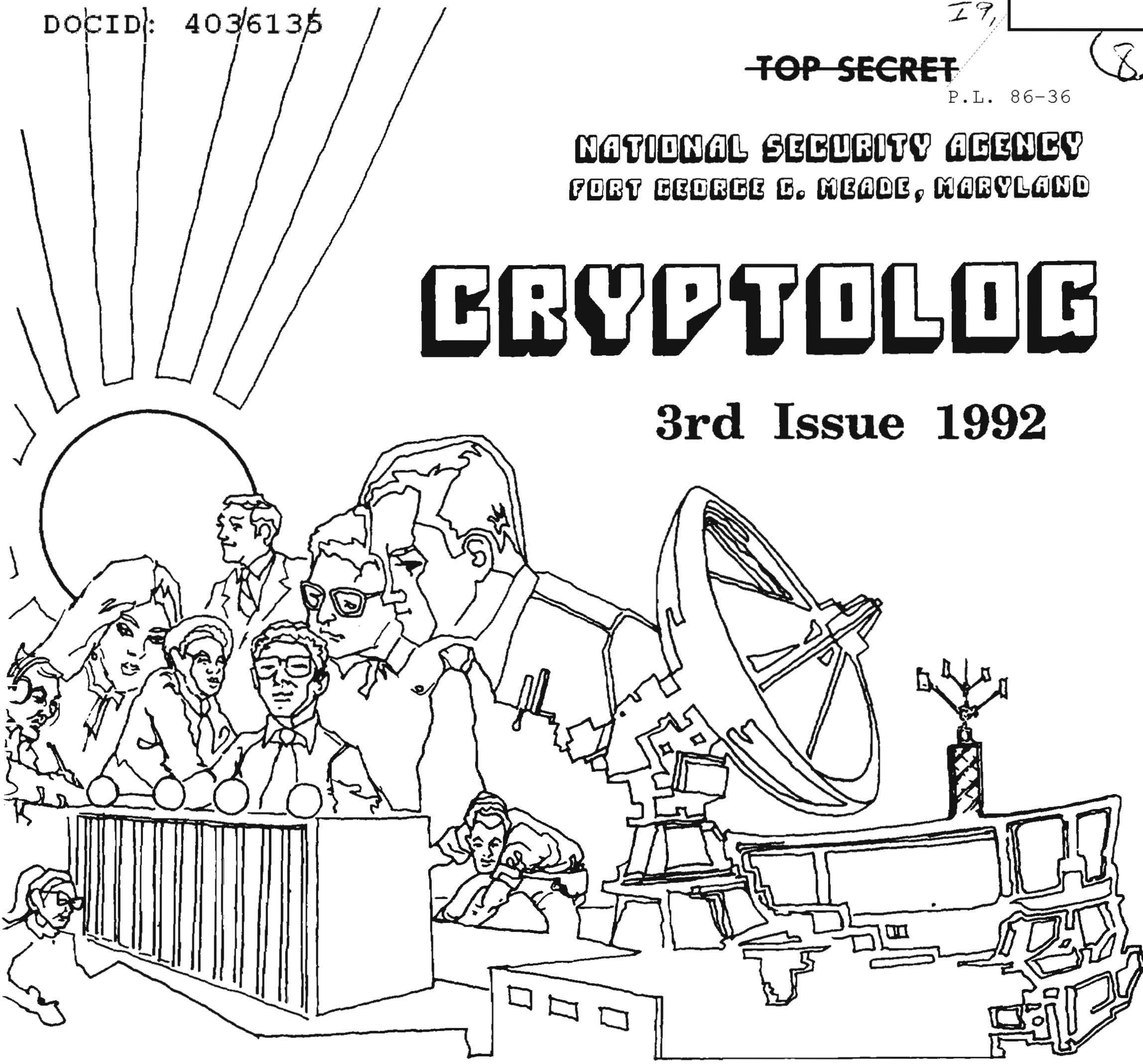
NATIONAL SECURITY AGENCY
FORT GEORGE G. MEADE, MARYLAND

CRYPTOLOG

3rd Issue 1992



CRYPTOLOG

Published by P05, Operations Directorate Intelligence Staff

VOL. XIX, No. 3 . . . . 3rd Issue 1992

PUBLISHER . . . . [redacted]

To submit articles and letters, please see last page

For New Subscription or Change of Address or Name

MAIL name and old and new organizations and building to:

Distribution, CRYPTOLOG, P0541, OPS-1

[redacted]

P.L. 86-36

Please DO NOT PHONE about your subscription or matters pertaining to distribution

The Third OPSEC Conference

George Jelen, D2

Good morning. It is my pleasure to welcome all of you to the third annual NSA/CSS OPSEC day.

We have all heard of the "Age of Enlightenment" and many of us have lived through the "Age of Aquarius." Well, I'm here to proclaim this as the "Age of Operations Security." After being around since the days of the Vietnam War, Operations Security, or OPSEC, has finally come of age. I attribute this to two causes: first, partly as a result of the 1988 presidential directive, more and more people have become exposed to OPSEC and have learned of its potential. This has led to the application of OPSEC to many different kinds of operations and activities. And second, people are beginning to discover that as they are forced to shrink their budgets for security, OPSEC and its methodology can be very useful in choosing where to focus their security protection and in applying that protection uniformly and consistently.

There are many indications that the interest in operations security is picking up. OPSEC tracks are now being offered as part of several major security conferences, including the International Security Systems Symposium and the conference of the American Society for Industrial Security. The National OPSEC Conference itself has been growing in attendance, and the membership in the OPSEC Professionals Society is continuing to expand steadily. The last couple of years have also seen a developing body of OPSEC literature. Here at NSA, a COSC has been established for the OPSEC field; and NSA/CSS OPSEC Association has been instituted and is aggressively planning activities; and there is a panel actively working to create criteria for professionalization.

Initially applied to military operations during the Vietnam War, OPSEC is now seeing application in a rapidly widening set of circumstances and activities. The Secret Service is applying it to personnel protection, the FBI to law enforcement, the defense community to weapon system acquisition, the Coast Guard and the Customs Service to drug interdiction, and the Intelligence Community to clandestine and covert operations. At a national level, we are now examining how we might employ OPSEC to help protect critical economic information as well. This latter application is clearly the growth area of the future.

OPSEC has also found varied application here at NSA. We have applied it to site closure plans, to SIGINT support to military operations, to sensitive acquisitions and procurements, to personnel movements, to financial transactions, to logistics shipments, and to counternarcotics support. The purpose, in every case has been to improve the effectiveness of our operations. We have come to understand that whenever there is some advantage in concealing our intentions, OPSEC proves quite useful.

There is a second reason why operations security has been receiving more attention of late, and why this is likely to continue. With the collapse of the former Soviet Union, the U.S. has entered a period during which our expenditures for security are being challenged. The changed world situation, and the altered security threat that it has brought with it, have caused many to question the continued need for security protection. The question I hear all the time is, "Where's the threat?" It's a reasonable question. Most would answer that there is still a threat, but that it is reduced and it is directed differently — focused more on economic and technology secrets than on military secrets. Motivated by a need to reduce our expenditures and encouraged by this generally-accepted reduction in threat, resources for security are being cut. As this happens, OPSEC and its methodology become ever more useful.

As resources applied to protection are reduced, if we are not to sacrifice too mush security in the process, this reduction needs to be done sensibly. Two basic premises of the OPSEC discipline are that not all information justifies protection and that not all vulnerabilities are worth correcting. We need to have some way to distinguish between what really needs protecting and what does not. Presently, we are spending far too much money trying to protect information that is either not worth protecting, is already known, or is fundamentally unprotectable. This makes no sense and we can not afford to continue to do it. This is where operations security can help. The more important it is to be selective in the application of our security resources, the more relevant the OPSEC methodology becomes. As resources for security grow more scarce, it becomes that much more important to apply them where they are most needed and where they can do the most good. The proper application of the operations security discipline and its methodology can be extremely useful in sorting out what most needs protection and in making sensible decisions about where and where not we can best afford to cut resources.

Our decision process needs to weigh the importance of the information, the motivation and the capability of our adversary, the ease with which that adversary could obtain that information, and the risk of leaving the secret unprotected versus the cost of protecting it. All of this is precisely what the familiar five-step OPSEC process does, and all OPSEC professionals know how to do it.

The OPSEC process imposes a rigor that can be profitably employed in many security resource decisions, sometimes with dramatic results. For example, when the U.S. was preparing for the arrival of Soviet inspectors as a result of the Strategic Arms Treaty, teams went around to a number of contractor facilities and Air Force bases looking at what special security arrangements would be required. Applying the OPSEC methodology, the teams were able to reduce the projected expenditures for security by more than seventy million dollars.

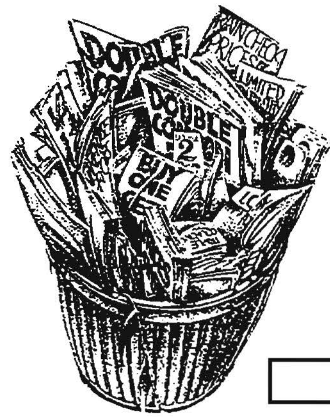
Once we have identified the information that is in most need of protection, it is equally important that we apply security resources consistently and completely — that we do not spend money on a robust lock for the front door and leave the back door unbolted. Here again, OPSEC can be helpful.

The various traditional security disciplines do a pretty good job of protecting against direct disclosure. But our secrets can be revealed indirectly as well as directly, and OPSEC complements these other disciplines by seeking also to protect those same secrets against indirect disclosure as well. Failure to consider ways in which an adversary might piece together the same secret from bits and pieces of information could mean that we spend a considerable amount of money in security protection and give away the secret anyway. Without operations security, the envelope of protection is incomplete.

We in D2 understand that our organization can not perform OPSEC for NSA or the CSS. OPSEC, by its nature, cannot be centralized. To implement OPSEC effectively, it has to become part of the normal way everyone of us conducts our daily business. In other words, the NSA and the CSS will only have effective operations security when every single person understands it and practices it within his or her organization. Everyone has to know what it is, how it works, what are its goals, methods and mechanisms. That is what this day is all about. It is aimed at deepening your understanding of OPSEC so that you might be more able to apply it to your own activities. Or employing the words of this year's theme, the day is intended to help you keep your mission on target through the sound application of OPSEC principles.

I am pleased to see you all here and I hope that many of you can remain for the rest of the day.

A QUANTITATIVE APPROACH FOR EVALUATING A FACILITY'S TECHNICAL SECURITY



[redacted] R551

This Article is classified ~~SECRET~~ in its entirety.

P.L. 86-36

Center for Security Evaluation (CSE) has been working on various initiatives to support the decision process in evaluating technical security at facilities. One recent effort, referred to as the Aggregate Countermeasure Effectiveness (ACE) model, helps to provide a quantitative measurement that integrates the multitude of factors that impact technical security. The model takes into account the value of the targets in the facility, the capability of the threat, and the overall effectiveness of the associated countermeasures. A prototype of the model has been developed and is currently being evaluated by CSE, with promising preliminary results.

HISTORY

In early 1991, the Standard Division of CSE decided that a new method was needed for evaluating the technical threat to a facility. At that time, a means was needed to help decide on the best combination of Technical Surveillance Countermeasures (TSCM) for the conditions particular to each facility. To this end, CSE set the wheels in motion to develop an analytical methodology that could be used to help make more informed decisions in two specific areas, namely:

- in establishing TSCM standards and policies, and
- in evaluating overall TSCM investment strategies.

THE INTENT OF THE EFFORT

Simply stated, the goal was to create a mathematical model that would tie together all of the factors that impact technical security at a facility. Results from the model would provide a quantitative "barometer" that could be used to compare the degree of technical threat at one facility with that of another. The model had to be understandable, represent the overall state of technical security, [redacted]

EO 1.4.(c)
P.L. 86-36

The intent was to support decisions in three general areas. First, to help evaluate alternative sets of TSCM, either by looking at strategies specific to one facility or policies applicable to U.S. facilities in general. [redacted] Finally, the model would be used to examine proposed resourcing concepts either at a particular facility or across a selected set of factilities. The model was to address such [redacted]

EO 1.4.(c)
P.L. 86-36

EO 1.4.(c)
P.L. 86-36

EO 1.4.(c)

[redacted]

[redacted]

THE SOLUTION: WHAT THE MODEL DOES

It was decided that this new methodology (referred to as the Aggregate Countermeasure Effectiveness model, or just ACE), would be created in two stages to expedite the development process, and to provide a quick turnaround product that could be readily evaluated for its usefulness. In the first stage, a prototype microcomputer-based software package would be built for a reduced portion of the problem. If the prototype is determined to be of value, then a production version (with full capability and user-friendly features) would be created.

To develop the model, technical security at a facility was analyzed as a general flow problem using basic engineering principles. [redacted]

DESCRIBING THE PHYSICAL PROBLEM

For this "flow model" concept to work, a series of mathematical expressions were developed that

[redacted]

EO 1.4.(c)
P.L. 86-36

represented the relationships between the amount of sensitive material potentially compromised at a facility and the multitude of factors that determine its state of technical security (the expression at the bottom of exhibit 1 illustrates a few of the factors). The first step in translating the real world problem (exhibit 2) to a "mathematically oriented model" was to define the physical relationship between the targets at the facility (conversations between people, workstations, copiers, computers, etc.), the TSCM intended to protect the sensitive material processed by these targets [redacted], and the various techniques used by the threat [redacted]

[redacted]

HOW ACE WORKS

There are three basic parts to ACE (exhibit 4). First, the value of the sensitive material processed at the facility is estimated for each target based on the volume of activity and the worth of the material. Second, the effectiveness of the TSCM is computed for each path by summing the attenua-

EO 1.4.(c)
P.L. 86-36

DOCID: 4036135

tion provided by each individual countermeasure. This, in turn, is degraded by the state of operational security at the facility, both in terms of the adequacy of the preventive maintenance program and the level of P[3] (personnel, procedural, and physical) security. The natural attenuation provided by the controlled access distance is then added, resulting in an estimate of the combined effectiveness of the TSCM for each path. Finally, the capability and intent of the threat is brought into play.

[redacted]

HOW THE RESULTS ARE USED

[redacted]

EO 1.4.(c)
P.L. 86-36

[redacted]

WHERE ARE WE NOW

EO 1.4.(c)
P.L. 86-36

In October 1991, the prototype version of ACE was completed. The microcomputer-based software package is being evaluated by CSE and run through a variety of scenarios. The purpose is to perform a preliminary "sanity check" (does ACE provide reasonable results? Can we explain and understand what is happening?).

To help evaluate the model, sample cases have been run using "rough estimate" data readily available on three facilities. The preliminary findings are very encouraging. The sample runs have provided realistic results and have highlighted interesting observations. [redacted]

EO 1.4.(c)
P.L. 86-36

At present, ACE is going through an extensive shakedown to validate its consistency and to examine its sensitivity to the precision of the input data (number of targets, attenuation of the TSCM, risk of threat discovery, etc.). Although a [redacted]

CSE is now using the prototype version to evaluate technical surveillance countermeasures. Actual data from an OCONUS site was input into ACE prototype. The results were verified by CSE and the R55 team. A brief of the results will be presented [redacted] P.L. 86-36 in September. Actual data from OCONUS sites will continue to be collected and input into ACE.

Golden Oldie

The Dying Programmer's Lament

The programmers spoke in a fading voice,
"That diamond shows it's a multi-choice
And a loop is seen where the line returns
And a block is cleared, but my fever burns:

"O, pin me not to a completion date
When the machine is down and the assembly late,
O, think of the errors I might have made
And the debug sessions so long delayed,

"The symbolic deck with the cards transposed,
Subroutines opened, that were not closed,
The card Operations dropped on the floor,
The Sponsor's shadow beyond the door.

Yet I fought the fight. It will surely run
At the next debug, or the next but one."

So we buried him on his completion date
When the machine was down and the assembly late.
And we sighed for the errors he might have made
And the debug sessions so long delayed.

[redacted] P.L. 86-36

reprinted from bits & bytes

C4 MACHINE INFORMATION PROCESSING BULLETIN,

Vol I, No.2
June 1965

EO 1.4.(c)
EO 1.4.(d)
P.L. 86-36

EO 1.4.(c)
EO 1.4.(d)
P.L. 86-36

DOCID: 4036135

EO 1.4.(c)
EO 1.4.(d)
P.L. 86-36

EO 1.4.(c)
EO 1.4.(d)
P.L. 86-36

3rd Issue 1992 * CRYPTOLOG * page 14

~~SECRET SPOKE~~

EO 1.4.(c)
EO 1.4.(d)
P.L. 86-36

EO 1.4.(c)
EO 1.4.(d)
P.L. 86-36

Writing in a journalistic style

Somali bitterness toward the United Nations runs deep; derived from what is perceived as its neglect of this country following the fall of the Siad Barre regime in January 1991. When the United Nations sent an envoy here, Algerian diplomat Mohammed Sahnoun, he publicly criticized the world body's performance.

Boutros-Ghali reprimanded and later replaced Sahnoun, who, in his six months in the post, had become widely appreciated here. His replacement, Iraqi diplomat Ismat Kittani, has run into trouble in his first weeks. His meeting with Aideed has been called a "disaster" by some of Aideed's advisers, who say Kittani spent the first few minutes criticizing Aideed for ignoring proper protocol in arranging the venue.

Boutros-Ghali is a former minister of state for foreign affairs in the Egyptian government, and Egypt is held in disdain here for having supported Siad Barre's dictatorship until Mohamed, as Somalia's interim president. Thus, Egypt—and by extension, Boutros-Ghali—is seen by many of Aideed's supporters as being biased and incapable of playing a mediating role.

Shortly after Bush made his surprise offer to send U.S. combat troops to this war- and famine-wracked nation, Boutros-Ghali was quoted as telling an interviewer that a U.N. trusteeship might have to be established here. Aideed's people describe that as a recolonization.

Aideed's ready acceptance of the American troop offer, after he had previously balked at allowing more U.N. peace keepers into the country, was seen as the warlord's way of trying to undercut the United Nations and bring America in as a political mediator.

Within days of the first U.S. Marines' arrival in Mogadishu on Dec. 9, Aideed and Ali Mahdi made their first public peace after 18 months of

P.L. 86-36 [redacted]P054

(U) During the past year, we have witnessed a myriad of local, national, and international events. "The Birds Move Into Camden Yards," "Hurricane Andrew Sweeps Devastation Across Southern Florida," and "American Troops Victorious in Persian Gulf" were but a few of the headlines that captured the attention of diverse audiences. What do each of these stories have in common? Well, no matter where you read, heard, or saw these stories, the information presented to you was planned, published, and distributed to you in a journalistic style and format.

(U) The journalistic approach to reporting tells a story. It seeks to provide the reader or viewer with a direct presentation of the facts, with minimum interpretation. You remember, the kind of reporting delivered by Walter Cronkite when he signed off at the end of each Evening News show with, "And that's the way it is . . ."

(U) The headline gives the reader or viewer a quick synopsis of the main element in a story. It attracts the reader or viewer and serves as kind of a marketing hook to get people to buy newspapers, watch television, or read NSA SIGINT reports. Never thought we were in the marketing business with our SIGINT products, did you? But indeed we are!

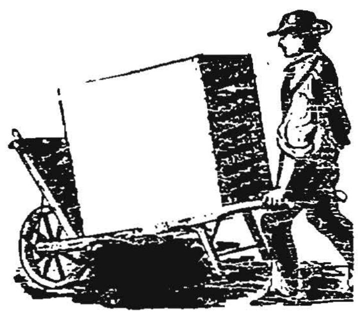
~~(FOUO)~~ The headline also allows each reader or viewer to pass over a story and move onto another of greater interest. Each reader and viewer has individual preferences. So in essence, every SIGINT customer constantly "tunes in and tunes out" topics of the day. Additionally, each customer also has individual preferences for receiving information. Some prefer to read, while others prefer to hear or see information presented to them. That's why newspapers, magazines, radio, and television have all flourished.

~~(FOUO)~~ Who determines story topics, what will be published for print and produced for television, and what priority each story will take? Well, in the media, a team led by a managing editor or executive producer works together to answer these tough questions. The team oversees day-to-day production while at the same time planning for the next issue or program. Quite often media senior managers get involved in these collaborative editorial group discussions to provide additional input into pre-publication planning, special event coverage, and post-publication reviews. The objective is to serve the customer's thirst for information in an accurate, timely, and highly competitive fashion. The journalistic process of identifying stories, prioritizing topics, and overseeing production never ends.

~~(FOUO)~~ The journalistic topic experts are the analysts, reporters, correspondents, writers, and producers who have intimate first hand knowledge of developing and breaking stories. The journalistic process of reporting empowers these individuals to produce and meet deadlines. The journalistic process also supports follow-up reporting and timely correction updates.

(U) The journalistic process of reporting is much like a total quality management (TQM) process. Journalism products are geared toward satisfying diverse customer demands for information, the editorial boards are collaborative process action teams, and the staffs and support mechanisms of the journalistic process are empowered to put out the best quality products possible while striving for the competitive edge. The journalistic process of reporting and TQM go hand in hand in satisfying both the customers' demand for excellence and the desire of journalists to contribute and be an integral and essential part of the process.

Reducing the Burden of Maintaining Software



P.L. 86-36

[redacted] P043

Seven suggestions will be proposed to help minimize the burden of software maintenance. If these methods are implemented, the resulting system will be less of a burden on the data processing department that supports it. One method used to improve the maintainability of software is for the project manager to set explicit maintenance objectives and priorities. Another is to use quality-enhancing techniques and tools that will improve maintainability and will improve the system's documentation. Establishing activities that assure quality, choosing a maintainable programming language, and establishing file systems that are independent from the programs are three design concepts that will greatly improve maintainability. Finally, I will give suggestions on how to contract for a maintainable system when that system is supplied by a vendor.

SETTING MAINTENANCE OBJECTIVES

The best way to build anything into a system is to ask for it; therefore, setting explicit maintenance objectives and priorities will improve the maintainability of a software system. By setting maintenance objectives and priorities, we make it possible to influence quality, and therefore maintenance. According to the Weinberg studies each project team will achieve the highest goal set for it; whatever system qualities are stressed, those are the qualities that will be delivered. The manager of the project team must tell the maintenance programmers what quality factors their system is expected to contain. A maintainable program will possess most of the seven quality factors (i.e., reliability, understandability, testability, modifiability, usability, efficiency, and portability) but it is seldom possible to cost-justify all seven for any one program. Another block to all factors receiving equal importance is the fact that some of these factors are in conflict; one must be sacrificed to improve another. A common example is the efficiency/understandability conflict. Again, the project manager must rank the seven quality factors, and emphasize to the programmer which factors are to be delivered for the specific system in question.

QUALITY-ENHANCING TOOLS

The use of quality-enhancing techniques and tools will improve the maintainability of a system. The quality-enhancing techniques and tools that will be discussed here are: structured techniques, restructuring, reformatting, and prototyping.

Structured techniques

Structured techniques should be utilized in all phases of a system; structured techniques improve understandability (i.e., quality) and therefore, reduce the overall software costs. Structured techniques standardize the style of the software system this standardization helps programmers become familiar with the system more quickly; their understanding of the system is more complete; the quality of the system is upheld. Structured code is the introduction of standardization into the program's form. Modularization is the traditional approach for enhancing quality; the theory here is that independent pieces will simplify the program's understandability, and the maintenance task. When modularization is taken another step further, structured programming results. Structured programming is a modularized system that represents a logical and hierarchical relationship. Coupling is low; the execution flow among modules is simple and easy to understand.

Restructuring

The objective of restructuring is to improve the understandability of the existing software system and, therefore, improve its useful life. Complex, error-prone, and frequently changed modules are prime candidates for restructuring. A caution: we must be careful not to take poor, unstructured code, and develop poor, structured code. The goal of any structured technique is to improve the quality, understandability, and maintainability of the system.

Reformatting

If a manager considers restructuring too risky, there is still a technique that can be utilized without the introduction of restructuring. Reformatting, the introduction of indented code, standard label conventions, one instruction per line, and standardization of keywords, are much less risky than restructuring, and have also been shown to improve understandibility.

Prototyping

Prototyping is a quality-enhancing tool that can be just as useful in maintenance as it is in the development cycle. It provides usability since it allows the maintainer to understand the needs and requirements of the end user. Prototyping is most useful during life cycle support, since it minimizes possible user misinterpretations. It is a valuable, though underutilized quality-enhancing tool.

IMPROVING DOCUMENTATION

Documentation, along with quality, is one of the factors that contribute to the difficulty of maintenance. If a system's documentation is improved, so will its maintainability. Documentation can be classified into four types: user, operations, program, and data documentation. Since each performs a specific function, maintainability is improved if all forms are present. This is primarily because the maintenance programming team will be able to find exactly the information it needs, without extensive searching.

User documentation provides instruction on the use of the system's programs. Instructions are provided for the entering of data, interpreting of output information, and reacting to error messages. Usually, this consists of a user manual, but a more usable approach is an online documentation system. This online transaction would be available on the end users CRT. High quality user documentation promotes system usability. When user documentation is poor, misinformed people report errors. These hypothetical errors are really differences in the interpretation of the system's functions. Most managers agree that proper documentation is a good idea, but they seldom require it.

Program documentation is used to help the maintenance programmer understand the internal structure of the program. It is also used to demonstrate the software's coupling (how the modules interact within the system), the systems interactions with the operating system, and within other software systems. Program documentation includes external program specifications, program flowcharts, source code commentary, and system flowcharts. The most useful documentation is high-level. This documentation explains the overall purpose of the program and describes the relationships among the various program components. External (separate from the source code) documentation is necessary. HIPO diagrams and Warnier diagrams are two examples of external documentation. Low-level documentation (line-by-line descriptions) is not necessary. The best way to provide low-level documentation is through the use of self-documenting programming languages. Program documentation is produced in the design phase; problems occur since it is rarely updated to reflect maintenance changes.

Data documentation is needed in addition to program documentation. There are two ways to document data: data modeling and a data dictionary.

- The data modeling provides a graphic model identifying the structure of the data and its functional dependence .

- A data dictionary lists all the forms of data used, their definition, how they are used, where used, and who is reponsible for them.

Data documentation needs to be included, but often is not.

~~FOR OFFICIAL USE ONLY~~

QUALITY ASSURANCE

Another method for improving the maintainability of software is the establishing of explicit quality assurance activities. Commonly called quality assurance audits, these activities are important in maintenance, as well as in development phases of a system.

Briefly, there are four types of quality assurance audits:

- ***Checkpoint reviews*** are used in the development of new software. They are used between the phases of development, to check the development work as it progresses; again, the sooner the error is found, the less expensive it will be to correct.

- ***An acceptance review*** is a special checkpoint review that occurs between the development and production stages. An acceptance review, sometimes called project turnover, is the last chance to ensure maintainability before the software becomes operational and becomes the responsibility of the maintenance staff.

- ***Periodic maintenance audits*** are used on operational software to recognize changes in quality. Since software systems are not static, periodic audits are necessary, and any changes in quality should be investigated. Because of the importance of overall system understanding in the maintenance function, it is helpful to have the maintainers involved in the development of the system. Ideally, they should be involved not only in the maintenance acceptance reviews, but also at other checkpoint reviews.

- ***The benchmark audit*** is used on packaged software, and it will be discussed in the section on improving maintainability in packaged software.

Audits are the most powerful techniques for introducing and preserving software quality. Managers often feel that quality assurance activities are not necessary, especially in a maintenance activity. They usually state that audits cannot be cost-justified, and quality cannot be measured. Although audits seem like a tiring, time consuming activity, they actually reduce the time allocated to maintenance; the earlier an error is discovered, the more easily recoverable it is, and the less costly the error is to correct. We have already discussed seven factors that contribute to quality (i.e., reliability, understandability, testability, modifiability, usability, efficiency, and portability) and by measuring these, we can measure the cumulative quality of the system. This should convince managers that they should practice quality audits.

CHOOSING A LANGUAGE

Choosing the proper language can affect the program's maintainability. Low-level languages are difficult to learn and understand, as are programs coded in a low-level language. 'C' language is easier to understand than Assembler, because Assembler is not structured well and does not support meaningful variable names. Recognizing this, the project manager should choose the highest-level language possible. Fourth generation languages should be utilized when possible.

Fourth generation languages are easy to use, understand, and modify. Therefore, development and maintenance in a fourth generation environment is faster. Since most fourth generation languages are non-procedural (defining what is to be accomplished, not how) these systems can be modified by the end users and may not require the help of an analyst. Even if analysts are required, they can obtain the results faster by the use of a fourth generation language than if they were to write out program specifications.

It is generally thought that the use of fourth generation languages will help the maintainability of a system. The quality of the system is improved by the use of fourth generation languages. Understandability is better, since the language is simplier and less complex. The code must be structured; since unstructured code is not obtainable with the use of a fourth generation language. These languages often are equipped with self-documentation capabilities, thus reducing the maintenance difficulty.

FILE STRUCTURE INDEPENDENCE

I think we all can agree that a file cabinet is a flexible way to store data; data can be added, removed, or rearranged without major difficulty. This is not always true of a computer file system; computer data files often have flexibility problems. They often have high levels of redundancy and inconsistency can occur when data are in different stages of update. Since computer files are inflexible requests for a new data arrangement can take weeks or months.

Seemingly trivial changes to a file system can set off a chain reaction (or two). The goal is to avoid changing programs when a file's physical structure changes. The data base environment was introduced to solve this problem. The intent of the data base environment is to isolate the program from changes in the structure of the data files. This environment allows the program's perspective to be different than the physical record. The programmer perceives a "make-believe" record and therefore does not worry about changes to the data's physical structure; the programmer can represent all data structures, and dynamically create new access paths. One of the factors (discussed earlier) that contributes to the difficulty of maintenance is the dependence of the programs on the file structure; the data base environment provides program and file structure independence.

PACKAGED SOFTWARE

Package software applications should be purchased with maintenance in mind. Again, the principle applies: you get nothing unless you ask for it. We should plan for the future maintenance when a software application package is purchased. Each vendor's reputation should be examined; will the vendor provide adequate service? The customer is dependent on the vendor to perform maintenance on the software; the software must be kept in good working order. This is especially true when the customer does not have access to source code and/or is not allowed to modify the software.

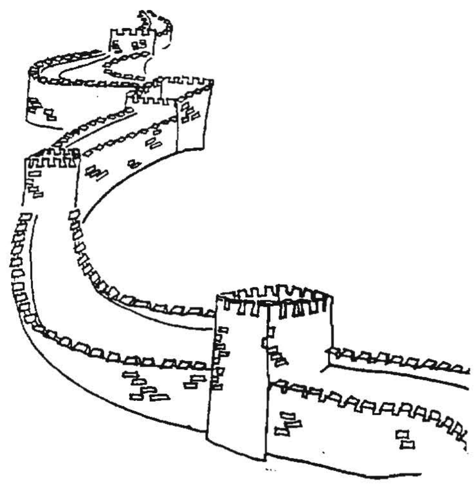
Current users of the packages under consideration should be sought, and potential users should discuss the package's performance. It is best to conduct a benchmark audit to ensure the quality of the protential software package. A benchmark audit is a program test conducted by the proposed users to ensure that the software package performs according to their expectations. If the vendor refuses to allow a benchmark audit as part of the selection process, another vendor should be considered. The conduct of these benchmark audits are an important part of the acquisition of an application package.

Once a package has been selected, an appropriate contract must be written. The user must have a good contract to ensure that the vendor will keep the software package in good working order. A contract contains some important specifics. A reliability clause is needed to guarantee adequate maintenance support; it should specify how quickly the vendor can be expected to respond to a request for service, how quickly the software error will be repaired after it is reported, the methods for correcting software errors, and the penalties incurred by the vendor if the reliability guarantee is not met. The software lease agreement should assure the customer knowledge of, and access to, new releases of the software package. The vendor should provide a renewal option in the contract. This clause allows the purchaser continuing maintenance even though the supplier has a short contract term. The final clause is the termination clause. This provides the purchaser with the source code in the event of vendor bankruptcy.

CONCLUSION

"Nice suggestions," you may be saying, "but I have heard all of them before. It just is not that easy to change." Although these statements are true, I feel that, with a little effort, the time and money that is put into maintenance each year will be reduced. Although all seven of these methods will help reduce the burden of maintaining software, the system should be considered when determining the appropriate method. Generally the first three methods will make the biggest impact on maintenance. Setting explicit maintenance objectives and priorities, using quality-enhancing techniques and tools, and improving program documentation will improve the quality and understandability of the computer system. Using these suggestions alone can reduce the burden of maintenance significantly. Sometimes, understanding the problem is half the solution. ☐

CETA



P.L. 86-36

[redacted] A631

~~(S-CCO)~~ Sometime this year a data tape will be delivered to B6 containing a data set of some 250,000 Chinese-English dictionary entries. Development of a local software package will make those entries retrievable by English term, or by Chinese term in characters —both long and short forms—or by Standard Telegraphic Code from networked workstations in B Group. The delivery of that tape will mark the high point in the Agency's long, and at times difficult, participation in an organization called CETA. For a period of six years I was the Agency (acting for DoD) representative to the CETA group. Following is a brief history of its 20-year existence.

COMPOSITION

(U) The Chinese-English Translation Assistance (CETA) Group is an organization of persons from the U.S. Government and the private sector who share a common interest in the development of Chinese-English translation aids. Its purpose is to promote cooperative efforts among linguists, lexicographers, computer specialists, and others in compiling and updating computer-stored, machine-readable, Chinese-English dictionaries and glossaries; and to make available the products of those efforts to its members, and insofar as feasible, to other users of Chinese-English translation aids.

(U) Membership of the CETA group consists of agencies and individuals interested in Chinese-English translation. There are no formal requirements for membership in the CETA groups, aside from an interest in the furtherance of its objectives.

HISTORY

(U) In mid-1964, the U.S. Government sent academia a list of government needs of gaps in China research. The government, in this case, was the China Committee of the interagency Foreign Area Research Coordination Group (FAR). The contact point in the academic word was the Joint Committee on Contemporary China (JCCC), established in 1959 under the auspices of the Social Science Research Council and the American Council of Learned Societies, and funded by the Ford Foundation. The late John Lindbeck of Harvard, then chairman of the JCCC, in his reply to the government initiative, made it very plain that the scholars he represented (covering most of the private China studies programs in the United States at that time) had sufficient vague indications of general interest in government-private cooperation to support such a project. He rejected most of the topics on the list, however, such as "Minority Groups in China" and "The Relationship between China's Foreign and Domestic Policies" as non-starters in developing truly meaningful government-private cooperation.

(U) Lindbeck focused instead on a lowly project near the end of the list: "Development of a Comprehensive Dictionary of Modern Chinese Terms." Here, he said, was an area where government and academia simply had to work closely together, if the need were to be met, and time was running out.

(U) Beginning in 1964, the former Foreign Documents Division (FDD), now under the Foreign Broadcast Information Service (FBIS), assembled materials for use in compiling a general-purpose listing of contemporary Chinese terms with English translations. One of the source materials was a Chinese-Japanese dictionary containing many Communist Chinese terms; others came from China, or from U.S. Government institutions including the Joint Publications Research Service and the Foreign Service Institute. The original FDD plan was to merge six dictionaries and glossaries with computer assistance, and to have the resultant compendium published ultimately in a thoroughly-researched, commercial dictionary form.

(U) A CETA Workshop in March 1972 was designed for just that purpose. It produced the guidance and momentum that led to joint funding of CETA by 9 of its 12 member agencies, beginning with FY 1973. In two days of panel and open discussion in the informal workshop atmosphere, the mixed group of 290 government representatives and 24 academic participants (from 20 private institutions) worked smoothly together.

(U) After a well-pitched keynote address by E. Raymond Platig, Director of the State Department's Office of External Research which had funded the workshop (including travel from all over the United States and from three foreign countries), the CETA dictionary effort was explained along with the varied projects of 16 other institutions that bore on the Chinese materials processing problem under discussion. At the end of the discussions, which many described with some emotion as the most fruitful they had ever known in a conference situation, four main conclusions were reached:

- Steady increase in the flow of materials from China on research on China was likely;

- Efficient processing of those materials in Chinese and English for both government and private use in research was a definable problem of considerable importance;

- The CETA man-machine system should be encouraged to attack not only definition of that problem but also its solution;

- Further development of CETA's "living" dictionary with appropriate purpose and quantity should retain first priority among the Group's efforts.

(U) The first run of the CETA dictionary was published and distributed in September 1971, but an oversight occurred that caused long and recurring argument within the group over the dictionary. The first run was quite crude and was intended primarily for contributors to look at and evaluate, with a view toward acquiring considerable guidance and input from them. But a caveat to that effect was omitted, and as a consequence, many people looked upon it as merely a crude compilation of reversed English-Chinese dictionaries of no great value; they overlooked the fact it was a printout from an online database that could be easily corrected, supplemented, and edited, as opposed to a typeset book with its attendant difficulties.

~~(S-CCO)~~ Agency representatives, however, were most interested in this database, especially in anticipation of the SEMESTER system. Eventually, after editing, a large subset of the database was loaded into the SEMESTER system and became the central core of computerized lookup capability for B Group Chinese transcribers. By 1986, many analysts were demanding more readily accessible and manipulable computer-resident dictionary database, so greater effort was expended to refine the CETA database and to make more readily and conveniently available to Agency analysts.

~~(S-CCO)~~ Now with the acquisition of the Chinese character set on the SUN system, and the opportunity to buy large, relatively inexpensive storage capability, all of the elements finally merged to bring the best of CETA onto the desks of Agency analysts.

▫

In commemmoration of the 50th anniversary of the World War II, 1991-1995

A History of Cryptology

Lambros D. Callimahos

The wartime contingent of linguists, codebreakers and cryptanalysts was an exotic mélange of multi-talented people, many of whom had already made their mark in the world. Some were foreign-born, some had already had prestigious careers in academia. There were missionaries and biblical scholars, mathematicians, classicists and linguists; there was a Russian Polish noblewoman who had been rescued by an American gunboat, a Hungarian prima ballerina who had been awarded a medal by the Pope, a professor of philosophy who was also a navy officer, cryptanalyst, and Chinese linguist, and there were others who later were to achieve fame: Eugene McCarthy, McGeorge Bundy, William Bundy, Edwin M. Reischauer; also columnists Joseph Kraft and Charles Barnett, journalist Al Friendly, and the elder statesman of bridge, Oswald Jacoby.

Among them were musicians, counted by the dozen. Scratch a cryppie, find a musician, or so it seemed. Peter Nickels, a conservatory-trained concert violinist doubled as the conductor of a dance band in the late 30's [redacted] *conducted concert versions of Gilbert and Sullivan at Arlington Hall—with a cast of equally gifted musicians;* [redacted] *who in 1962 received the 10th Annual International Jazz Critics' Poll's "Best Unknown Trumpet Player in Jazz",* [redacted] *a jazz piano player-cum editor, and so very many others.*

And then there was Bandmaster L. B. "Red" Luchenbach, USN Band #16, assigned to the CALIFORNIA. On December 7 his ship was hit and sunk. Meanwhile his bandsmen vanished. The fleet personnel officer refused to tell him where they were, but Red persisted, and eventually joined them at Combat Intelligence. Though it was to be a temporary assignment, the musicians proved too valuable to be replaced, and so they served as machine processing specialists and cryptanalysts. By the end of the war Red had a commission. Later, in civilian life, he represented IBM at the Navy Security Station.

The star was the late Lambros D. Callimahos, dubbed "the Paganini of the flute" by music critics, a flute virtuoso on the international scene in the 30's. At the prestigious Mozarteum in Salzburg, Austria, he was the youngest professor ever.

Also in the early 30's he developed an abiding interest in the history of cryptology. LDC was a collaborator with William Friedman on MILCRYPT I and II, wrote MILCRYPT III and IV and other papers and monographs. He developed and taught the seminal course in cryptanalysis, CA-400, that Gradus ad Parnassum *encompassing the cryptography of the then known cipher systems.*

This lecture was delivered at an unknown time and place and to an unknown audience.

This lecture was entitled on your programs, "The History of Cryptology." It's had several other titles, one of which was "26! or Bust." 26 x 25 x 24 . . . x 1. To those of you who use the slide rule, to 1 digit of accuracy, it is four times 10 to the 26th.

Cryptology is an ancient profession; in fact, the second oldest profession, one that abounds in drama and fascination, and one that has had a

P.L. 86-36

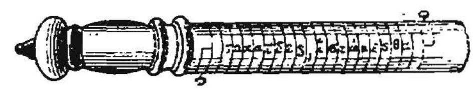
~~FOR OFFICIAL USE ONLY~~

profound impact on the turn of events in history.

We start off with communication. The Greeks did have messengers. In sending communications from one commander to another, they had the usual runners, but sometimes they wanted to be on the safe side and conceal the messages. For one idea, the Greeks would shave the head of the slave, and inscribe the message on the bare skin of the head. And then, you wait awhile—deferred message—and the slave was dispatched, not executed, but dispatched to the distant commander who would shave the head of the slave and read the message. If the message were particularly sensitive, it would be a one-time slave. These slaves are also, as you well know, normally distributed. The Greeks also used secret inks, the juices of various berries, milk, etc., brought out by heating. They used sputum and other effluvia.

On concealment, Hieronymous mentions that messages would be concealed in the belly of a hare, or inscribed on a wooden tablet and then covered with wax, or even inscribed on the leaves covering the putrid ulcers of disguised beggars. Also, on concealment, Sir Francis Bacon in 1623 wrote his renowned work, *The Advancement of Learning*, wherein he showed how he could disguise some of his innermost thoughts, thoughts which might be considered heretic in those days, by means of concealment within a covering text. It is the same system used today in our modern teleprinters. On the subject of concealment, there is also Boccacio, those of you who may have read the magnificent work in the original Italian, you know that Boccacio gave methods of information retrieval, that is, by means of the positioning of curtains or shades, but that's neither here nor there.

Cryptography was practiced, among other things, by the ancient Egyptians, only God knows why, because their hieroglyphic writing was enough of a disguise as it was, except for the learned class, the priests. And on the subject of priests, let us not forget that it is the ruffians and priests and scoundrels who have made great advances in cryptography and cryptanalysis throughout the ages. (I, myself am the son of a priest. Greek priests: a married man may become a priest, but priests cannot marry.)



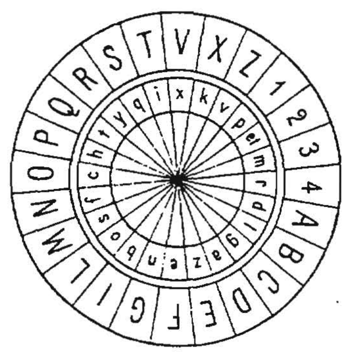
ΣΚΥΤΑΛΗ

400 B.C. Lysander used a device known as the scytale; in the dictionary it is pronounced "sitale" s-c-y-t-a-l-e, but I don't like "sitale", it is a Σ Κ Υ Τ Α Λ Η. The scytale consisted of a baton that the marshall carried, about yea long, tapered with a notch at one end, and the marshall would take off his belt, and wind the belt around the scytale and then write the message (in Greek, of course) across the bars; then he would take off the belt and dispatch that to the distant commander, who having a scytale of the same size, would affix the one end of the belt on the notch, wrap it around, and lo and behold, the Greek plain text came out. So that was our first transposition system. Also in 400 B.C., we have Aeneas the Tactician, who wrote a voluminous tome on the defense of fortifications, one chapter of which was devoted to the subject of cryptography. This was the first treatise on cryptography.

100 B.C. Julius Caesar, in corresponding with Cornelius, Baudus, Opius—whoever the devil they were, and others—used a simple scheme of replacing each letter in the Latin alphabet by one three removed from it, in other words, A plain would become D cipher, B plain would be E cipher, etc. But this was too much for Augustus' brain; he preferred a simpler scheme, where A plain was replaced by B, B by C, etc.

Now we jump to 1200 A.D. and the Papal States. By this time there were active cryptographic bureaus in the clergy, and the Papal States were the first ones to engage in systematic crypto correspondence. They often substituted vowel marks for the vowels, leaving the consonants unchanged. In other words, Mississippi would be spelt M.SS.SS.PP., where one dot might be an I, two dots an E, etc. In

1378, Gabriel de Lavinde of Parma, who worked for Clement VII—and if there are any historians here, they know Clement the seventh did not flourish at that time, it was the anti-pope Clement the Seventh. Lavinde wrote an SOI, a signal operating instruction, which is on deposit in the Vatican right now, and he gave many alphabets for which he had multiple equivalents for the single Latin letters.



Leon Battista Alberti's cipher disk

In 1470, Leon Battista Alberti wrote his *Trattati In Cifra*. He was an architect, painter, musician, writer on art, and the most universal genius of the First Renaissance. He invented, among other things, the cipher disc. In 1404, on Monday, July 4th—imagine what prescience the man had—Cicco Simonetta, an Italian, but nevertheless born on the 4th of July, wrote a little tract on cryptanalysis, the oldest tract extant, and he observed what you can do with Latin secret writing, by capitalizing on frequencies, on patterns of words, and on vowel identification. His methods were so good that they hold even today. He didn't get very far; in 1480, he lost his head. That was Monday July the 4th, 1404.

In 1531 David Trithemius, the German Abbot, wrote volumes I and II of a projected 4-volume work. He never finished the work but it was the first extensive treatise on cryptology. He was also a magician—after all, in my business, every little bit helps—and he was accused of being in league with the devil, and his books were burnt, but fortunately he wasn't.

Now the meaning of 26! There are 26! ways of scrambling the letters in the sequence ABCDEFGHIJKLMNOPQRSTUVWXYZ. You all know the number. If you set up here a set of letters for the plain component for the cipher, under **A** plain you may put anything you want—suppose you put an **X**. You had 26 choices here. Having an **X** here, there are only 25 choices here—make that an **O**. To gain some idea of the great size of this number, if you had 1,000 machines capable of testing 1,000,000 different alphabets per second, it would still take you over one billion years to go through the gamut of all alphabets. However, since you have a .5 probability of hitting before you reach half-way through, you can say roughly, you have the expectation after 500 million years.

One of Giovanni Battista Porta's cipher disks

You don't, though, in solving a simple substitution cipher, have to make all these trials. We do work on frequencies, the fact that the letters composing language are not equiprobable. We do work on repetitions, not only of letters, but also of digraphs, trigraphs, et cetera, and of long sequences that you hope are complete words. And we do work on patterns, like the word CEMETERY has an A B A pattern

Three Victorian amateur cryptologists: Sir Charles Wheatstone, inventor of two important cipher systems; Lyon Playfair, First Baron Playfair, who gave his name to one of Wheatstone's ciphers; and Charles Babbage, who solved many difficult ciphers.

for the 'e m e'. The word BATTALION has a A B B A pattern for the repeated letters 'a t t a', et cetera. So, quite early in the game, when people realized the weakness of simple substitution, they thought they'd get around it by having variants. So, where E is 13 per cent in English, instead of having one cipher symbol that would stand out 13 per cent, you'd now have 5 symbols that stand out about 2 or 3 per cent each.

All these remarkable ideas in cryptography were offset by even more ingenious ideas in cryptanalysis. We'll come back to codes in a minute. The departure was in the latter part of the nineteenth century—I'll skip over that for a minute, and go on to what happened.

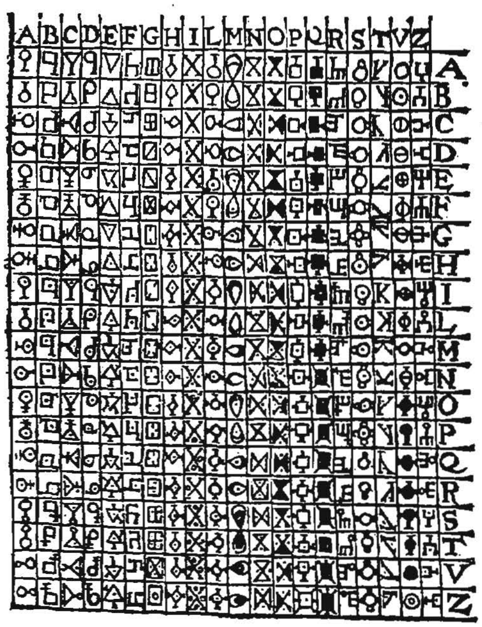
In 1914, a First Lieutenant Mauborgne published a paper put out by the Army Service School's Press, Ft. Leavenworth, Kansas, and the title, "An Advanced Problem in Cryptography and Solution," had to do with the Playfair cipher. (It was invented by Sir Charles Wheatstone, who did invent the Playfair cipher, but it was Lord Lyon Playfair who sponsored it in the Foreign Office, and gave it its name. But that's all right, because Wheatstone didn't invent the Wheatstone bridge. Wheatstone applied Cristi's dimension to the measurement of the bridges, so you see, it balances out in the end. On top of that, for those of you who might be interested, Wheatstone also invented the concertina.)

Instead of encrypting one letter at a time, in the Wheatstone Playfair, you encrypt two letters at a time. So EN is CP; this diagonal of the imaginary rectangle is enciphered by the other diagonal of the imaginary rectangle; and PC is NE.

This was a brilliant idea, because it suppressed the frequencies of single letters. However, Mauborgne did show them one method of solution. And he made good anyway, because he rose to be Chief Signal Officer.

Leon Battista Alberti (1404-72) considered the father of western cryptology. Alberti was also a talented musician, writer, artist, and athlete—a universal man.

In 1902, a chap by the name of F. de la Stelle—F I suppose was François, but it could be Felix, nobody knows—wrote in 1902 a book called *Traité de Cryptographie*. He mentioned in this book a system by which some letters might be enciphered by single digits and other letters by pairs of digits. This was an academic curiosity until the '30's, when this system took on major proportions, not as is, but with certain kinds of disguise. I mentioned briefly some ideas of simple substitution, and also the idea of transposition as exemplified by the scytale.



The earliest known digraphic system: Giovanni Battista de la Porta replaced each pair of letters with the sign at the intersection of their row and columns.

In the 16th and 17th centuries, transpositions came to the fore. In a transposition system, you retain the same language elements of the original message, except that you permute them about. Their identities remain the same, but their positions have been changed. The substitution systems, of course, the positions remain the same, but the identities change.

The first idea of polyalphabetic substitution was given really in 1470 by our friend Alberti in his cipher disc. In 1563, we have Giovanni Batista de la Porta. He was a physicist, inventor of the camera obscura, which was a predecessor of the Kodak, and he was a healer of the sick, just as we are healers of sick messages. He is known as the father of modern cryptography because it was he who pushed the idea of polyalphabeticity. In 1586, a French gentleman by the name of Blaise de Vigenère was travelling, and one of the things he picked up while in Italy was the idea of a square table to which he gave the name 'the Vigenère tableau'. He didn't invent it, moreover, he never said he did, but he gave his name to this idea with which you could perform true polyalphabetic substitution with no limitations.

That was in 1586.

In 1765, there flourished in the true sense of the word a truly great man, Giovanni Jacopo de Seingalt, otherwise known as Casanova; he was a remarkable scholar, a savant, a person who, well I can't say we should all emulate, but at least study because I'm sure we could all learn from him. One of his remarkable exploits was the fact that—thank you, somebody who was asleep is now awake—was his solution of a polyalphabetic cipher 100 years before the method of attack was announced to the world by the German Major Kasiski. Of course, Casanova was a privileged person; he managed to get cribs in the most remarkable places. But, the way this came about, I'll give you very briefly the background of all this—oh my gosh—the background will have to rest—if any of you wish to see me privately I'll give you the full story about what happened to the Marquis when he solved her cryptogram.

1863 was the date of Kasiski.

Now we come to codes. The ancients went from simple substitution to variant systems, to dissimilar writing wherein certain groups of characters, or for that matter, certain plaintext words, took on a new meaning, like 'ALMONDS' means 'I won't be home until Friday.'

So codes came out quite early in the game, but it wasn't until 1640, the great French cryptanalyst Orsignon that the two-part code came into being. In a one-part code, the code groups are arranged alphabetically and the vocabulary elements are in alphabetic order also. In a two-part code, one section, the encoding section, has the vocabulary elements in alphabetic order with a scramble of the code elements, and then another part with the code elements in alphabetic order for ease in decoding. The idea of an enciphered code soon followed, because of certain weaknesses of unenciphered code systems.

We might go back for a moment —to the earliest cryptanalysis. 550 B.C.—Please note the date—Daniel read a cryptogram for Bathsheba. It was a good stunt: there were symbols on the wall. He not only pronounced the symbols, but also gave the meaning. And since there was nobody there to contradict him, why, he made hay while the sun shined.

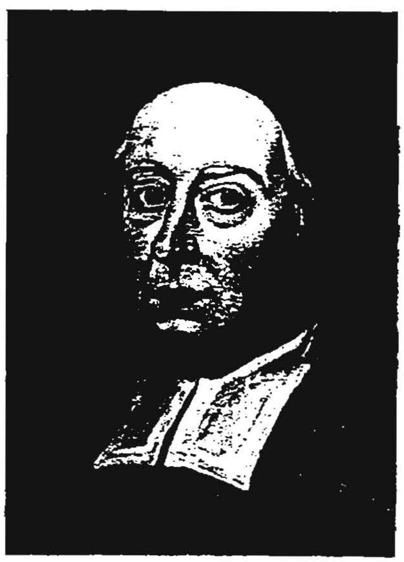
In 1510, Yulan deSoto of Venice solved many ciphers, including those of Charles the Fifth, which had been intercepted by the Papal Court and not solved by it. So he was an outsider, like an NSA consultant.

In 1525, the British lion begins four centuries of successful cryptanalysis.

In 1556, Alberti solved a message for the Portuguese ambassador who had lost his own code. This wouldn't happen today. I mean, if a diplomat loses his own code, he doesn't come to NSA asking for help.

1567. A prior in St. Peters, according to Vigenère, deciphered in less than six hours, a large page of cipher in the Turkish language, of which he did not even know four words. Having travelled, I can imagine what the four words were.

1589. François Viete, also know by his Latin name of Vieta, who as Privy Counsellor in France, solved a 500-group code of Spain's Philip the Second. Philip bitched to the Pope that France was using sorcery, so there was a miniature Pearl Harbor investigation on Viete, who to avoid conviction of sorcery, a capital offense, told all.



John Wallis, clergyman and mathematician, England's first great cryptanalyst

Antoine Rossignol, the father of French cryptology

That's 1589.

1595, in June, Viete, by this time was a good blabbermouth, in conversation with the Vatican ambassador to France, revealed that his ciphers were being read. That's the worst of all the sins that one could commit.

1626, Rossignol, who was remarkable for keeping his trap shut, began a cryptanalytic career with Louis the Fifteenth. When Louis was dying, he told the queen that Rossignol was one of the men most essential to the State. He was fifty-six years a civil servant. Brigadier Tiltman is second only to Rossignol. Wallis in 1645, the great English mathematician, began a career of five decades as an active cryptanalyst. (The Brigadier is in the middle of his fifth decade.) He was under Cromwell, and he solved the secret cipher of Charles the First. In 1689, still going strong, he solved the cipher of Louis the fourteenth, a 600-group two- and three-digit code.

In 1821, Jean François Champollion, a miserable little fellow, with a pale complexion, yellow skin, slaty eyes, was very gifted as a child. He told his brother, who later became his keeper—not in that sense, I mean his brother sacrificed his life for the more adept junior— little Champollion became imbued with the idea of reading the Egyptian hieroglyphs. He fell to work reading everything he could get his hands on, Modern Egyptian, et cetera—he was a gifted linguist—and in 1821, Champollion succeeded in breaking the secret of the hieroglyphs.

The Rosetta Stone

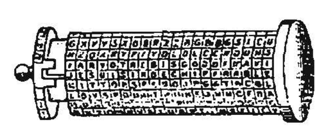
The hieroglyphs were broken by means of the Rosetta Stone, a piece of black basalt, three feet high two feet across, with three inscriptions, evidently parallel inscriptions, in Greek, Egyptian demotic, and hieroglyphs. He had what we call isologs, and he was very, very fortunate, because to recover an unknown language takes some cryptanalytic 'in', some cribs, and this was the height, where he had parallel texts in three versions.

We come now to cipher devices, again to our friend Alberti in 1470, the first one to dream up a cipher disk. Then by 1500, the idea of a cipher disk occurred to many many people, a disk, let's say of 26 letters revolving inside a frame of another circle of 26 letters. The idea was invented dozens of times.

In 1867, Sir Charles Wheatstone, inventor of the Playfair, thought up an ingenious little mechanical contrivance, to which he modestly gave the name, 'The Wheatstone Cryptograph.' It, for its day, was the most sophisticated idea in cryptography, but it lay buried in the archives.

Incidentally, although Sir Charles' idea was in 1867, in 1807, an American got there first, Decius Wadsworth, who was later chief of Ordnance, U.S. Army. He invented the same principal as the Wheatstone cipher device and executed it even better, mechanically. This device rested in limbo and the British knew about it, and in World War I, the British wanted to use the idea. Now the solution was known, the solution where the plain component was the normal sequence and the cipher component an unknown mixed sequence. But there was no known solution to the Wheatstone Cryptograph with two unknown sequences. But we'll come back to that story after a few minutes Let me do some more on cipher devices.

In 1891 a French major on the General Staff, a reservist, by the name of Etienne Bazeries, published in 1901 an essay showing the idea of a spindle with 20 disks permutable on the spindle, each of the disks had an alphabet a mixed sequence inscribed on the periphery, you arrange the disks on the spindle according to the key, set up the disks along a guide bar for your first twenty letters of plain text, and send the cipher text, every other row. At the other end, the deciphering clerk would have the same disks arranged in the proper order, he would set up the cipher text on the disks and look around. One and only one row



Bazeries Spindle

~~FOR OFFICIAL USE ONLY~~

would have plain text all the way across. He had a picture of this device in his book and across the guide bar was the sentence "Je suis indechifférable," "I am indecipherable."

Let's see, the book was published in 1901, it was 1891 when Bazeries thought of it, but it was in 1893 when a buddy of his, the Marquis d'Viares, another one who didn't have to work at cryptology, who was very skilled—I consider this piece of cryptanalysis the greatest piece of cryptanalysis, considering the age in which it was done. D'Viares showed Bazeries how he could arrive at a general solution of the device having only possession of the disks. This you would assume in time of war that the device would be captured.

It was Bazeries who published it first, then in 1915 an American army captain by the name of Parker Hitt invented the device again, He was the third inventor, because I didn't mention that the first inventor was Thomas Jefferson, but his papers weren't discovered until 1926, and he showed in his papers the idea of 36 disks on such a spindle, and that was a most remarkable cryptographic idea of its day.

Now we come to World War I. Radio.

To coin a phrase, radio is a two-edged sword, and you can go on from there. Every lecture you hear at NSA has that phrase in it, so it's wise to remember it. On the 26th of August, as you all remember, in 1914, the battle of Tonnebre, lasted three days; 100,000 men were killed or wounded, and missing in action. Two Russian commanders—Samsonov commanded the Second Army, Rennenkampf commanded the first Army. They didn't like each other, they had no contact with each other, not even on the staff level.

Samsonov went out in the field with the old code, but in the meantime Rennenkampf received the new code from Moscow, so he promptly destroyed his old code. He sent a message to Samsonov, who couldn't read it. He asked for a relay in the old code. He couldn't get it because Rennenkampf had destroyed his old code. So then Rennenkampf proceeded to send his messages in plain language. The Germans couldn't believe their ears. They read the messages where the Russians were supposed to be, they sent out reconnaissance patrols, found out that the Russians were there, then every day—it wasn't every day for long, just a few days, they waited for the day's take before they made up their battle plans, and in three days everything was lost. So that is a fine example of how things can go wrong.

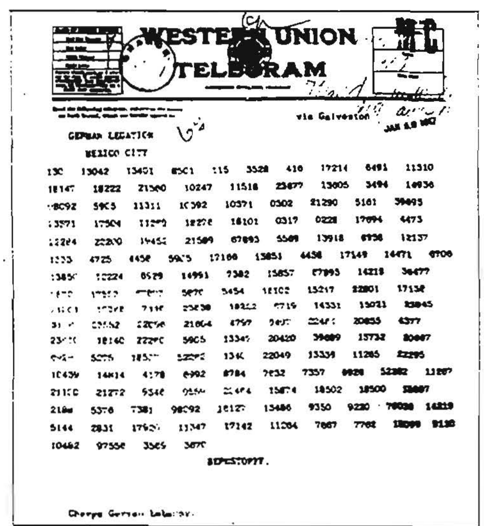
The next item I wish to cite in World War I, was 16 January 1917. The Zimmerman telegram. Perhaps the most famous cryptogram in history. Zimmerman, the Foreign Minister, sent a message to Bergstoff here in this country for transmittal to Eckhardt in Mexico. This telegram offered Mexico parts of Texas, Arizona and New Mexico if Mexico would enter the war on the side of Germany. Of course this is a dreadful abuse of the hospitality of a neutral country. The British solved the message, and conveyed it to us with some misgiving, because they didn't want to reveal what they had been doing. At the beginning, the anti-British faction here thought that it was another trick to get us into the war, but Zimmerman was queried in Berlin, and he admitted to having sent the telegram. Six days later we were in the war.

The U.S. Army went to war with three dreadful systems. One was the War Department Telegraph Code, which was safe because it was large enough so you couldn't hide it under your tunic, you'd have a bulge, and it had to have a certain amount of security to go with it because of the size, and the U.S. Army Cipher Disk, with reversed standard alphabets, the solution of which you'd do in the first lesson of Military Cryptanalytics Part II, and you could solve a single message, you could solve a portion of a single message.

In any case, that's what we had as the mainline system, and for emergency, we had the Playfair, but on every SOI every two or three days when they changed the key word, there was a warning: please don't use it because it's weak, insecure.

So that was on the U.S. Army side.

The Germans were more fortunate, they had double transposition, they had complex polyalphabetic substitution systems, and very ingenious combined substitution transposition systems known as the ADFGVX cipher. It was so remarkable that if I sent you a message, and then I had a power failure—not me, the station—when I sent only half the message, you still could reconstruct the entire message from the half that had been sent. You can't do that today.



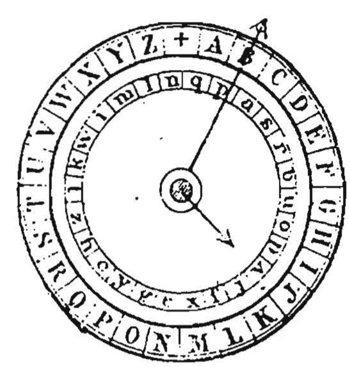
Zimmerman Telegram

The Germans were methodical so-and-so's, and instead of having check procedures to make sure that the system was working right, what they did was they sent aphorisms or parables early in the morning first day of change and they had various axioms, one of the most frequent parables was "Morgen Stunde hat Golden Munde," "the early bird catches the worm." So whenever there was a key change or a system change, the Allies would search through early in the morning for a short message, find it, put against it one of these parables, and in an hour or two they'd have that system.

Now we'll come back to the Wheatstone Cryptograph episode. The British wanted to introduce the Wheatstone Cryptograph into World War I, but were reluctant to because if the Germans captured it they too would have the indecipherable cipher—remember, there was no known solution if you had two mixed components. So it was judged unsolvable by the British, the French, and the Americans, both in the AF and in Washington, until someone remembered the group of people out in Geneva Illinois, industrial laboratories called Riverbank Laboratories, headed by a megalomaniac named George Fabyan. A private concern.

At this joint there was a young geneticist named William F. Friedman. What happened was that this chap Fabyan had a hobby of Baconism. He wanted to proved that Bacon wrote Shakespeare or vice versa, and he got a young lady in his employ by the name of Miss Elizebeth Smith, to read what there was in English (there wasn't much, God knows) to help him with his hobby. So then she had small classes in cryptology, and one of the gents was this William Friedman, who did very well, and so he later took over the section.

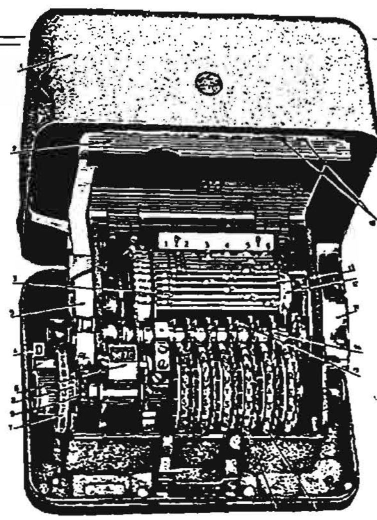
Anyway, Somebody remembered Friedman and the people at Riverbank so they decided to send six short messages out to Riverbank Laboratories, to see what could be done with them. Now they were only about twenty or thirty letters each, and that's really not a very good test, but Friedman by 'horsing around'—those were the words he used—managed to



The Wheatstone Cryptograph
Plain is outside, cipher is inside

scrounge out a cipher component. But he didn't know what to do with it, because it wasn't until 1923 that he discovered a very strong principle, the reduction to monoalphabetic terms.

So not knowing what to do, he called in Miss Smith, told her to sit down, make herself at ease, put on lipstick—I suppose she had some on before, but—anyway, and "I'm going to give you a word, you tell me the first word that comes to your mind." So he said 'machine', she said 'cipher'. I forgot to mention that the cipher component was based on 'machine' —a transposition mixed-alphabet based on 'machine'. So he didn't know what to do, but he asked Miss Smith, and she gave it to him. The plain component was based on 'cipher'. To keep it in the family, he married Miss Smith. The solution that went back to the British was quite embarrassing, because the method he described for getting the cipher component was so strange that even today you can't fathom it. It was part astrology, part cryptanalysis, that's hard



The Hagelin machine

enough to understand, but then about the plain component: I asked Miss Smith, and she gave it to me. So that's the story about the Wheatstone Cryptograph in World War I.

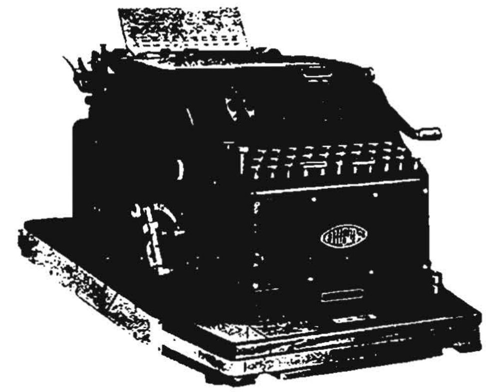
Then in the early twenties, a number of cipher devices came forth, the early Damm device, not damn, D A M M, Aubry Damm, operating with chains and gears and what not. This Damm firm was predecessor to the Achting-Bolotek Cryptograffe, which is the firm headed by the Swede Boris Caesar Wilhelm Hagelin. More of that later. It was Damm who really invented the first rotor. In 1924, a German by the name of Alexander von Kryha invented a gadget which had an astounding number of possibilities, like the number I wrote on the board before, the Kryha machine, and he got a buddy mathematician to explain in precise mathematical language, but even if you went through all the alphabets in time, blah blah blah, you could never go through them. However, the device is solvable and even on a single message.

This broke the man's pocketbook and also his heart; in fact, he committed suicide a few years back, perhaps because he couldn't push his device. This mathematician buddy of Kryha's came up with the statement that the number of possibilities with this machine was 1.4 times 10 to the 64th. And since the number of atoms in the universe, according to Sir Arthur Eddington is only 3 times 10 to the 74th, you can see it's a very favorable comparison indeed. Factorial 26 is only in the order of magnitude of 10 to the 26th, and here we're talking about 10 to the 64th.

B. C. W. Hagelin—a brilliant engineer, who came forth in the early 1920's with a whole series of devices. The first one was an ingenious contraption—fractionating principle. What it amounts to is this: you press a key on the typewriter keyboard, it sets into action two rods, mechanical rods, which are the left-hand the row components and column of a fractionating square; in other words, A is 1 6, then K is 2 0, for instance, this 1 6 would be enciphered separately, by separate schemes, recombined through this square, to get a single letter output. In other words, one letter input is fractionated into two halves, they go their separate ways, in a complex fashion, and join together in holy matrimony at the other end.

This was a wonderful idea, except when examined by William Friedman and his people in the early S.I.S days (Signal Intelligence Service of the U.S. Army), when this was found wanting. It's like a young girl who has had many offers of marriage, and never the knight on a white charger, or whatever it is.

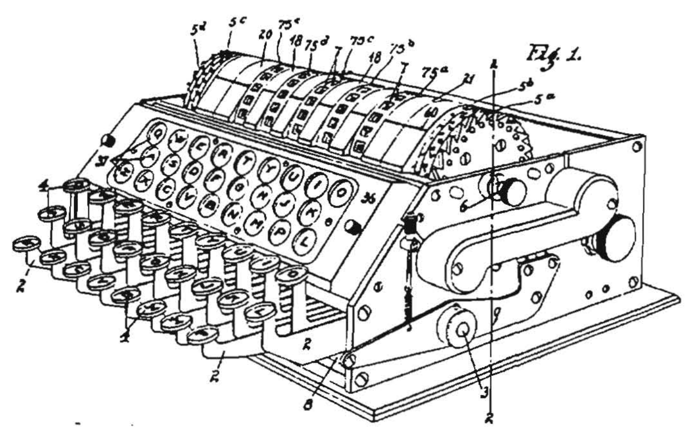
They broke up every cryptographic idea that had been proposed to them, or that they heard



The Enigma

of—they demolished, but they didn't come up with any good ideas themselves. They realized the enormity of their crime. They know no matter how complex something looked, the solutions may not have the same order of complexity. That's why in 1923 we adopted for the lower level cipher device, a tactical cipher device, the M-94, which is this disk device that harkens back to Jefferson days. This was in use until 1942. The solution is a childishly simple matter, especially since it was published in 1893.

Significantly, we're the only nation that adopted it; the French refused to have anything to do with it. We had nothing better until one of Hagelin's devices came along, accompanied by Hagelin, and this ingenious device about seven pounds, this big, gave rise to a very long key, the alphabets were known components, reversed standard alphabets, but the key was 26 x 25 x 23 x 21 x 19 x 17. So that's a very long cycle. It didn't mean that for a solution you had to wait for a message that long, but it did have certain fundamental weakness that nevertheless, in spite of the advice of Friedman and the Army, we did lease the North American rights and thousands of the devices were made by the L.C. Smith-Corona Company during the war.



Hebern Electric Super Code

The Hagelin machine was a very fine advance over the M-94, but still no good compared to what we should have as a major power. The better idea, now this was a tactical machine, something you could carry about with you, they even had the paratroopers landing with the 131 pins, their own two pins and the 131 pins of the M-209, and all 54 lugs zeroized on the assumption that after the poor devil hit the ground, if he was still capable of so doing, he'd get his key list out and do this while the tracer bullets were going over his head.

I'm sure there must have been more than one violation of security, where a man dropped with his machine already set up.

The first patent for a wired wheel machine was in 1918 in Germany. In 1923 the Enigma was patented. These machines involved discs, known as wired wheels there's switching combinators. With every encryption or decryption, one of these discs changes its position to give rise to a new alphabet. So if you had five of these cipher wheels, you have a potential of 11,880,000. The inventors of the first German rotor machines, we're not sure whether it was Scherbers or Korn.

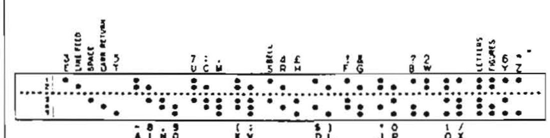
In 1923, an American by the name of Edward Hebern, out in California, also invented a rotor machine, to which he gave the name "the Hebern Electric Super Code." He interested the Navy in this device, it was a three-rotor machine, but he was asked to build a five-rotor machine. At that time the Army and the Navy were not talking to one another, and Friedman, who was on the Army side of the business, found out that the Navy was interested in buying some of the these. The Navy wouldn't let them have one of the machines, so the Army bought two. Then Friedman said the machine was weak. Actually he said so out of his hat, because he had no idea how to tackle it. It had a potential 90 billion cipher alphabets. Any self-respecting cryptanalyst should have thrown up his hands in horror.

Anyway, he found out that the machine the Navy had was not the same as the one that he had, because he asked for some letters to be encrypted with a certain setting across the wheels. So, to make a long story short, the Navy sent ten messages encrypted on this device, and Friedman, who was terribly unmathematical—in fact, he was just very, very lucky. He blundered his way into solution time and again.

In 1917 an American engineer by the name of Gilbert S. Vernam thought of a way for encrypting teleprinter signals. As you know, the teleprinter code is a binary code—I used to say two things taken five at a time, but mathematicians objected—it's five things each of which can take one of two states, on or off, whatever you want to call it. So the symbol for an **E** let's say, on a teletype tape is a hole and in the next four hole positions, there's no hole. That's an **E**. This is a T, et cetera. So Vernam thought up an idea of having a key tape prepunched random tape used to key a plaintext message tape. Then he thought he could do one better (this key tape, of course should be one-time, because then the security is infinite if the key tape was produced at random). But of course, it's clumsy, and then there's the difficulty of distributing the tapes, et cetera. So then he thought why not have two key loops, let's say of a thousand characters and 999 characters, so, since they're relatively prime, it takes 990,000 encryptions before you get back to the same arrangement of the two tapes. He even had an idea for less security—he proposed the idea of a single key loop. In MC II I wrote that the security of this scheme, however, is either negligible or only two or three times that amount. To us it makes sense.

It's interesting that Friedman, in 1919, meantime had been a good boy and gone overseas; he was a captain with the AEF in the code compilation section. After the war he went back to the Riverbank Laboratories, and there examined some traffic sent for test purposes by the State Department.

So the State Department was about to use this



Tape Showing Printing Telegraphic Code

two-tape idea. To be on the safe side, they sent a series of messages to Riverbank. Friedman and his staff worked two or three weeks on this traffic and got no place. They were sore in mind and body. This was the first time they worked for such a long period without solution. Everything they had come across they solved, and they couldn't understand what was wrong with them; there must be something wrong with them; they were all losing their buttons, and they wanted to leave him. It was a total of five weeks that they worked. One by one they wanted to leave and Friedman said "Give me a last chance. Let's go over our steps, we'll work one more week, if we get no place, we'll give it up." So they went over their steps (by the way, they had no tape printers for getting hard copy;

they had the teletype tapes sent to them, and they had to transcribe the holes, et cetera, with the Baudot code in front of them on sheets of paper.) And in the transcription, one chap left off a character that happened to be at the crucial spot. By the end of the week, they solved the system.

WORLD WAR II

We can go in the last five minutes, to World War II to give you some idea of the COMINT successes we've had. By the way, these items came out in a very dry article in *Time* magazine shortly after the war, when they had the Pearl Harbor investigation, when everything was revealed, to our detriment. That particular *Time* issue said that through MAGIC, which was the then cover name for our COMINT product, it enabled a relatively small U.S. force to intercept a Japanese invasion fleet and win a decisive victory in the battle of the Coral Sea, thus saving Australia and New Zealand. It gave the U.S. full information on the size of the Japanese forces advancing on Midway, enabling our Navy to concentrate ships which otherwise might have been three thousand miles away, and thus set up an ambush which proved to be the turning point of the Pacific War, directed U.S. submarines to sea lanes where Japanese convoys would be passing; and made possible the reading of messages from the Japanese Ambassador Oshima in Berlin, often reporting interviews with Hitler, giving our forces invaluable information on German war plans.

When the lid was blown, which was a dirty shame, because the world at large knew the cryptanalytic potential of the United States. In the report of the Joint Committee on the Pearl Harbor attack, there are two other quotes I would like to read you:

> With the exercise of the greatest ingenuity and all obvious resourcefulness regarded by the committee as meriting the highest commendation the War and Navy Departments collaborated in breaking Japanese diplomatic codes. Through the exploitation of intercepted and decoded messages between Japan and her diplomatic establishments, the so-called Magic, a wealth of intelligence concerning the purpose of the Japanese was available in Washington.

Another quote:

> Important diplomatic messages were intecepted, transmitted to Washington, decoded and translated, and disseminated with utmost speed. Not infrequently , they were in the hands of the authorized recipients of Magic in our government as soon as they were in the hands of the Japanese overseas. Many of the civilian and military personnel engaged in the handling of the Magic worked long hours, far in excess of those prescribed, without additional compensation or special recognition. Now this is in italics: T*he success achieved in reading the Japanese diplomatic codes merits the hightest commendation, and all the witnesses familiar with Magic material throughout the war, have testified that it contributed enormously to the defeat of the eneey, greatly shortened the war, and saved many thousands of lives.*

In fact, one estimate—they went through three or four sheets of foolscap to show this —and it was General Chamberlain, who was then G2 out in the Pacific—that one dollar spent on COMINT during World War II was the same as a thousand dollars spent elsewhere.

Editor's Note:

It seems evident that LDC in many ways inherited the many talents of his father, who was a renown theologian, a gifted linguist, and an authority on Byzantine music. Coincidentally, his father was my mother's professor of music and theology in Athens.

So it seems only right that I, in turn, became LDC's student —not in theology or music, but in that landmark course in cryptanalysis, CA-400.

MT Report:

Fujitsu's "Atlas G-160" Machine Translation System: A Preliminary Evaluation

[redacted]

In 1986, [redacted] began evaluating commercially available Japanese-to-English machine translation (MT) systems developed and manufactured in Japan. After a three-year evaluation period, which focused on appraising the quality of raw translations produced by nearly every system on the market, [redacted] decided in July 1989 to purchase Fujitsu's Atlas G-160 system, a personal computer (PC)-based system just placed on the market that April. [redacted] initiated purchasing arrangements in July 1989, and took delivery of the system in late December 1989. [redacted] has been evaluating all aspects of the Atlas's role in producing translations from a range of original Japanese text. This report is a product of that evaluation.

In order to present a meaningful evaluation of the Atlas G-160 system, it is first necessary to briefly introduce the system's components and each component's role in the document processing-translating sequence. This information is presented in Sections I and II. Just as crucial to the output quality as the machine itself is the type and quality of documents inputted for processing.

Section III of this report outlines the kinds of documents [redacted] is translating or hopes to translate using the system. The actual evaluation of the system as a whole begins in Section IV. Based on the background offered in Section I through II, the system's strong points and weak points with regard to specific [redacted] applications are discussed and the overall system evaluated. Recommendations for use of the Atlas system at [redacted] as well as some general conclusions about current machine translation technology in general, follow in Section IV.

While it specifically addresses the Atlas system and its use within the [redacted] environment, this report is being presented in the hope that it will help readers gain some insight into the machine translation process, enable them to measure current system capabilities against their own particular translation requirements, and ultimately place them in a better position to judge whether a machine translation system could be "right" for them.

I. The Role of Each Component in the Document Processing-Translation Sequence

The document processing-translation sequence using the Atlas system involves five basic steps:

- Japanese-Language inputting and creation of a corresponding Japanese-language file on the system (inputting can be done in any one of three ways-via keyboard, floppy disk, or OCR);
- pre-editing of the Japanese-language document in the newly-created file (optional);
- translating the file using one of two programs: "batch" or "interactive" translation;
- post-editing of the translated file (optional);
- outputting of the translated document as a printout in one of several formats: original Japanese text with side-by-side English translation, English translation only, etc.

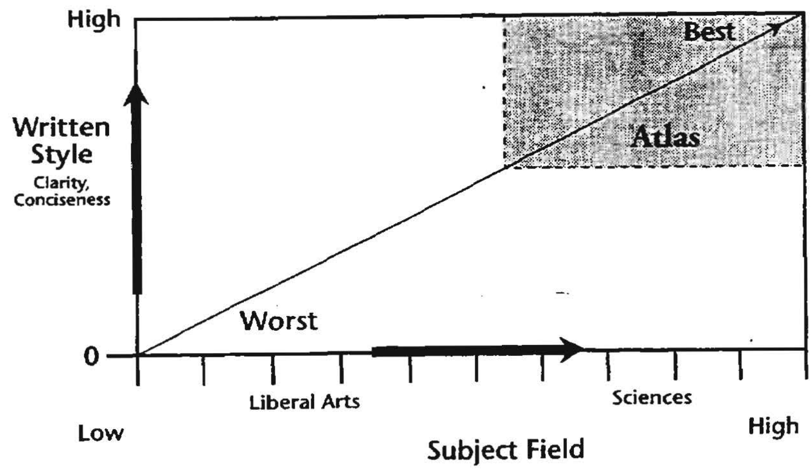


Figure 1. Document Types Atlas Most Effectively Translates

Document Inputting and Creation of Japanese-Language File

This step involves taking a Japanese language source document—either in printed form or on a floppy disk-and creating a new electronic version of that document in a Fujitsu "EPOWORD-G" file which subsequently can be processed by the Atlas program. This process can be done manually via the keyboard (basically, re-typing the original document into the computer), by transferring from floppy disk, or by use of the OCR.

• ***Manual Input:*** The Atlas incorporates a JIS key-board for the manual inputting (typing) of documents into the system. Conventional Japanese text consisting of kana (syllabary) and kanji (Chinese characters) is input as either Roman letters or kana. Additionally, keystrokes convert the letters or kana to the required kanji. The keyboard enables the input of 10,000 kanji (including the most common Chinese simplified kani), the English, Greek, Russian, French, Italian, Spanish, Portuguese, Danish, Scandinavian, and Norwegian alphabets, and hundreds of symbols used in mathematics, science and technology, and graphics compilation.

• ***Floppy Disk Input*** : The system incorporates an MS-DOS conversion function that enables input via floppy disk.

• ***OCR Input*** : This device, consisting of a scanning unit (which looks much like a desktop copier), a connection unit and related software, "reads in" hardcopy documents placed on the scanner, brings up a "bit image" of the document on-screen for the operator to selectively edit, and then produces a standard "EPOWORD-G" file containing the newly-created electronic document. The OCR can process typeset or word-processed Japanese text in a variety of fonts, pitch, and type sizes, although it cannot process text whose foreground or background contains color. It has a character recognition speed of about 20 characters a second and can process a page containing about 1,400 characters in roughly three minutes.

The OCR was designed to process Japanese text only, although Fujitsu is currently developing an English-text OCR. The device is simple to operate: the operator sets various processing parameters (vertical or horizontal text format, pitch, document name, etc.), pushes the START button, and feeds the document through the device. Each page's bit image is displayed on the monitor, allowing the operator to electronically adjust skew, erase spurious images, and select exactly how much of a page is to be processed. Each page is then processed by the OCR and compiled as a Japanese-language document. The operator then merges the pages to form a single document, and begins proof-reading prior to translation.

Pre-editing

Pre-editing, along with post-editing, is what separates a "raw" translation from a "polished" translation. It basically involves clarifying and simplifying complex grammatic structures into text more amenable to successful machine translation. Since the Atlas system provides "better" (i.e., more accurate and more understandable) translations from text characterized by relatively short, clear and concise sentences, pre-editing, even of the most basic kind, result in significantly improved end translations. On the Atlas system, the process includes, for example, artificially pluralizing nouns and pronouns, enclosing words and phrases in brackets to clarify governance and string boundaries, and replacing complex syntactic structures with simpler ones. Pre-editing can be carried out in two ways on the Atlas, manually or through a software program called "suiko" ("brush-up").

• ***Manual Pre-Editing***: This method, carried out either before or during the course of "interactive" translation (see "Translation Methods"), involves the operator simply going through the document manually and changing words and/or grammatical constructions which the translator perceives, based largely on previous experience, will represent a stumbling block for the Atlas translation program. Generally, the operator is concerned with only the most egregious errors, since minor changes can be made during post-editing.

• ***"Suiko" Pre-Editing***: This method employs a distinct program selected from the Atlas menu which is run against a document file; the program "flags" on-screen such items as incorrect kanji (not quite the same as a "spell check" function), the incorrect use of parentheses, overly long sentences, missing subjects, inappropriate kanji, and superfluous or ambiguous expressions. The operator then has the option of correcting the items "flagged" or ignoring them. Although extensive, the process is extremely time-consuming, especially if a multi-page text is being processed. In fact, [redacted] operators do all pre-editing manually during "interactive" translation.

Translation Methods

The Atlas system carries out translation in two user selected modes: "batch" and "interactive."

• ***Batch Mode***: In this mode, the system attempts to translate an entire document file in a single operation. No operator intervention is involved. The only information displayed on the monitor screen during batch translation is the total number of sentences in the original text and the number of the particular sentence being translated.

• ***Interactive Mode***:: The interactive mode involves extensive operator involvement; the operator selects exactly which sentences in a text are to be translated, pre-edits, re-translates, adds or changes words in the Atlas dictionary system, or post-edits as necessary.

Atlas Dictionary System

The Atlas system incorporates an 80,000-word standard dictionary and a user's dictionary as standard onboard programs. In addition, specialized dictionaries in over a dozen scientific fields, such as physics and information processing (see Section 2.B above, representing the full set of specialized dictionaries offered by Fuijitsu as of late 1989) also are available options. The dictionary system allows the operator to adjust English-Japanese word equivalents in order to improve translation accuracy-changing words, adding new words, deleting words, or adjusting word priorities.

In the user's dictionary, there are two ways to register word priorities. The first method is the so-called "kan'i" ("simplified") method used for registering nouns only. When a Japanese noun encountered in a text has not been registered (correctly) in any of the three dictionaries (standard, user's, specialized), the operator extracts the noun, inputs it onto the "kanji" dictionary screen, types in the appropriate English translation, selects the appropriate noun class (object, personal noun phenomenon, organization, etc.), and selects the appropriate English plural form. The operator then goes back to the text and re-translates the sentence(s) containing the newly-registered noun. Registration of other parts of speech is a far more complex process, involving, for example, careful classification of a verb's semantic and syntactic functions in both English and Japanese.

P.L. 86-36

Post-Editing Methods

The Atlas system allows for post-editing of the translated English language text either as an integral part of the interactive translation process (see above) or as a separate step once the English translation has been isolated in a separate English language text file. The software features a number of word-processing functions common to U.S. word processors, such as "move" and "copy" commands, but is still somewhat awkward to use.

Output Methods

The Atlas system can output translations in two forms—printout and floppy disk.

• ***Printouts***: Through the use of Atlas's "media conversion" program, processed documents can be printed out as original Japanese text only, original/pre-edited Japanese text only, original/pre-edited Japanese text with English translation (with or without Japanese-English vocabulary for those sentences the system cannot translate), or English translation only. This flexibility allows the operator to maintain a hardcopy of each stage of document processing/translating if desired.

• ***Floppy Disks***: Atlas output, i.e. English translations, can be output onto a floppy disk, but when transferred to English-language word processors, will contain only the English translation, and no part of the original Japanese source document.

II. Document Types For Translation On The Atlas

Documents processed thus far on [redacted] Atlas system are of two broadly-defined types: draft Japanese-language information reports produced by Japanese nationals [redacted] and, other Japanese-language documents ranging broadly from technical reports and newspaper articles to contracts and specification tables and charts. The differences in the two types of documents are important to consider, as they have a major impact on how the documents are processed on the Atlas system and what results are achieved (see SYSTEM EVALUATION).

[redacted] **Draft Information Reports**: These Japanese-language documents, produced in-house by [redacted] analysts, contain largely science and technology-related information extracted from original sources. After translation into English, these draft reports are edited and eventually published as final-form information reports. Most of these draft reports contain English-language glosses for the more complex specialized terms, and all reports submitted for machine translation are in the form of work-processor printouts. Because the reports are written by the [redacted] own analysts, the style in which the reports are written can be controlled to a certain extent. The draft reports are generally no longer than four or five pages.

Other Japanese-Language Documents: These documents, while in Japanese, come from various sources and generally do not contain English-language glosses for specialized vocabulary. Moreover, [redacted] has no control over the style in which these documents deal with communications and electronics and are at least ten pages long. All are submitted for translation in the form of xerographic copies or hardbound books.

P.L. 86-36

III. Sysytem Evaluations Capabilities vis-a-vis ASD Applications

Having briefly described the components of the Atlas system, the role each component plays in the MT process, and the types of documents meant to be processed on the system, a more meaningful evaluation of the Atlas's capabilities in processing [redacted] documents can not be offered. The central question for [redacted] of course, is this: Can the Atlas provide acceptable translations of the two broadly-defined types of [redacted] documents?

P.L. 86-36

The answer to this question is, essentially, this: since its operational introduction in February 1990, the Atlas system has shown itself to be generally NOT suited for producing POLISHED translations of [redacted] documents, particularly given the wide variety of topics and writing styles these documents encompass; rather, the Atlas is most efficient and effective in producing RAW

~~FOR OFFICIAL USE ONLY~~

1. ROSTERS OF PROPER NAMES (personal names, place names, company names, etc.)
   POOR without devoting a lot of time to dictionary registration; OUTSTANDING after completing dictionary registration
2. NEWSPAPER ARTICLES
   FAIRLY GOOD for articles on new products or technologies; TERRIBLE for articles on politics, economics (Atlas cannot handle journalistic Japanese w/out extensive pre-editing)
3. ABSTRACTS OF TECHNICAL PAPERS
   EXCELLENT; concise style, well defined terminology
4. TITLES OF S&T-RELATED BOOKS AND PAPERS
   OUTSTANDING; concise style, technical vocabulary probably already in technical dictionaries
5. CONTRACTS
   WORTHLESS; too many problems with long-winded "legalese" styles
6. USER'S MANUALS FOR EQUIPMENT
   EXCELLENT, unless the translation must contain both Japanese and English text (e.g. translation of Atlas User's manuals)
7. TABLES, CHARTS, MAPS
   GREAT translations, but cannot reformat back into original graphics the latest Atlas Ver.13 DOES have this capability, however)
8. "SHUKAN" MAGAZINE ARTICLES
   VERY POOR, generally; very colloquial style requires extensive pre-editing
9. UDC DRAFT INFORMATION REPORTS
   POOR TO VERY GOOD, depending upon style and subject of original document (see fig. 1)
10. NUMERICAL DATA (price lists, etc.)
   OUTSTANDING

Figure 2. Results of Testing Atlas' Ability to Raw Translate Various Types of Documents

P.L. 86-36

translations from documents written in a very concise style and dealing with as narrow a technical field as possible (see fig. 1). In other words, the Atlas system cannot provide acceptable translations of ALL types of [redacted] documents; instead, its strength lies in its ability to provide in a minimum amount of time a high volume of raw translations of certain kinds of documents (see fig. 2).

Underpinning this evaluation are the strengths and weaknesses of the Atlas system itself. There are many positive features to the system which streamline and simplify the processing-translation sequence. However, it is unfortunately the system's inherent technical weakness which place restraints on the Atlas's overall translation capabilities.

Atlas Strengths and Weaknesses: Their Impact on the NT Process

Among the system's strong points is its ability to input via both floppy disk and OCR (in addition to manual keyboard input). Also, its interactive translation mode and extensive dictionary systems add an important measure of flexibility and expendability to the MT work process. Overall, the Atlas menu system and operating environment are well-designed and user friendly, and the fact that the system is PC-based makes the system a very attractive alternative to larger, more expensive mainframe-based MT systems.

Its weak points, on the other hand, include technical limitations of the OCR in processing certain character styles and document types, the need to manually check all OCR output, the extreme difficulty of the pre-editing process, the fact that the translation program provides no translation whatsoever in cases where it finds the text too difficult, and the relative awkwardness of the system's English work-processing capabilities in the post-editing process.

Evaluating the system's strengths and weaknesses in the various steps of the MT process reveals the following specifics:

Input Process: More than 99 percent of the documents processed to date on the Atlas have been input via the OCR. The device has proven indispensable to the Atlas translation process, as it allows hardcopy documents to be "read into" the computer without having to type them in manually word-by-word. This, of course, saves time on the part of the operator and represents a major system strength. However, in many cases much of that time can again be lost when the operator goes to proof-read and correct the OCR inputted file. Depending upon document characteristics, the OCR device can reproduce text with an accuracy rate as high as about 95 percent (for original hardcopy that has good contrast and contains only Japanese text) and as low as 40 percent (for third or fourth-generation xerographic copy with spurious or colored images, poor skew, English-language text, or mathematical formulas, etc.). Any inaccuracies must be corrected through manual proof-reading (Japanese work-processing systems do not have highly-developed spell-check capabilities), which in the case of many OCR mistakes - can be extremely time-consuming and tedious.

Specifically, the OCR has difficulty with text containing any of the following:

- non-Japanese script (including English)
- handwritten text
- text with color foreground/background
- poor xerographic copies
- complicated mathematical formulas (Greek symbols, etc.)
- underlined characters
- free-form "designer" characters
- characters smaller than 7-point in size
- "Layered" characters not clearly separated (see fig. 3)
- unusual characters outside the finite set included in the OCR software

Although this seems long, most normal word-processed or typed-set Japanese-language text conforms to standards within the range of the OCR's capabilities. One solution to the OCR "dilemma" (i.e., quick read-in time but possibly time-consuming proof-reading and correction process) is input via floppy disk. If the source document can be initially created (such as a [redacted] draft information report) or downloaded (such as an on-line data base file, etc.) as an electronic document, then it could be input into the Atlas system directly via floppy disk, bypassing the OCR and accompanying proof-reading entirely. Having both the OCR and floppy disk as input options provides much flexibility and is a definite strong point of the Atlas system.

P.L. 86-36

Pre-Editing Process: If the Atlas system has an "Achilles heel", then this step is it. The reason being that it is the difficulty of this particular step which makes completing a truly polished translation such a painful task. This process is the most complex of all system procedures. It requires near-native Japanese-language proficiency to be done properly, and requires a considerable amount of time to pre-edit any but the simplest of writing styles. Moreover, the "suiko" or "brush-up" program that comes with the Atlas and which is intended to help facilitate the difficult pre-editing task actually does little to reduce the amount of time required. For these reasons, [redacted] operators have found that attempting polished translations-and doing a complete job of pre-editing-is generally impractical, opting instead for a raw translation. A quite satisfactory raw translation can very often be produced relatively quickly for S&T-type documents, so long as the source text is clearly and concisely written and characterized by a clearly defined set of vocabulary, which is often the case with such documents. If subjected to some simple post-translation English-language "post-editing", moreover, such raw translations can sometimes approach the quality of fully pre-edited polished translations.

P.L. 86-36

Translation Process and Dictionary Systems: The "batch" translation mode, because it does not allow for human interaction for editing purposes and cannot be viewed by the operator, has been found by [redacted] operators to be almost useless. The interactive translation mode, on the other hand, has proven to be the better method by far and certainly represents one of the Atlas's strong points. This mode offers the operator tremendous flexibility. It gives him/her the freedom to decide how much or how little to pre-edit, the ability to make dictionary changes in the midst of the translation process, and the option of re-translating any particular sentence as many times as desired. An operator, for example, would generally have the system translate the first few sentences of an input text. Based on how well these sentences were translated, the operator would then register any new words in the user's dictionary, pre-edit as necessary, and re-translate the entire text interactively, repeating this procedure. This procedure generally leads to a translation that, while not a truly polished one, is substantially better than a raw one.

Post-Editing: The Atlas system, while it does possess some rudimentary English-language word-processing functions, generally lacks the

A. "Layered" characters not easily delineated

トップパブリッシングの

B. Characters not clearly separated

団体定期保険制

Figure 3. Example of Character Types Frequently Misread by the Atlas OCR

U.S. English-language word-processing programs. It has thus been found that using a Wang or IBM System for post-editing is far more efficient than using the Atlas.

Output Methods : The ability to furnish to post-editors and consumers hardcopy printouts containing the pre-edited Japanese text, the English translation, and English translations of individual words in sentences the Atlas was unable to fully translate represents a strength of the Atlas System. However, since files for transfer to English-language word-processing systems via floppy disk can contain only the English translation, the use of floppy disk output for anything but polished translations is prohibited.

IV. Conclusions and Recommendations

The Atlas G-160 system represents state-of-the-art machine translation technology, yet at the same time displays some of the fundamental technical limitations seemingly generic to all such systems today.

Simply stated, while the machine has been well engineered and slickly packaged, it is not able—nor does it claim to be able—to produce consistently coherent and accurate translations across broad or varied subject fields. Current algorithms and associated natural language processing techniques simply are not that far advanced. And while the use of extensive, time-consuming pre- and post-editing procedures can somewhat remedy this basic shortcoming, such procedures are almost always extremely inefficient. For these reasons, the Atlas MT system does not represent the "ultimate" answer to all [redacted] translation needs. Rather, it is a tool which—with the proper investment to time and effort—can yield definite benefits in improved productivity in specific documents fields.

For example, a 20-page typewritten document dealing with telecommunications and written by specialists in a very "plain" style can be input via the OCR, proofread, translated in raw form, and output as an Atlas printout in anywhere between 4 hours (very few OCR mistakes) and 10 hours (many OCR errors). This raw translation can be given directly to a consumer for determining whether further translation is necessary. If no further action is required by a translator, hundreds of dollars have been saved by not having to work for two to three weeks to produce a polished translation that is not needed by the consumer. If a polished translation IS required, substantial amounts of time and money are still saved. [redacted] operators have found that on average, and Atlas raw translation of the same type of document followed by off-system post-editing by a translator results in producing 10 pages in the time it takes to translate seven pages by conventional methods only.

On the other hand, [redacted] operators have found that a polished translation of even five or six pages of a handwritten document not related to an S&T field and written in a too-elevated, too colloquial, or too illogical style invariably takes anywhere between four and five times longer on the Atlas system than by conventional translation methods. Use of the Atlas for translating this type of document is simply not worth the trouble.

The important thing, therefore, is to use the MT system where it is most effective, where it can assist by speeding up the translation process instead of slowing down that process. For [redacted] that means using the Atlas for documents which are concise and well written from the standpoint of style and vocabulary, which is most often the

~~FOR OFFICIAL USE ONLY~~

P.L. 86-36

case for documents in S&T-related fields. Moreover, it means concentrating—for the most part—on producing a high volume of raw translations for subsequent screening by consumers. Encouraging the use of input via floppy disk—[redacted] draft information reports, for example—would also contribute to increased productivity by eliminating the need to manually check for OCR errors. In sum, if used in a way so as to exploit its strengths instead of its weaknesses, the Atlas MT system can be a valuable tool in increasing translation productivity; yet, it is no "dream machine" and should never be purchased or used as an excuse for not hiring, training, and retaining the very best translator work force possible.

P.L. 86-36

V. System Components

Hardware

- Fujitsu G160 oc with a single 5.25" floppy disk drive
- Expansion memory to 8 megabytes
- 135-megabyte hard disk drive
- JIS keyboard
- Color display
- Mouse
- Kanji printer
- Optical character reader
- OCR connection unit

Software

- Atlas-G set
- 30-dot character group
- MS-DOS data connector
- Image processing option
- OCR control option
- Dictionary (biology & medicine)
- Dictionary (industrial chemistry)
- Dictionary (meteorology, seismology, astronomy)
- Dictionary (mechanical engineering)
- Dictionary (civil engineering & construction)
- Dictionary (physics & atomic energy)
- Dictionary (transportation)
- Dictionary (electricity & electronics)
- Dictionary (mathematics & information)
- Dictionary (plants & factories)
- Dictionary (automobiles)
- Dictionary (biochemistry)
- Dictionary (information processing)

Cost

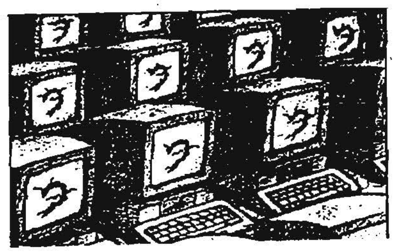
Cost of Hardware:	¥5,974,700
Cost of Software:	¥3,614,950
Total Cost:	¥9,589,650

We welcome reviews and reports of hardware, software, training materials, books, technical literature, and conferences.

~~FOR OFFICIAL USE ONLY~~

NSA and Computer Viruses

Richard Sylvester, B Group



(U) There has been considerable discussion in the Agency about computer viruses and the possibility that NSA's computers and computer networks have been invaded. In discussing the viruses, writers must be aware of the classifications that apply to the discussions.

(U) As most of us know, a computer virus is a software program designed specifically to reproduce itself and to modify or destroy computer software, damage or destroy equipment, or compromise sensitive data. Some of us, however, are unaware that **any** personal computer network or office automation system is susceptible to invasions by a virus, infecting any host in which the program is used. Owing to the insidious nature of a virus, any unwitting user can become an unwitting propagator.

(U) We have established classification guidelines to discuss viruses that may be summarized as follows:

- (U) the fact that NSA is aware of computer viruses and that we take steps to minimize the risk of introducing viruses into our automated information systems (AIS) or networks is UNCLASSIFIED.

- (U) the fact that we employ commercially produced software to scan for virus infections also is UNCLASSIFIED.

- ~~(FOUO)~~ an admission of vulnerability, i.e., the acknowledgment that NSA has experienced the intrusion of a computer virus in any of its systems is classified, at a minimum, CONFIDENTIAL.

- ~~(FOUO)~~ The disclosure of the extent of infection or the name of the specific virus that may have been discovered in an NSA AIS or network is classified, at a minimum, SECRET.

- ~~(FOUO)~~ Specifics concerning an infection, such as the severity of the infection, the extent of damage done, the complexity and expense of eradicating the disease, or the impact of the virus on operations is classified TOP SECRET; in some instances, if certain details concerning the AIS, the network, or the database in which the virus was discovered are revealed, any of the above revelations may require handling in COMINT channels (HVCCO) or even in codeword channels.

~~(FOUO)~~ In summary, be careful what you say and to whom you say about NSA and computer viruses. For further information concerning the classification matters pertaining to computer viruses, contact your classification advisory officer (CAO), your local Computer Security Officer or Computer Security Manager, or J06, the TCOM office of Operational Computer Security.

From the Past

The Department of Defense is commemorating the 50th Anniversary of World War II in 1991-1995. Unlike the immediate post-WW I period when cryptology went underground, the organizations involved during World War II remained on the post-war scene. Even in the darkest hours of the war it had become evident that one of our one-time allies was already an adversary. To a great extent this cryptologic agency owes its continuance to the Cold War that had its origins in the bitter war years.

EO 1.4.(c)
P.L. 86-36

Letter

To the Editor:

When General Matthew Ridgeway retired as Army Chief of Staff—this was the man who had entered Nazi Germany as commander of the First Airborne Division, who relieved General Douglas MacArthur as Supreme Allied Commander in Korea, who stopped the Chinese advance while in command of the Eighth Army—he was asked what he considered to be most important accomplishment of his career. His response was simple: "I protected the mavericks." General Ridgeway's answer may seem surprising, but it reflects a profound understanding of both the realities of organizations and the requirements for their success.

In theory, "protecting the mavericks" is easy: "mavericks" are the people whose new ideas and approaches make their organizations uncomfortable today but that will be invaluable in solving the problems of tomorrow. We all know the TQM "school solution"—new ideas are "good" and people who have them must be encouraged, rewarded and protected. What could be plainer, and why was General Ridgeway proud of such a simple thing?

Of course, the real problem is much harder. Good ideas are like gold: they are rarely unalloyed, they are often found in unattractive surroundings, and they sometimes require lots of work to refine and forge a final product. Mavericks are the prospectors who find them.

But sometimes it's easier for a leader to decide that the sparkle of a new discovery is iron pyrite—"fool's gold"—and tell the prospector to leave it in the ground. Unfortunately, when that happens too often, the loss can extend far beyond one idea. Like prospectors whose claims never pan out, some mavericks just give up and stop having new ideas. Still others keep having them, but for new employers after they resign or are penalized for sins like "unpredictable creativity"—against which at least one military officer has been strongly counseled.

The truth is that mavericks make us feel uncomfortable; they "question authority," they "rock the boat," and they "don't understand how we do business."

But these three phrases are often the best descriptions of really new ideas. For example, Kodak applied them in 1948 when they turned down an inventor named Charles F. Carlson when he proposed a new copying process. Carlson's idea was to use high-voltage electrostatic charge to attach fine black powder to plain paper, then to heat the paper and melt the powder into its fibers. Kodak turned Carlson down flat; the process was too complex, the machines were too expensive, and, anyway, the whole thing was unnecessary—everyone at Kodak knew that if you want a picture on paper, the best way was to start with a picture on Kodak film.

Kodak was wrong. Carlson took the ideas that Kodak turned down and sold them to an unknown company named Haloid Corporation. Today Haloid's name is Xerox.

There are two important lessons here, but they're not simple ones, like "don't turn down another Xerox" or "don't make `bad' decisions." There won't be another Xerox—the next revolution will start with another idea—and at the time the decision to turn it down wasn't a "bad" decision, it was a "good" decision made in the wrong context.

Carlson's process *was* expensive, it *was* complex (the "copier repairman" is still a standard figure in business cartoons), and it *would* have been in direct competition to Kodak's traditional business—and there was a good chance it wouldn't work outside the lab. In the everyday context of a Kodak operating manager, *investing*—not declining—would have been the "bad" decision. After all, the project involved high-cost, substantial risk, and the best foreseeable outcome would be launched another competitor for Kodak's existing positions in the markets for film and photographic supplies. The "real" measure of the right decision—Xerox's subsequent commercial success—wasn't available until years later.

So what are the two lessons? I think they are these:
1. Always look for the *largest reasonable context* in which to evaluate a decision, and,
2. Have the courage to *risk* some "bad" decisions.

In the Kodak-Xerox case, making the decision in the largest context would have begun with recognizing that Kodak's real business was "putting pictures on paper," not "selling film and photographic supplies." Seen this way, Carlson's invention of plain-paper copier that could be used by people with no special training right in their offices fits right in as a logical extension of (and not a competitor to) Kodak's existing business.

Of course, neither of these lessons is as easy to apply without the benefit of hindsight; as the saying goes, "When you're up to your ass in alligators, it's difficult to see that your objective was to drain the swamp." It's even more difficult to see something like "economic enhancement through the provision of retail jobs in a suburban shopping facility in a soon-to-be-drained swamp" as the "largest reasonable context." With alligators alongside, it may seem like the *only* "reasonable" context for any decision is "keeping all my body parts" or just "not becoming someone's lunch." At times like these, we need reminders, like the slaves who accompanied victorious Roman generals in their triumphal parades, reminding them of their mortality. But our reminders should be to "consider the largest context" and "have the courage to take risks."

And that's exactly why we need mavericks. They are the people who ask questions in the "largest context" and suggest the new solutions for difficult problems—questions like "Why *are* we draining the swamp?" and solutions like "First thing, let's tame all the alligators." Their ideas may be annoying, frustrating, embarrassing, and even threatening (perhaps we picked the swamp ourselves!), but they may also lead us out of the muck and up to drier ground. Perhaps Robert Kennedy thought of himself as a maverick when he wrote:

> Some people look at things as they are and ask "Why?"
> I think of things as they could be and ask "Why not?"

Don't think of your mavericks as "trouble-makers" or "boat-rockers," think of them as "principal staff for thinking of things 'as they ought to be' "—and, if they do tame the alligators, how easy it will be to drain the swamp.

Anonymous

Bulletin Board

~~(FOUO)~~ L143 has a degausser for large objects like disk drives. It can handle objects 8" x 18" x 22". The degausser is located in SAB-3. It has been approved by T03, now J06.

~~(FOUO)~~ In the past we discarded defective disk drives. This will allow us recoup our investment. For example, SUN disk drives cost about $16,000 each.

~~(FOUO)~~ To degauss your drive, write a memo to J06, with specifics about the drive.

Editorial

ABOUT THE DISTRIBUTION OF CRYPTOLOG

We regret to inform subscribers that we lost our ability to distribute individual copies beginning with 3rd Issue 1991. We're looking for an alternative. For the time being we ask you to be patient. And as there are many ongoing changes in organizational designators and people are moving about, it will take quite a while until everything is on even keel again. So please be patient. Meanwhile, if you have missed copies, you can ask for them, using the form shown below, or write a note to CRYPTOLOG Distribution, P0541, Ops-1. **As there are over 3,000 subscribers, it stands to reason that you should not call about your subscription or distribution.**

For your convenience, the contents of the issues published beginning with 3rd Issue 1991 are shown below. **NO TELEPHONE REQUESTS FOR BACK COPIES WILL BE HONORED.**

3rd Issue 1991

2nd Issue 1992

EO 1.4.(c)
P.L. 86-36

P.L. 86-36

1st Issue 1992

3rd Issue 1992

P.L. 86-36

EO 1.4.(c)
EO 1.4.(d)
P.L. 86-36

PLEASE SEND ME THE ISSUES MARKED BELOW

3RD 1991———	NAME——————————
1ST 1992———	ORG——————————
2ND 1992———	BLDG——————————
3RD1992———	

MAIL TO: CRYPTOLOG DISTRIBUTION, PO541, Ops-1

~~FOR OFFICIAL USE ONLY~~

CRYPTOLOG

Editorial Policy

CRYPTOLOG is a forum for the informal exchange of information by the analytic workforce. Criteria for publication are: that in the opinion of the reviewers, readers will find the article useful or interesting; that the facts are accurate; that the terminology is correct and appropriate to the discipline. Articles may be classified up to and including TSC.

Technical articles are preferred over non-technical; classified over unclassified; shorter articles over longer. Comments and letters are solicited. We invite readers to contribute conference reports and reviews of books, articles, software and hardware that pertain to our mission or to any of our disciplines. Humor is welcome, too.

Please note that while submissions may be published anonymously, the identity of the author must be made known to the Editor. Unsigned letters and articles are discarded.

N.B. We regret that we must now do cost accounting. For every original submission—not written for another purpose— please indicate how much time you spent writing your article or letter, and what your hourly rate is.

If you are a new author, please request "Guidelines for CRYPTOLOG Authors."

How to Submit your Article

Back in the days when CRYPTOLOG was prepared on the then state-of-the-art, a Selectric typewriter, an article might be dashed off on the back of a used lunch bag. But now we're into automation. We appreciate it when authors are, too.

> *N.B.* If the following instructions are a mystery to you, please call upon your local ADP support for enlightenment. As each organization has its own policies and as there's a myriad of terminals out there, CRYPTOLOG regrets that it cannot advise you.

Send two legible hard copies accompanied by a floppy, disk, or cartridge as described below, or use electronic mail. In your electronic medium (floppy, disk, cartridge, or electronic mail) please heed these strictures to avoid extra data prep that will delay publication:

- do not type your article in capital letters
- do not right-justify
- do not double space between lines
- but **do** double space between paragraphs
- do not indent for a new paragraph
- but **do** paragraph classify
- do not format an HD floppy as DD or vice-versa–our equipment can't cope
- label your floppy or cartridge: identify hardware, density of medium, software;
- put your name, organization, building and phone number on the floppy or cartridge

The electronic mail address is [redacted] (pass to CRYPTOLOG) P.L. 86-36
or [redacted] (pass to CRYPTOLOG)

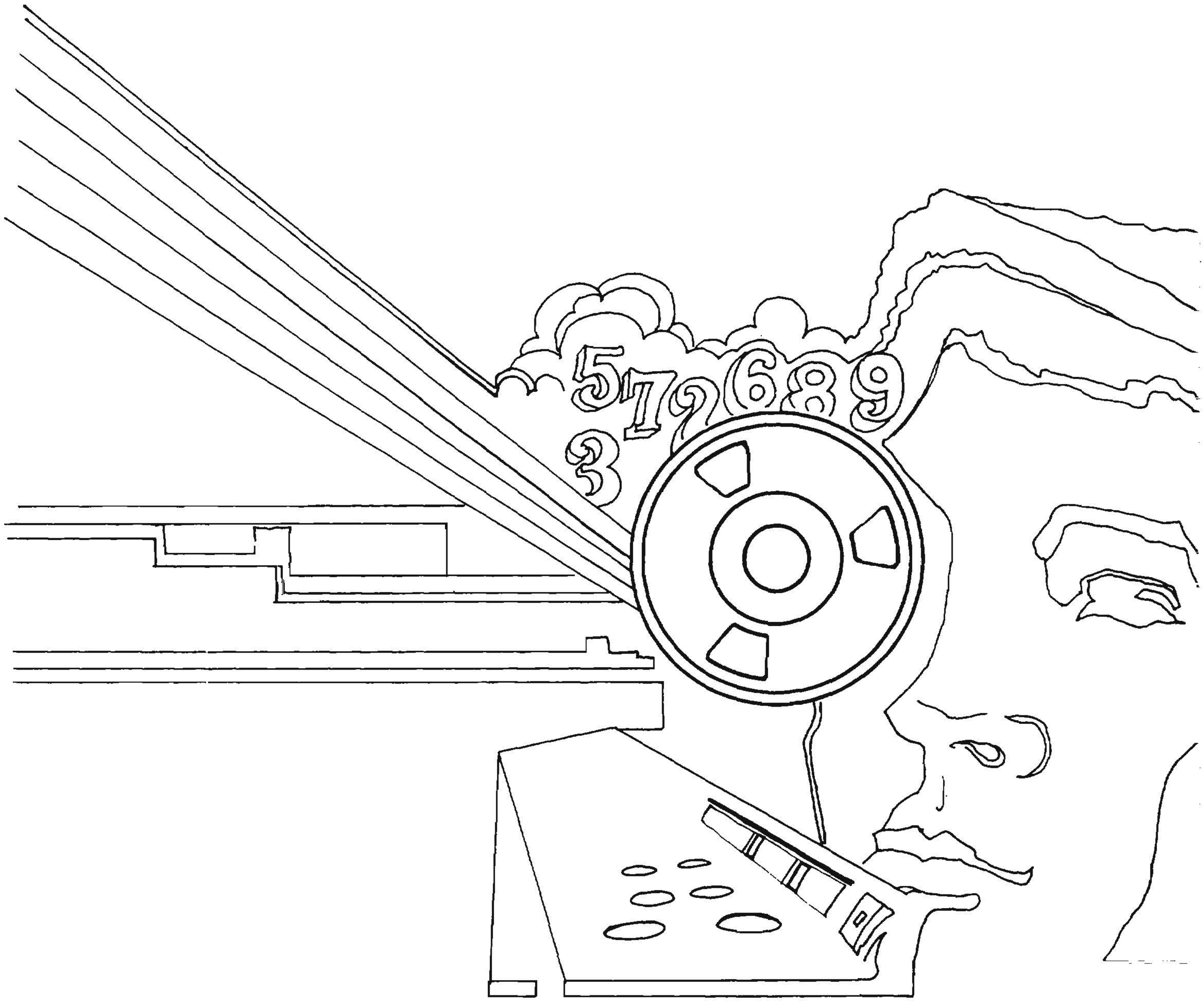
CRYPTOLOG publishes using Macintosh and Xerox Star. It can read output from the equipment shown below. If you have something else, check with the Editor, as new conversions are being added.

SUN	60 or 150 MB cartridge	ascii only
XEROX VP 2.0, 2.1	5 1/4" floppy only	
WANG		Stand-alone or Alliance
Macintosh	3 1/2" DD disk only	Please furnish a copy in TEXT as well as in your software, as we may not have all the software upgrades
IBM & Compatibles	3 1/2" DD or HD 5 1/2" DD or HD	Please furnish a copy in ascii as well as in your software, as we may not have all the software upgrades

~~FOR OFFICIAL USE ONLY~~

DOCID: 4036135

TOP SECRET



THIS DOCUMENT CONTAINS CODEWORD MATERIAL

TOP SECRET

NOT RELEASABLE TO CONTRACTORS