Additionally, the concatenation–intersection (G&) might well be required. In both of these connectors, the location of the value in the multivalued segment may wish to be used as opposed to the normal GER and G& where only the presence of the word is used. As a final example, consider the following for a special G-negation:

An application is defined with one logical file containing 78 segments or fields. A predefined pseudo-segment called SEG80 is defined as the concatenation of Segments 1, 7, 13, 17, and 18.

Thus:

Seg 80 ≡ (Seg 1 GOR Seg 7 GOR Seg 17 GOR Seg 18)

A user, on an interactive terminal, wishes to access the file with the following request:

(Seg 1 GOR Seg 7 GOR Seg 13 GOR Seg 17 GOR Seg 18) = (Word & WD)

It is identical with

Seg 80 = (Word & WD)

Another user might wish to access the same file except that, for this user, the concatenation should not include Seg 13. The special G-Negation (G−) would allow this as:

Seg 80 G− Seg 13) = (Word & WD)

As long as only the GOR and the G− are the only available G-connectors, the processing formalism is not too complex; once, however, other G-connectors are required (especially the GER & the G&), the composite meaning may become most complex.

As can be seen, we have attempted to define four areas in which additional man-machine conversational linguistic expressions can and, we believe, should be defined to make easier the way our users may communicate with the computer. To define his search strategy (especially as the user becomes more knowledgeable), the user will want ever-increasing capabilities with even more simplified forms of expression until, in the long run, the user will be able to converse verbally in natural language with the computer.

LITERATURE CITED

(1) Boyer, C. B., "A History of Mathematics," Wiley, New York, N. Y., 1968, p 633.
(2) Augustus DeMorgan, 1806–1871.
(3) Boole, G. S., "Investigation of the Laws of Thought," Dover Publications, New York, N. Y., 1854.

Data Security†

M. J. ORCEYRE

IBM Corporation, Poughkeepsie, New York 12602

Received December 13, 1974

Data security is a rich and complex subject dealing with the protection of the computing capability from all threats to its continuity. Some fundamental elements of the process of achieving a reasonable, prudent measure of that protection are considered.

Data security is a rich and broad topic, one that is of increasing concern to data-processing-oriented people at all levels, and is receiving more (and more formal) attention from users and manufacturers alike. The definition of data security should indicate the scope, complexity, and pervasive nature of the subject: it is simply the safety of data (and necessarily also of the system) from improper disclosure, modification, or destruction—whether these are accidentally or intentionally caused. Note that this definition applies as well to the manual as to the EDP operation. Note, too, that it is a global definition; no threat to the continued well-being of the operation is excluded. I intend this to be a complete—however brief—discussion, so you may expect me to deal with all sorts of situations that threaten the safety of data, ranging from technologically complex penetrations of computing systems by highly trained intruders, to earthquakes, to coffee spilled into the machinery, and so on.

† Presented in the "Conference on Large Data Bases," sponsored by the NAS/NRC Committee on Chemical Information, National Academy of Sciences, May 22-23, 1974.

In fact, when you yourselves deal with data security, keep this breadth of scope in mind. After all, to protect data you must understand the threats to those data. To protect data completely against all threats is an unrealizable goal; to protect data to some reasonable extent against reasonably predictable and probable threats is a prudent and practical goal. To accomplish the latter, you must undertake a risk assessment, which involves gaining the clearest possible understanding of the nature of that which you must protect and also of the relative probabilities of the events that threaten the well-being of what you must protect. If you do not have this understanding, you cannot assess risks; if you cannot assess risks, you cannot prudently determine protective measures; and if you cannot prudently undertake protection you cannot know that you are protected.

It is my intent to review some fundamental elements of the process of achieving protection. One of these elements is a clear understanding of the need for protection, and this need—which you all must have to one degree or another—springs from a number of sources:

(a) Data is a major asset of the enterprise. To the extent that it was costly to collect and organize, and would be costly to replace or reconstruct, it must be protected.

(b) The availability of data and the means to operate upon it is to some extent critical to individuals, to departments, even to entire organizations. Loss of such availability will have measurable effects. These should be known and adequate protection provided.

(c) Data may be classified, or proprietary, or personal and private. Protection that satisfies applicable legal/regulatory requirements, and that demonstrates concern for "fairness," must be provided.

(d) Data handling capabilities often present the temptation of personal profit to those involved, particularly in relation to the complexity of the capability. It is neither responsible nor fair to entrust too much capability to people when there exist no means of control or audit of their activities. By extension, it is neither responsible nor fair to subject a larger number of people than absolutely necessary to suspicion of improper activity when, in fact, improper activity takes place.

(e) It makes no sense at all for those entrusted with the management of the operation or any part of it to fail to act prudently in protecting the operation, or to be unable to demonstrate to more senior management that they have acted prudently in assessing risks and undertaking reasonable precautionary measures.

There is simply not yet enough hard information gathered in one place to enable a truly scientific study of the causes of losses to the data-processing community at large, or to special segments of it, but there is enough information to make valid general statements. I would like to do this now, because the general conclusions that can be drawn, although not blessed by years of research, are (most probably) accurate and may be significant to your efforts to take reasonably prioritized steps to protect against reasonably prioritized threats.

The available information on losses suffered by the data-processing community indicates that the causes can be grouped into six categories, which are as follows (in the order of decreasing significance):

1. Errors and omissions by authorized people of unquestioned loyalty and honesty but who occasionally lapse in judgment or competence account for over 50% of all dollar losses. This category includes, to give an idea of its scope, operator mismounts of removable media, programming flaws, keypunch errors, coffee spilled into terminals, and management failure to provide adequate backup for recovery from inadvertent damage to data. Losses in this category are very frequent and range from very small to quite large—many thousands of dollars.

2. Abuses of their capabilities by dishonest authorized people, usually for personal profit, are the second largest category of loss. These are people competent not only to perform misdeeds but also to conceal what they have done. If we estimate that the improper activities known by us to have occurred are 10% or so of all such activities (in the case of bank fraud and embezzlement the FBI estimates that 15% or so are reported), we can see that not only is there more dishonesty than generally suspected but also that it is often not discovered, at least for long periods of time. Compared to the first category, however, these losses are relatively infrequent but on the average are extremely large; some estimates have the average loss as high as $600,000. We estimate that about 10% of losses are due to dishonest employees.

3. Losses to fire are next largest on our list. I would like to point out here only that the technology of fire detection and quenching is much further advanced than are the general skills in applying this technology. Too frequently the machine room and tape library themselves are equipped with the best available protective gear while the real threat is that fire will break out on the unprotected floor below where paper is stored (thus baking the machine room) or on the floor above, from which the quenching water will stream into the machine room and destroy equipment. The simple message here is that common sense is called for in dealing with protection against fire. I should note here that fires almost never start inside computing equipment, and hardly ever anywhere else in machine rooms that are well-managed, where good housekeeping is the rule.

4. Actions of disgruntled employees have fourth place among these categories. Total losses here are relatively small and cases are relatively infrequent, but the per-incident loss tends to be high. Concern is definitely warranted, but then again it is most often a lack of management concern, sensitivity, and good practices that gives rise to these cases.

5. Water damage unrelated to fire is the next largest cause of losses, but again this is relative. Hurricanes, tidal waves, and floods are understood phenomena; don't put the system where they are likely to do damage. Defective plumbing and leaky roofs upstairs, and inadvertent discharges of sprinkler systems do cause considerable damage and must be considered in any complete program of protection. Wet machinery won't run and may never run again. Wet punched cards cannot be used. Wet continuous forms cannot be fed into a printer. And so on.

6. The last category is a catchall, encompassing all threats not included in the first five groups, that accounts for something less than 3% of all losses, but much more than 3% of all publicity. Natural catastrophes, physical attacks, and technologically elegant intrusions into systems by resourceful and determined outsiders are all included here.

Having given the general presentation of the threat categories, let me observe that there are other useful ways of considering threat distribution.

(a) I believe that people cause on the order of 70% of losses by dollar measure. Other causes amount to about 30%. By far, most of the people who cause loss are acting within their job-defined domains.

(b) I believe that accidental events account for at least 70% of losses. Intentionally caused problems account for no more than 30%.

(c) I believe that destruction and improper modification of data and other resources account for some 90% of losses; disclosure of data results is no more than 10%.

(d) Where people are immediately involved in a loss, the chance is less than 1% that they are outsiders.

To put this still another way, we observe that more than 70% of dollar losses are caused by authorized employees acting in the otherwise normal domain of their employment. People doing what their jobs necessarily enable them to do constitute a very serious threat to data security. The threat lists above can be recast as follows: (a) incompetent, careless, dishonest, and disgruntled insiders account for more than 70% of all losses (by dollar measure); (b) fire, water, and other physical damage accounts for less than 27% of all losses; (c) all other causes account for less than 3% of the total.

There is no way today to get a direct measure of the total loss to dishonest employees misusing systems or data for their personal gain. Data on the damage done by strangers, the 1% or so maximum noted above, are generally better than the data on dishonest use of ADP by employees, for the same reasons that the data on armed bank robbery are better than the data on embezzlement in banks. That is, banks, like all other organizations in both the public and private sectors, prefer not to disclose losses attributable to sources wholly within the organization. It is feared by such organizations that depositors, shareholders, policyholders, and voters might regard such losses as indications of management failure, which is often the case. Banks, however,

do not mind reporting, and are in fact required by law to report, armed robbery.

The loss to armed bank robbery was $21.7 million in 1973, but the loss to *reported* bank embezzlement was $158 million. The FBI estimates that the reported embezzlement constitutes about 10% of the total occurring. Other sources, including our own data on customer loss, suggest that the figure may be as high as 15% reported. Whichever is correct, it is clear that total bank embezzlement losses exceed $1 billion per year. The total loss to embezzlement for all industry areas is estimated to be about $6 billion, not all, of course, involving EDP.

It is important to note that, of the 10–15% of detected cases of embezzlement involving EDP which are reported to law enforcement agencies, only about one in five is successfully prosecuted. A body of admissible evidence is essential to a successful prosecution. This is not often available. Improvements in journalling capability—keeping automated records of significant activity on systems—will clearly ease the task of developing a body of evidence not only to guide management and assist auditing but to support prosecutions and thereby provide a significant deterrent to people who would misuse systems.

One further note: experience and available information also indicate that there is no significant tendency for employees to step outside their normal job domains or to subvert today's built-in user-isolation and access-control features in committing fraud or embezzlement or in otherwise causing real losses to system owners. A number of conclusions can be drawn from this:

1. Today's user-isolation and access-control features are inadequately granular; they do not permit enough distinction among system elements to enable customers to define job scopes in adequately fine detail.
2. Employees most readily perceive and seize opportunities for misuse of those capabilities with which they are most familiar.
3. Employees perceive little risk of apprehension if they misuse the system, and little risk of significant penalty if they are apprehended.

Several principles emerge from all this, and these lead me to my concluding discussion of EDP system features and functions that, properly employed and integrated, can yield adequate protection in most environments. First, the principles.

Security features must be such that:

> The system owner is able to bestow upon his employees the least privilege necessary for each employee to do his work.
>
> It is difficult for a user to misuse a capability that he is authorized to have.
>
> The employee can be held personally accountable for his activities on the system.
>
> The employee perceives a high risk of apprehension if he attempts to misuse his privilege.
>
> They are useful in identifying and reducing the incidence and impact of errors and omissions on the part of system users.
>
> They are adequate to address other, normally lower-probability threats that may be significant in a particular system environment.

And now for a closing discussion of specific protective measures that may be employed in unique ways by different installations to achieve predetermined reasonably adequate levels of security. The measures are supportive of four basic capabilities or attributes of the system and its environment: (1) identification of people and system elements, such as devices, software entities, and data objects; (2) authorization of interactions among these people and system elements; (3) surveillance of interactions or events, including record-keeping, or journalling, and subsequent inspection of the journals; (4) system integrity, or predictability, completeness, and correctness of the hardware, the software, the physical security, and the operating procedures.

I should note that an attempt to achieve state-of-the-art implementation of each of these classes of security measures would probably exhaust most normal installation budgets far short of the goal, and would doubtless result in security overkill at most installations.

Positive unique identification of individuals and of significant system elements is necessary in order to achieve meaningful control of system activities and records of such activities. Identification of people can be achieved by inspecting what they know (a password, for example), or what they possess (a magnetic striped credit card, for example), or what they are (a fingerprint or voiceprint, for example). The first of these is in wide use today, is well understood, is least expensive, and is least effective. Knowledge is too easy to share, wittingly and unwittingly. No change is observable to either the donor or the recipient, and they are indistinguishable to the identification mechanism. The donor may never know his password has been compromised. A physical possession like the mag-striped credit card, on the other hand, is more difficult to share unwittingly, or to counterfeit. If used for identification in conjunction with a password, then masquerade by a thief is much more difficult. This technique should satisfy most requirements. Physical attributes, on the other hand, afford the most credible identification of all, but to date the technology is such that implementation costs for most known testing methods are impractically high. Identification of devices is today generally of concern in teleprocessing environments, and is generally accomplished through use of hardware features available today with many terminals. Identification of software and data objects is done by simply looking at their system names—and relying on system integrity to ensure that the names in fact correspond to the objects.

Means of specifying and controlling operational domains of users and of selectively constraining interactions among people and system elements to the finest degree possible, consistent with the least privilege required for the work to be done, is one of the most important measures in achieving data security. To whatever extent you can, you should avoid enabling users of your system to do more than they have to, or ought to. We are seeing more authorization capability in systems today than ever before, and more (and more careful) use of these capabilities, and more interest expressed in them by more installations of different kinds, than ever before. And we expect this to continue as the need for security becomes more widely understood and accepted.

The objective of surveillance mechanisms is to ensure that installation management can detect certain events and certain user activities, some in real-time and others post facto, and can take appropriate steps for self-protection as a consequence of knowing of their occurrence. Above all, what these measures provide is a means of achieving strict accountability of people for their actions. To the extent that appropriate management is unable to determine—and to prove—what each system user has really been doing, it is in an unsafe position and the system is not auditable. Surveillance is the answer to the most common causes of dollar loss—improper and mistaken actions of people doing what they must be allowed to do to get their work done. The record of events, when properly analyzed, is useful in studying

error patterns and in discovering or at least correctly attributing improper activity, and this is a crucial deterrent capability.

System integrity refers to the proper functioning of the total system. In the case of hardware, this refers to the battery of error-prevention, error-detection, error-correction, and error-notification measures available today and under continuing development. In the case of software, this refers principally to freedom from surprises—the extent to which the software does correctly everything it is supposed to do, does nothing it is not supposed to do, and permits no unexpected behavior. Manufacturers are expending significant resources today in striving to enhance the integrity of computing equipment and of software.

Adequate physical security and thoughtfully developed and enforced operating procedures are most important aspects of data security. Achieving and maintaining appropriate protection from these elements of the security program is very much a matter of intelligent exposure analysis on the part of informed installation management, and the array of measures and techniques is well developed in the literature.

The Large Data Base File Structure Dilemma†

DAVID LEFKOVITZ

Moore School of Electrical Engineering,
University of Pennsylvania, Philadelphia, Pennsylvania 19174

Received December 13, 1974

This paper first presents a brief tutorial on the principal random file organization methods for handling two major applications—Transaction oriented systems and Information storage and retrieval systems. It then addresses a particular large data base dilemma, not satisfactorily resolved by any of these methods, and which is currently under active investigation. Two approaches to a solution are described. One is called the *hybrid inverted list;* the other is based upon an old technique called *super-imposed coding.* The former has been implemented and has recently been installed in an operational system. Some statistics related to file characteristics in this application are provided, but operational cost and performance statistics are not yet available.

Figure 1 classifies the principal types of information processing systems into two categories, *Transaction oriented systems* and *Information storage and retrieval systems.* Transaction systems are characterized by files that contain a diversity of record types, in which the records have relationships to one another, frequently referred to as Master-Detail relationships. These records may be arrayed in the same or different files. The structures that describe these record relationships are called trees, rings, and networks. Information storage and retrieval systems, on the other hand, tend to have records of similar type but which develop relationships among specific fields or components of these records called *keys.* The structures of files that support these various relationships are called Sequential, Indexed Sequential, Indexed Random, Mapped Random, Multi-list, and Inverted List. These two classes of file structures and the data processing that is characteristic of the respective systems present different requirements to the system designer. The data access requirements of the Transaction oriented system are considerably simpler than those of the Information storage and retrieval system, but accuracy with respect to the maintenance of data integrity is usually far more critical. The Information storage and retrieval systems, of which the bibliographic systems are one particular type, tend to deal with larger masses of diverse data types, and access to information tends to be more manifold; however, whether or not particular data items or records are or are not accessed for a given request tends to be less critical than in Transaction oriented systems.

†Presented in the "Conference on Large Data Bases," sponsored by the NAS/NRC Committee on Chemical Information, National Academy of Sciences, Washington, D. C., May 22-23, 1974. Supported in part by Contract NO1-CM-33719 from the Division of Cancer Treatment, NCI, NIH, DHEW.

Figure 2 graphically illustrates the tree, ring, and network file structures. The dots in the figure represent individual records. In the *tree,* a particular record may exhibit a master to detail relationship to one or more other records, and these other records may be in the same or a different file. Frequently, the detail records in a randomly organized file system will reside in a different file from that of the master records. The illustration shows a single master record at the top or root of the tree pointing to three detail records. One of these detail records is itself a master record and points to two other detail records. The two detail records at the third level will, in general, have a different record format and may themselves be contained in yet another file. The characteristic of the tree is that a given detail record may be pointed to by at most one master record. The *network* is a more general graphical structure, and no particular rules concerning the number of record relationships that can be established to or from a particular record exist. In the network, therefore, the master detail relationship may lose meaning. Any record may relate to any number of other records, and any number of records may relate to a particular record. The *ring* is a network in which all of the details emanating from a particular master are chained together, and the chain is closed on the original master. This forms "rings" of details hanging from a particular master, and a particular detail may itself, as a master, spawn its own "ring" of details, as shown in the illustration.

Figures 3 and 4 schematically illustrate the six basic file structures for keyed record relationships, characteristic of Information storage and retrieval systems. In Figure 3 the Sequential file is shown as a one-dimensional array of records. Access to each record is sequential. That is, in order to access, say, the third record, one would have to access