409, August 1932.
16. Hill, L. S., "Cryptography in an Algebraic Alphabet," *American Mathematical Monthly,* pp. 306-312. June-July, 1929.

17. Hill, L. S., "Concerning Certain Linear Transform Apparatus of Cryptography," *American Mathematical Monthly,* pp. 135-154, March, 1931.
18. Levine, J., "Variable Matrix Substitution in Algebraic Cryptography," *American Mathematical Monthly,* pp. 170-178, March, 1958.
19. Levine, J. L., *Some Elementary Cryptanalysis of Algebraic Cryptography.*
20. Tuckerman, B., *A Study of the Vigenere-Vernam Single and Multiple Loop Enciphering Systems,* IBM Corporation Report RC 2879, May 14, 1970.
21. Miller, G. A., Friedman, E. A., "The Reconstruction of Mutilated English Texts," *Information and Control,* pp. 38-55, 1957.
22. Shannon, C. E., "Predilection and Entropy of Printed English," *Bell System Technical Journal,* pp. 50-64, 1951.
23. Fiellman, R. W., *Computer Solution of Cryptograms and Ciphers,* Case Institute of Technology Systems Research Center Report SRC-82-A-65-32, 1965.
24. Edwards, D. J., *OCAS—On-Line Cryptanalytic Aid System,* Massachusetts Institute of Technology Project MAC Report MAC-TR-27, May 1966.
25. Gridansky, M. G., "Cryptology, the Computer and Data Privacy," *Computers and Automation,* pp. 12-19, April 1972.
26. Feisel, H., Notz, W. A., Smith, J. L., *Cryptographic Techniques for Machine to Machine Data Communications,* IBM Corporation Report RC 3663, December 27, 1971.
27. Garrison, W. A., Ramamoorthy, C. V., *Privacy and Security in Databanks,* University of Texas Electronics Research Center, TM 24, November 2, 1970.
28. Kugel, H. C., "Three Cipher-Decipher Programs Make Good Os/360 Demo's" *Canadian Datasystems,* pp. 38-40, April 1972.
29. Carroll, J. M., McLelland, P. M. "Fast 'Infinite-Key' Privacy Transformation for Resource-Sharing Systems," *AFIPS Conference Proceedings,* Vol. 37, FJCC, pp. 223-230, 1970.
30. Reed, I. S., Turn, R., "A Generalization of Shift-Register Sequence Generators," *Journal of the Association of Computing Machinery,* Vol. 16, pp. 461-473, July 1969.
31. Savage, J. E., "Some Simple Self-Synchronizing Digital Data Scramblers," *Bell System Technical Journal,* pp. 448-487, February 1967.
32. Skatrud, R. O., "A Consideration of the Application of Cryptographic Techniques to Data Processing," *AFIPS Conference Proceedings,* Vol. 35, FJCC, pp. 111-117, 1969.