DOCID: 4009825

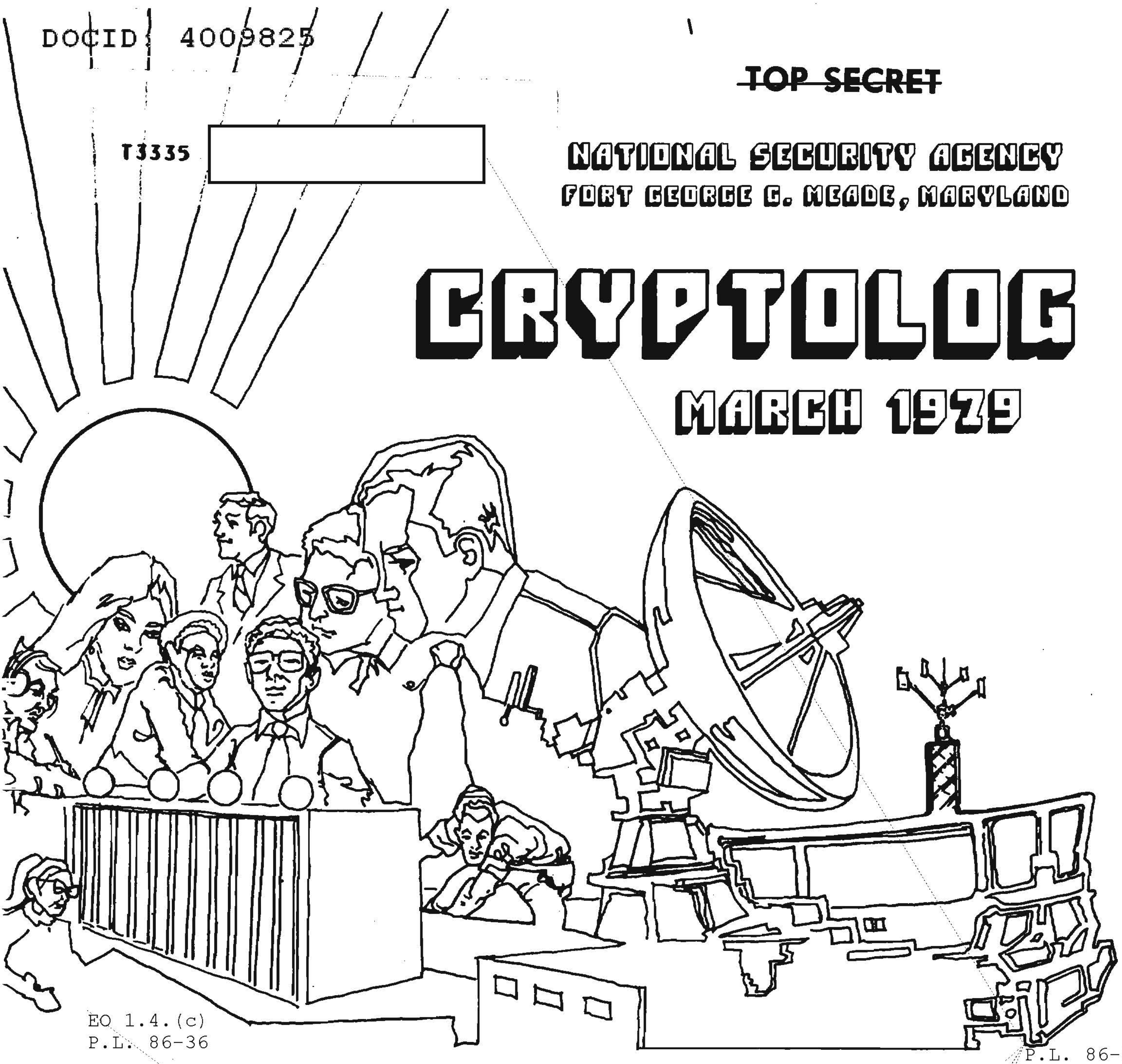
~~TOP SECRET~~

T3335 [redacted]

NATIONAL SECURITY AGENCY
FORT GEORGE G. MEADE, MARYLAND

CRYPTOLOG

MARCH 1979



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

P.L. 86-36

~~THIS DOCUMENT CONTAINS CODEWORD MATERIAL~~

~~TOP SECRET~~

~~CLASSIFIED BY NSA/CSSM 123-2~~
~~REVIEW ON 2 MAR 2009~~

~~TOP SECRET~~

CRYPTOLOG

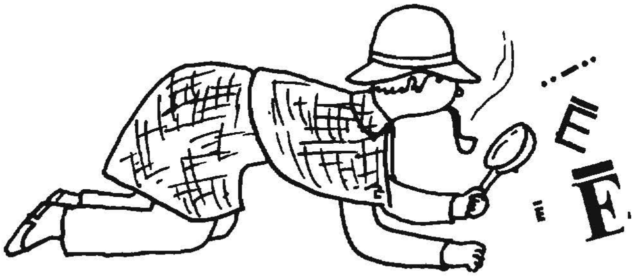
Published Monthly by P1, Techniques and Standards,
for the Personnel of Operations

VOL. VI, No. 3 MARCH 1979

PUBLISHER WILLIAM LUTWINIAK

For individual subscriptions
send
name and organizational designator
to: CRYPTOLOG, P1

~~TOP SECRET~~



Pursuit of the [redacted]

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

Ē. LEIGH SAWYER, B4

[redacted]

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

[redacted]

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

[redacted]

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

[redacted]

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

[redacted]

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

[redacted]

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

SOMETHING TO
BEAR IN MIND

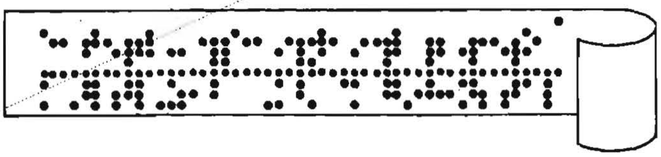
CISI SPRING CONFERENCE

The Computer and Information Sciences Institute (CISI) will hold its 1979 Spring Conference during the week of 21—25 May in the Friedman Auditorium.

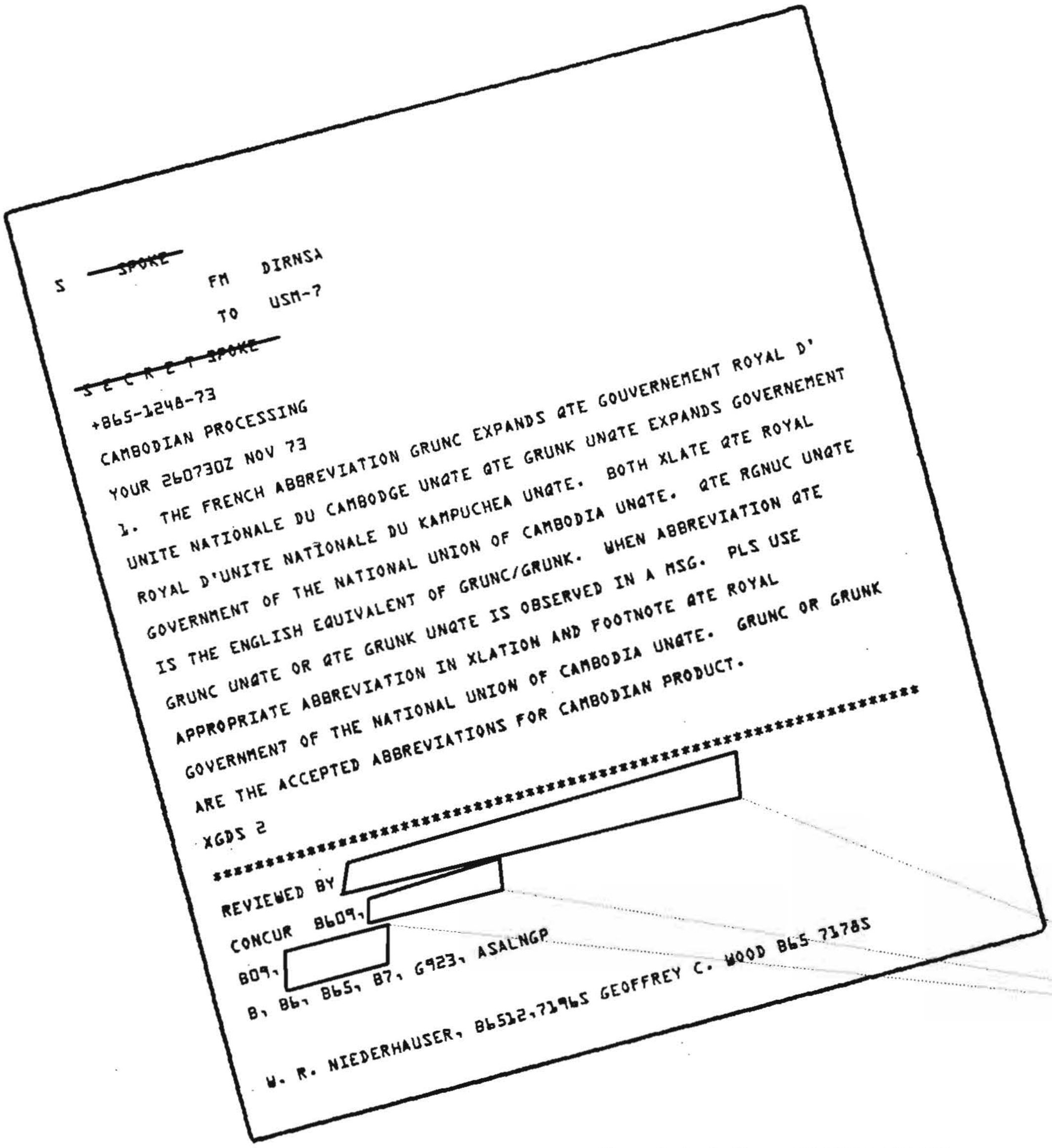
The theme of the conference will be "The User...It's About Time." Subjects of the presentations and times of the sessions will be announced later. (U)

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

CLASSIC CABLES



[redacted]



S ~~SPOKE~~ FM DIRNSA
TO USM-7

~~S E C R E T SPOKE~~
+B65-1248-73
CAMBODIAN PROCESSING
YOUR 260730Z NOV 73
1. THE FRENCH ABBREVIATION GRUNC EXPANDS QTE GOUVERNEMENT ROYAL D'
UNITE NATIONALE DU CAMBODGE UNQTE QTE GRUNK UNQTE EXPANDS GOVERNEMENT
ROYAL D'UNITE NATIONALE DU KAMPUCHEA UNQTE. BOTH XLATE QTE ROYAL
GOVERNMENT OF THE NATIONAL UNION OF CAMBODIA UNQTE. QTE RGNUC UNQTE
IS THE ENGLISH EQUIVALENT OF GRUNC/GRUNK. WHEN ABBREVIATION QTE
GRUNC UNQTE OR QTE GRUNK UNQTE IS OBSERVED IN A MSG. PLS USE
APPROPRIATE ABBREVIATION IN XLATION AND FOOTNOTE QTE ROYAL
GOVERNMENT OF THE NATIONAL UNION OF CAMBODIA UNQTE. GRUNC OR GRUNK
ARE THE ACCEPTED ABBREVIATIONS FOR CAMBODIAN PRODUCT.
XGDS 2
****
REVIEWED BY [redacted]
CONCUR B609, [redacted]
B09, [redacted]
B, B6, B65, B7, G923, ASALNGP
W. R. NIEDERHAUSER, B6512,7196S GEOFFREY C. WOOD B65 7178S

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

(Reprinted from DRAGON SEEDS, December 1973) ~~(SC)~~

VORD *is* A Better Idea

[redacted] P1

P.L. 86-36

For the past two decades NSA has been using a language aptitude test which is both weak and outmoded. This article summarizes the work performed by James Child, and others, both here and in other agencies, to develop a more reliable aptitude test.

In developing a new test for language aptitude I assumed the existence of an unqualified aptitude for learning foreign languages, although it might be argued that this skill is subsumed by general verbal aptitude and need not be tested by an artificial language. I have also not dealt with the possibility that there may be *two* kinds of aptitude, one for participating in face-to-face exchanges using foreign languages and the other for analyzing linguistic content.

The problems of the new test are in the main syntactic and require a skill in absorbing grammar forms that reflect quite different kinds of relationships within the sentence from those most students are accustomed to in English, Spanish and other European languages. The lexicon I have developed has played a minor role so far, but if the exercise were "powered" (i.e., required to be taken under pressure of time), it could be used to test vocabulary memory as well.

Before taking up the test proper, I would like to express my appreciation to my colleagues at NSA and other agencies for their willingness to give time and considerable effort in helping validate the test and in making a great many useful suggestions to improve it. The weaknesses of the new model are my responsibility alone.

This new test, which I have named VORD, has undergone extensive trial and is being unofficially used in the screening of both prospective and present employees. My colleagues and I have developed and refined this test in response to the need for an instrument better predictive of success in non-Indo-European languages than the Army Language Aptitude Test (ALAT) currently in use at NSA and long used at the Defense Language Institute (under the designation DLAT and with some differences in format and norming).

ALAT, designed and validated in the late 1950s, has served both agencies reasonably well in foretelling student success particularly in the learning of West European languages, although even here the very high and very low scores correlate much better with proficiency test results than do those scores in the middle. It has not been of much predictive value, however, in the learning of languages like Korean and Vietnamese. Indeed a careful study carried out in late 1971 or 1972 by COL Kibbey Horne, linguist and one-time commandant of the DLI school at Monterey, California, showed an almost random correlation between aptitude scores and course grades for these two languages, and our experience at NSA supports his findings.

Hence the impetus for the new test. In the paragraphs below I will first touch on the linguistic features of ALAT, then discuss at greater length the philosophy behind our test design, the various forms the test has taken over four years and the results we have obtained so far. In comparing the two tests, I will try to show that the key to language aptitude as understood in this paper is the degree of skill required in mastering a language system vastly different from one's own, as opposed to mastering a language with a similar system.

ALAT is a 57-item test based on an artificial language which has been variously described as formally similar to Turkish (e.g., by Kibbey Horne) and as western Indo-European in typology. Actually, the test so closely resembles English syntactically that a test subject who can quickly memorize the few grammar rules and somewhat more numerous words and grammatical forms can make a relatively high score. This is not to say, though, that ALAT is a memory test as such; the examinees can refer to their rules and lists as often as they like. However, the time limitations (7 minutes to study grammar and vocabulary, 20 minutes to do the problems) are what drive the test. The few problems of linguistic interest come toward the end, between questions 50 and 57, but as almost no one gets that far the issue is academic. In short, the test stresses quick look-up and "photographic memory."[1]

Obviously, if some ability to perform linguistic analysis is an ingredient of language aptitude (at least the kind of aptitude NSA requires) a new model was badly needed. This I launched in September 1973 in the form of a test based on an artificial language structurally like Turkish. Since Turkic languages are very different in structure from most European languages, and not many job applicants are likely to have studied them in depth, this typology seemed a good choice. The test itself in its original form contained 32 questions, ten each on nominal and verbal morphology, and 12 on phrase and sentence level syntax. The questions were designed to be progressively more difficult, the latter 12 requiring the subject to supply a fair amount of language forms to establish sentence patterns (unlike the ALAT which has always been multiple-choice in format and hence machine-gradable).

The first ten questions called for simple suffixes to be added to nouns, while the next ten required the subject to select correct verbal forms from multiple-choice listings. (Unfortunately, since the test will soon be official, I cannot cite examples from it.) As analytic skill rather than memory was the chief object, I decided not to set a time limit until we had at least a few cases for which the running time was recorded against which I could make a rough projection.

At this point my long-suffering colleagues came into the picture. We all thought that it would be most useful to try VORD and ALAT on the subjects the Educational Testing Service (ETS) found for our 1973-1974 CLOZE[2] test trials in German, Portuguese, and Russian (about 100 subjects in each language). Thus, since time was short, I hurriedly completed an inhouse trial to see if it was workable at all.

The limited number of cases—less than ten—suggested that the test did work, although the scores were on the high side because of the linguistic sophistication of the subjects. Their running time averaged about 45 minutes, so we allocated one hour for the ETS experiment. We then printed the test and turned it over to the contractor.

The results of the field testing were encouraging, though, to be sure, we were not using the model in a purely predictive situation; most of our 300 subjects had studied at least one natural language in high school or college. The correlations of VORD with CLOZE results in German and Russian were about as good, though certainly no better, than those of ALAT with CLOZE; the Portuguese results were more encouraging:

	VORD vs CLOZE	ALAT vs CLOZE
German	.35	.36
Russian	.29	.33
Portuguese	.52	.35

In all three test comparisons, however, we noted that the relatively small number of questions on VORD (32) together with the ample time alloted to it led to a bunching of scores at the high end of the scale.

The options for strengthening the test would have been to add another more difficult section which would have had the added advantage of face comparability with ALAT, or to make it a power test which, as I pointed out above, I did not want to do because we could not truly test analytic ability.

As I was discussing the questions with my colleagues, I was overtaken by events, in the form of an opportunity to test VORD and ALAT (again through ETS) on 150 subjects who were also to be tested in Arabic, Chinese and Japanese (about 50 of each). This time the correlation between VORD and the respective proficiency tests was stronger than between ALAT and the CLOZE forms:

	VORD vs CLOZE	ALAT vs CLOZE
Arabic	.53	.48
Japanese	.22	.06
Chinese	.52	.07

We found the figures for the Chinese testing particularly gratifying.

[1] The inadequacy of ALAT prompted DLI to develop a new and much longer aptitude battery (Defense Language Aptitude Battery), which has proved to be a better predictor of success in learning some languages than DLAT. However, because it requires considerably more time and special equipment than either ALAT or the new NSA test, and because it is still unproved for many languages, I have not treated it in this paper. For full information see "The Development of the Defense Language Aptitude Battery (DLAB)," by Calvin R. Petersen and Antoine R. Al-Haik, *Educational and Psychological Measurement*, 1976, Vol. XXXVI. No. 2, pp. 369-380.

[2] The CLOZE language testing technique, which has been in use for several decades, involves deleting letters, syllables, words, or any other linguistic unit at some arbitrarily chosen interval (say, every fifth position), and requiring the test subjects to restore the missing material. Our use of the term is somewhat inaccurate since we establish our deletions at points of particular linguistic interest rather than doing so mechanically.

Exciting as these results were, the raw scores supporting them were still insufficiently spread out to give us plausible ranges within a STANINE or STATEN structure (the same problem is also encountered in ALAT, with its apparent 57 problems which for the vast majority of subjects amounts to about 45).

We therefore decided in July 1976 that we should add more questions to bring the test into a range of 50 to 60 problems. At this point I devised a CLOZE test along the lines of the models we use for proficiency testing (running VORD test, with deletions, on the right hand side, and English facing translation on the left). Twenty-eight items were deleted and, in the interest of maintaining a totally machine-gradable test, five multiple-choice alternatives were listed below each (lined) blank.

Once again we chose subjects for feasibility testing, but this time we had the leisure to be selective: four of our guinea pigs were multilinguists with at least some training in formal linguistics; eight worked with Romance languages; and seven were Turkish linguists. The not too surprising result was that the first and third groups scored very high, while the second group, with two brilliant exceptions, brought up the rear. The raw scores ranged from 26 out of 28 down to 12 out of 28 correct, distributed in a reasonable bell curve. Three items did not work well so we restored them, leaving us with, concidentally, a 57-item test.

Since 1977 we have been trying this test out on outside applicants for language jobs and comparing the results with the scores made by these people on language proficiency tests. The some general relationships appear for ALAT/VORD and Russian CLOZE tests as obtained in the ETS experiment, if we consider only the 32-question part of VORD (Part 1): .31 and .26, respectively, for a population of about 100. When Part 2 is correlated with the Russian CLOZE the result is a nonsignificant .06.

In languages other than Russian, the figures are too scant to permit the drawing of any firm conclusions. We have done some inhouse testing for persons scheduled to take Chinese, Korean and Arabic (about 56 all told), but the respective courses are not far enough along to permit serious proficiency testing and data comparison. The most we can say at this stage is that the subjects screened had either scored high on ALAT/DLAT as well as VORD or were linguists with considerable experience in several other languages who did well on VORD.

We plan to continue administering VORD to prospective students of these languages. We believe that when enough cases have been collected the new test will prove to be a much stronger predictor than ALAT. (U)

ATTENTION: MILITARY TRAFFIC ANALYSTS!

Are you a professional Traffic Analyst? Why not fill out a Professional Qualification Record (Form P7940), and submit it to the TA Panel, H115, for evaluation against the published criterial. Let's find out how close you are to professionalization.

Are you required to do this? The answer is NO. There is no current requirement for members of the SCAs to take this action. So why should you? What's in it for you? The best answer is self-satisfaction and pride. As a military man or woman assigned here, you are a member of the NSA cryptologic team. Civilian members of the team, many of them former military, are aspirants for professionalization and know where they stand in seeking certification.

A few SCA members have applied for certification and have received personal notification pf their status. Specifically, there are 30 military aspirants for TA certification at this time; this is about 16% of the total assigned military TA population participating in the professionalization program. Only five military personnel, currently on the rolls, have achieved certification in the TA field. Of note is the fact that the highest score attained on the most recent Related Fields Examination, a basic requirement for certification in both the TA and SR (Special Research) career fields, was achieved by an SCA member (Navy).

You are invited to participate. Fill out a PQR and submit it to H115, Room 1W155. If you have any questions, do not hesitate to call us, ext 3573s. (U)

SOLUTION TO NSA-CROSTIC NO. 22
(*CRYPTOLOG*, February 1979), by DHW

B[ill] Crowell, "[A] Computer Scratch Pad
[at Home or at Work?], *CRYPTOLOG*, June 1978

"Almost unnoticed at NSA, the outside world has undergone a revolution in their approach to computer support. The day of the microcomputer has arrived. Not only have thousands of very small businesses begun using them, but...even individuals are buying them and ...creating new applications on them." (U)

BELP WANTED!

The four roads around the Agency's main buildings are named Towler, Engstrom, Herzog and Wray. Before everyone who knew the men for whom these roads were named has either retired or died, it might be appropriate to write short articles about the contributions and personalities of each of them. If you have any information of this kind, please send your recollections to: [redacted] *P12, who hopes to coordinate the project.* (U)

P.L. 86-36

READERS' SURVEY By DHW

In last December's issue, CRYPTOLOG printed a survey questionnaire, asking for reader comments about the magazine. We'll be publishing the results of the poll in an early issue. Two of the responses, however, merit publication on their own.

One of the questions asked for opinions on why there had been so few women contributors to CRYPTOLOG in 1978; out of a total of 83, only 11 were women.

After weeding out the extremist pro- and anti-feminist sentiments, I was left with three responses, all of which said more or less the same thing. It was best expressed by one young lady who wrote:

> "I believe this relates to the percentage of women at NSA, particularly in the higher grades. Your 14 per cent female participation for the year is not bad considering that only 9.5 per cent of the workforce at or above grade 11 is female—and I would imagine that the majority of your articles are written by persons at those grade levels."

I have no doubt that this is true. CRYPTOLOG articles tend to be written by managers, analysts, computer specialists, engineers, and others in comparable jobs, who, as noted, tend to be people in the higher grades.

A quick review of the back numbers of the magazine shows (although I'm willing to be corrected by someone with a better memory than mine) that we have never carried a piece by any member of the secretarial or clerical force. Why should this be?

There must be quite a few people in those categories who have things to say which would be of considerable interest to CRYPTOLOG's readership. Personally, I can think of more than one young lady around here who could write nonstop for several hours on "A Secretary's Lot is Not a Happy One (If you Work For a Clown Like I Do)" or "Prematurely Gray at Age 26."

I'm sure there are also serious topics worth taking up, such as "Six Shortcuts to Office Efficiency" or "Why Doesn't Somebody Invent a _______!" So, come on there, ladies; let's hear from you. Put something of your own creation through your typewriter; that dreary memo can wait. I'll be looking for it.

The following response is printed in full, and with it goes an invitation to the right person (in N? in D?) to respond to it.

"How's chances of an article on the budget process? In these days of fiscal austerity, where just about any cryptologist seeking to do his or her job better (or, sometimes, just to maintain the status quo) is faced with a myriad of problems in competing for extremely hard-to-obtain funding, one is puzzled, or even baffled, by the process itself and, in particular, by the associated terminology.

'For example,

What does "over guidance" or "below the line" mean?

-Who makes up the CRG, or the RRG, and what specific roles do they play?

-Once the NSA budget is prepared and "blessed" by ADPR and the Director, to whom is it submitted, and what happens next?

-Who else reviews, comments, cuts, rearranges, etc., our budget proposal?

-What are Congressional Review Books (Congressional Justification? Books), and what role do they play in the budget review process?

'Those questions are intended to be illustrative, not all-encompassing. Certainly, there are lots more buzzwords or steps in the process that I have missed. How about a series of articles, like the old Saturday afternoon matinee serials, designed to keep us sitting on the edges of our chairs until the next issue.

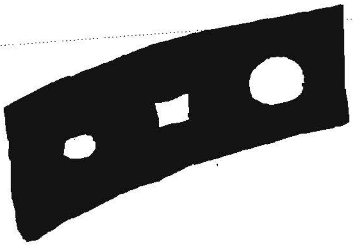
"I realize this request is a tall order, but please consider that there are a lot or us out here who contribute to some or all of the information-gathering activities which support many of the budget review procedures, yet we do not have a full understanding of what is going on, or why it is necessary to recast information in different formats over and over again.

'Perhaps the explanation I have requested above would be useful to educate us. Armed with this knowledge, we may be able to be more responsive to the various requests, and, who knows, we may end up with a better product, or even more money.

"And, by the way, please keep the article simple to understand." (U)

[redacted]

P.L. 86-36



Let's Not Lose Our TA Skills

One thing a middle-level supervisor in the Production organization realizes very quickly is that good traffic analysts are hard to find. Those traffic analysts with a skill in a specialized area such as frequency and callsign recovery are scarce. [redacted] As indicated in the A/DDO memorandum, the underlying causes for this decrease in traffic analysts are the rapid change to automated methods of collecting and producing SIGINT, and the personnel limits imposed on the size of the NSA work force. Since NSA cannot hire personnel to fill shortages in critical skills, the traditional skills have been reduced to accommodate increases in linguists, signals conversion personnel, collection technicians, and data systems analysts and programmers. As a result, we are creating a static pool of traffic analysts, retarding the development of our analytic talent and altering the career-progression patterns of the traffic analytic work force. It is these effects that I wish to discuss.

The end of the Vietnam War, the subsequent tightening of purse strings and the resultant reductions in traffic analytic spaces altered the availability of traffic analysts. By limiting the hiring of new traffic analysts and not replacing those lost by attrition, the size of the analytic career field was set [redacted] The immediate effects were minimal since the number of traffic analytic jobs was also decreasing, with the reduction of many of the timely requirements for information on Southeast Asia. Also helping to offset any immediate effects were the great strides made in mechanizing the traffic analytic processes during the Vietnam War. Efficiencies had been created and a degree of timeliness using methods of intelligence production never before possible had become routine.

The long-range effects probably will not be apparent until the late 1980s, but some symptoms are already beginning to appear. Our traffic analytic work force is getting old. Most of the younger analysts were hired during the 1960s and are now GG-11s or higher. Most basic traffic analytic work is now done by the military, either at the field sites or at NSA. No substantial group of young analysts at the lower grades is available for the future. The more aggressive analysts have already moved into management positions to further their careers. To aggravate what is rapidly becoming a bad situation we have retarded the development of the younger traffic analyst. In the earlier growth days of our Agency, a traffic analyst could grow in a specific target area, become recognized as an expert, and advance in grade and responsibility within his chosen career field. Today, the aggressive young analyst soon recognizes that his future is not in the technical side of the traffic analytic business. To advance and achieve a modicum of success, he must move into management or to one of the critical-shortage skills. As a consequence we deplete our analytic talent base and few people are left to form a nucleus for the future.

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

Those who are left usually have a sincere desire to remain in the technical side of the intelligence production business. Even those people are prodded by management to move into the more critical areas of data systems or linguistics. Since chances of promotion are mathematically better in these skills, many of the remaining talented young people do indeed transfer.

Those who remain face a slower career progression since the money provided for the special considerations given to the critical career areas reduces the total sum that would normally be equally divided among all those eligible for promotion. This means the traffic analyst must face stiffer competition for the promotions that are available and ultimately his chances to achieve a position of leadership within the Agency are diminished.

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

As a result, probably in the near future, we are going to be faced with a severe analytic shortage similar to that which we now have with linguists. A more serious consequence will be the loss of analytic skills that can be learned only by years of experience. Specialists will be nonexistent and major analytic recoveries will suffer. Although these problems can be alleviated to a degree by hiring from the SCAs and by programs such as the intern program, these are not immediate solutions. Unlike the data systems and, to a certain extent, the linguistic fields, our colleges and universities are not graduating many traffic analysts. It is a career field where experience is the best teacher.

To avoid future shortages we should begin hiring some Traffic Analytic Technicians right now. These technicians could be hired out of high school at the GG-2 level and put through a program similar to that used for training linguists. Given the proper incentives, training, and experience, these people would be ready to take over the analytic work load in about 10 years. If we fail to act now, we will have to react later when our chances of success are fewer. Traffic analytic skills helped make our Agency what it is today. Let's keep it that way. ~~(C-CCO)~~

While [redacted] article was being prepared for publication, it was shown to [redacted], Chief, Traffic Analysis, Office of Techniques and Standards and he was asked if he would like to add any comments. He has submitted the following addendum.

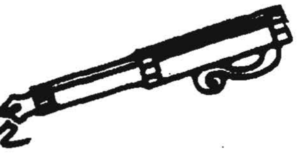
Ed.

P.L. 86-36

We could *also* hire ex-military traffic analysts, as we have in the past. This has the advantage that each "recruit" already knows what TA is, likes doing TA, and wants to make a career of it. That cannot be said of high school hires, and one must therefore expect a higher rate of "drop-out" than would apply to those already trained and experienced in TA (ex-military).

There must, of course, be some disadvantages to hiring ex-military traffic analysts. Otherwise, an agency as smart as we are would already be doing it. (U)

Letter to the Editor



To the Editor, CRYPTOLOG:

In looking at this issue of data standards, I find myself of two minds. I firmly believe in order and organization, but I also know that a "structured orderliness" imposed arbitrarily upon an analytic organization *can* inhibit, and sometimes nullify, analytic initiative. And that initiative, elusive as it is, is the key to whether an analytic effort is alive and responsive, or just plodding and pedestrian.

My roots are in analysis, and I think the greatest challenges I have found have not been solving technical problems but, rather, encouraging others to solve them. That's the essence of being a cryptologic manager. I have come to the conclusion that each analyst has only so much "analytic energy" or attentive capability. The more complex we make *our* system (or that part of it which touches the analyst), the more we force the analyst to spend on *us* — and the less he has left over to spend on *them* (his analytic targets, or tasks)

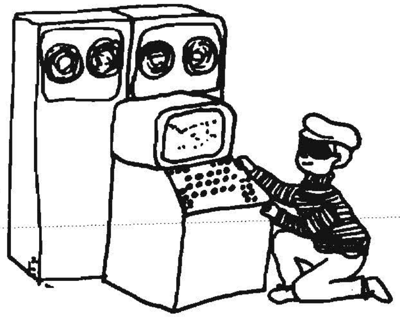
As a line supervisor, I found that in trying to adhere to the many rules, conventions and rituals around here, I could usually "follow the book" so long as it wasn't too costly in "analytic energy." But there was a limit. There was usually a threshold beyond which I would not go, beyond which the bother did not justify the result. My response then (and yours too, I suspect) was to ignore the system or go around it. (We are, after all, a building full of people whose business it is to go around someone else's systems; it isn't that hard to go around ours.)

For the sake of overall order and organization, our rules and rituals ought not to be complex *from the point of view of the one who has to comply.* This holds whether we're talking about forms control, time cards, or data standards. These things are needed, but we have to get our priorities right.

[redacted]
P14 (U)

P.L. 86-36

COMPUTER OPERATING SYSTEM VULNERABILITIES



[redacted] S86

P.L. 86-36

Can my system really be penetrated?" This is the question so often asked by computer system managers. The inevitable answer is "Yes. Any computer system can be penetrated by a knowledgeable user." Large computer systems, in particular, by their size and complexity, leave themselves open to attacks by unauthorized users. Let us examine some of the vulnerabilities of computer systems, as well as some of the possible defensive measures.

COMMON OPERATING SYSTEM VULNERABILITIES

Operating system vulnerabilities generally fall into one or more of the following seven classes: [1,2]

- Incomplete parameter validation
- Inconsistent parameter validation
- Implied sharing of privileged confidential data
- Asynchronous validation and inadequate serialization
- Inadequate identification, authentication or authorization
- Violable limits
- Exploitable logic error

Let us look in detail at each class of flaws and see how they affect the system operation.

Incomplete parameter validation. Whenever a user requests any type of service, the operating system must verify that the user is authorized to make that request and that a proper parameter string has been provided by the user. This verification is done to prevent the user from compromising a control program which is performing services for all users. Flaws in some operating systems may allow a user to "fool" a control program into:

providing him access to data which he would not otherwise be allowed access rights,

placing the user program into privileged or executive mode, or

severely degrading the operation of the ADP system.

The following is a good example of incomplete parameter validation:

Using a file dump routine, User X requests a dump of 300 records from File A, but File A contains only 200 records. The system honors the user request, and User X is allowed access to not only File A, but also to whatever data is stored beyond the address area of File A.

Security requirements should make the control routine validate the parameters and either reject the user request or dump only those records which apply to File A.

Inconsistent parameter validation. Inconsistent parameter validation occurs whenever there are multiple definitions for the same construct within the operating system. For example, a system control program may validate a user program's parameters but trusts another system routine's parameters as valid without verification. Therefore, a user who can fool the system into believing his code is system routine code can obtain unauthorized privileges. System routines should verify all input parameter strings, even those from another system routine.

Implied sharing of privileged or confidential data. In a multiprogramming environment, the computer's facilities are shared by many users. The operating system must have the built-in capability to isolate each user from all other users. Failure to provide this segregation can result in a possible compromise of privileged information. In modern operating systems two problems are generally noted in this area.[3] The first is the matter of sensitive residue. This involves information left behind in memory or other storage media after a run has terminated. An unauthorized user can enter the system and obtain access to these "leftovers." This technique is commonly known as *scavenging*. The second problem involves the system sharing user space for its own storage. To save space, the operating system frequently shares the user's buffers to store temporary working tables. This may allow the user unauthorized access to the system tables, i.e., password tables, etc. This is frequently known as the *unerased blackboard problem.*[1]

Asynchronous validation and inadequate serialization. System integrity is guaranteed only if information passed between program sequences is protected. If the operating system allows asynchronous operations and the operations are not performed in a timely sequence, the information may be modified or compromised. An example of this would be permitting the user to perform I/O into a checkpoint or restart file so that his restarted program is given unauthorized or supervisory privileges.[4] To be secure, an operating system must be able to enforce timing constraints to a controlled state.

Inadequate identification, authorization or authentication. Most operating systems maintain some type of job initiation procedures which monitor authorized vs. unauthorized access. A system flaw exists whenever a system permits a user to bypass these security mechanisms. A user who finds a way to obtain executive operation mode can "walk" through the system without being questioned by the system monitor. Operating systems must require proof of access rights for all user requests. Security mechanisms must be protected from user tampering. For example, password files should be encrypted or protected from common access and must be unusual enough to void any guessing or permutation attempts.

Violable limits. Because of architectural limitations, the operating system has to limit the resources a user can control. These limits or "hands off" policies are usually described in the system documentation. Whenever an advertised limit is not enforced, a security flaw exists. For example, a user may be limited to operate within an assigned partition of storage; but a flaw in the system allows him access to another partition on an overflow condition. Because the operating system did not enforce the "rules of the road," a user could accidentally or deliberately cause a system overload, resulting in system degradation or crash.

Exploitable logic error. With four to five million lines of code, it is inevitable that there will be bugs in any major operating system.[4] A knowledgeable user may exploit these errors to his advantage to obtain access to information or programs to which he is not authorized. Logic errors can especially be created whenever the original design or coding has been changed. Logic modifications compromise any security measures designed into the original system. Examples of exploitable logic errors are frequently found in error-handling procedures. A user may request modifications or dumping of a file belonging to another user. Incorrect error handling may initiate the actions without first verifying that the user has access rights to that file. There is no way to avoid logic errors in large operating systems; however, these errors should be corrected when discovered to avoid prolonged compromise of sensitive information.

PENETRATION TECHNIQUES

Now that we know what some of the potential operating system flaws are, we need to know how a knowledgeable user, or *penetrator*, will exploit these flaws to obtain unauthorized access to the system. In planning his attack, the penetrator will have to answer the question, "What do I want—information or system degradation?"[5] The answer to this question will determine his method of attack. The penetrator's next step is to obtain all available system documentation. Valuable information which may point to vulnerabilities is available in the documentation. After reviewing the manuals, the penetrator can then decide on the techniques to be used in the penetration attempt. The penetrator's main objective is to attack one or more of the seven major flaw classes discussed earlier.

Probably one of the most available and easiest system penetration methods is the use of utility programs.[3] These service routines often execute user requests without requiring proof of access rights. Some types of utility routines are storage dump facilities, operations support programs and maintenance support programs.

Another widely used penetration technique is operator "spoofing." A penetrator can use trickery, such as giving his program the same name as a system routine, to make the operator think that his program is a privileged system routine. He may then request a load of privileged disc packs or magnetic tapes.

The penetrator can also obtain access to privileged information by creating a *Trojan horse.*[3] A Trojan horse is a program which, in addition to doing what it is advertised to do, does something else which its user doesn't know about and wouldn't want done. A Trojan horse is usually hidden in a utility program. An example would be a performance monitor which also dumps user information into a file somewhere (account numbers, passwords, etc.).

System penetration can also be obtained

by using any of several covert attacks.

Wire tapping. Also known as eavesdropping, this act involves the penetrator connecting some listening device to a communications line somewhere between a peripheral device and the computer central processing unit being penetrated. This is a passive operation.

Between lines entry. This is similar to wire tapping except that the process is active. The penetrator enters spurious commands onto the communication lines which were meant only for the legitimate users. This operation is usually done when the intended terminal is at an idle state.

Clandestine code. This operation involves the entering of changes, possibly a Trojan horse, into the coding of the computer operating system.

Masquerading. This involves logging into the computer system as a legitimate user whose account number and password have been acquired by begging, borrowing or stealing.

DEFENSIVE MEASURES (COUNTERMEASURES)

So, if our system is so susceptible to unauthorized access, how can we set up a defense against these measures? The best approach is to build security into the initial system design.[3] Patches to the design at a later time may create more flaws than they patch. The problem with most current operating systems lies in the fact that they were developed in the 1960s with no thought in mind for security requirements. Even with security in mind, we must remember that operating system security is not a binary yes-no condition. No large operating system currently in use can be completely certified as secure.[2]

Here are examples of measures which we can take to protect our system from attack.

Data encryption. Data encryption is becoming more widely used by both the government and private industry. Encryption should be performed whenever sensitive information, such as password files, payroll data, defense statistics, and the like, is stored or sent over data communication lines.

Using a minicomputer as a front-end security controller. This technique could be used to control access to the host computer from remote terminals. This would remove the security overhead from the host computer's operating system. The smaller operating system in the minicomputer would also be easier to certify as secure.

Mathematical models. Models allow systems analysts to study the complete operating system environment and pick each area apart for security analysis.

Kernels. Kernels are small portions of software blocked together to perform a single function. These small software modules could be certified secure.

Software verification tools. Many tools have been or are being developed to certify the security of computer software.

A LOOK AT FUTURE RESEARCH AREAS

Many areas in computer system security need to be explored in the future. Some of those areas are:

1. Development of better control structures (audit trails);[2]

2. Expansion of kernel theory to develop a "secure" operating system;[3]

3. Cost analysis studies (Where do we draw the line between cost of computer security and need? How do we measure security?)[3]

4. Development of strong consistent management policies to govern the use of computer facilities;[4]

5. Development of software verification tools to certify computer software;[3]

6. Development of some type of virtual machine monitor (an operating system which isolates each user into his own mini-operating system), which when properly designed and implemented is "spoof-proof";[3] and

7. Development of a security specification language which allows security requirements to be programmed into the operating system by the security officer.

I hope I have been able to provide some insight into just how vulnerable modern computer operating systems are. Department of Defense studies have shown a need for protecting data relating to the nation's defense because of the many opportunities for fraud and embezzlement.[2] We must also realize that software security is only one aspect of the total security environment. We must also consider administrative, personnel, physical, communications, emanations and hardware security. As modern technological advances are made, with their applications for computers, we will have a continuing requirement for operating system security.

No matter what misuses take place, we must realize that people are still going to use that magnificent adding machine, the computer. It has been proven that there are people with skills to crack safes, yet people still use safes. The same correlation can be made to computer usage. Our job as system managers is to attempt to protect against accidental or deliberate destruction, modification, or disclosure.[2] Security policy (administrative, personnel, physical, communications, emanations, hardware and software) and practices must be sufficient to make up for the computer's inability to protect itself.

---

1. Webb, D.A. and Frickel, W.G., "Handbook for Analyzing the Security of Operating Systems," Lawrence Livermore Laboratories, 1976.

2. Abbott, R.P. et al., "Security and Enhancements of Computer Operating Systems," National Bureau of Standards, Rept. NBSIR 76-1041, April 1976.

3. Hoffman, L.J., *Modern Methods for Computer Security and Privacy*, Prentice-Hall Inc., New Jersey, 1977.

4. Chin, J.S., "Analysis of Operating System Security," Lawrence Livermore Laboratories, December 2, 1975.

5. Linde, R.R., "Operating System Security," *Proceedings of National Computer Conference, 1975*, 1975, pp. 361—368. (U)

"DATA STANDARDS WITHOUT TEARS"

A COMMENT BY [redacted] P1

Much of what [redacted] says in "Data Standards Without Tears"* has merit. The Data Dictionary concept can play a role in the standardization process, but not in the "magical" way he outlines. You can only have standards with *sweat* — without tears, perhaps, but certainly not without considerable labor. I am afraid that we have to indict [redacted] for not really giving due credit to the standardization process that the NDSC has long been pursuing, and also for presenting a few half-truths here and there along with the nuggets of wisdom.

"No one agrees that data standards should be enforced on his project at the expense of operational necessity."

Right. The NDSC has not tried to shut off anyone's job because of failure to observe standards. On paper we have the authority: both NSA Regulation 80-9 and USSID 414, "Standardization of Data Elements and Related Features for SIGINT Activities," Annex B ("Implementation of Standard Data Elements and Related Features in NSA/CSS Computer Projects") give us the *authority* to make life very unhappy for sponsors whose jobs ignore or conflict with published standards. In theory we can point to the concept of enforcement of data standards, even to the short-run disadvantage of a computer project. In actual practice, we sacrifice the long-term benefits to the Agency that would follow from a rigorous enforcement of the standards we already have.

"...we view standards as something which not only can be but must be imposed in an inflexible, hard-handed manner."

The Center never "imposes" standards in this way but issues them only after a long and rigorous process. This begins with a recognized need, research and discussion, drafting of a "proposal," etc., and continues with coordination through the Senior Data Representatives (SDR) of the DDO elements. There are draftings and redraftings to meet objections, suggestions, etc., and final approval comes, in many cases, only after a painfully long process. This is far from an "inflexible, hard-handed manner." A proposed standard always has wide circulation throughout the Agency.

"It goes without saying that [standards] cannot be achieved without some degree of magic. On the practical level the magic machine already exists for rendering coarse materials into fine standard gold..."

I guess a good name for this philosophy of standardization might be the "Rumplestiltskin Syndrome" — after the legendary gnome who was able to weave straw into gold to further his nefarious designs. Let us not accuse our good friends from the DED/D team of such plotting. Everyone would like to have the magic machine dispense usable and workable standards without going through the long and often painful process outlined above.

This philosophy is, I'm afraid, a naive one when viewed in the harsh light of the standardization process. I think I see what [redacted] is saying here, however. He is pointing out:

- the DED/D will expose people to the already-published standard data elements in the dictionary part of the system;
- the DED/D will show people, in the dictionary portion, what the current usage of data fields is along a wide spectrum of different Agency applications. Exposure to this usage will gradually lead us towards the necessary standardization. (The author of the essay does not explicitly state this, but this is my understanding of his concept.)

[redacted] goes on to separate the data features we deal with into two "domains" — Data Elements and Data Fields. I agree that this

---

*CRYPTOLOG, February 1979

P.L. 86-36

is a good approach, both conceptually and physically, within the DED/D. The pure Data Elements go into the dictionary, along with their codes, definitions, configurations, and so forth. The baser Data Fields people use in many of their applications would go into the directory part. In other words, Data Elements point to "things" — classes or categories of information; Data Fields point to "homes for things" — the receptacles for containing data items. Fine. We have no quarrel with this. The problem comes in the fact that conceptually the essayist is mixing a Data Element with a Data Field. DATE OF BIRTH, for example, he would call a Data Element, which is incorrect. The Data Element is DATE, which has a standard definition and an approved configuration for recording it: YYMMDD. DATE OF BIRTH is a *field name* or Data Use Identifier. This latter term is not a red herring, thrown out to confuse people, as our author states. It is a well-respected term, defined in *Funk and Wagnalls Dictionary of Data Processing Terms* as: "A name, title, or description that specifies the intended use of a Data Element."

The main point here is that Data Elements and Data Use Identifiers (or Data Fields) are different and the DED/D should carefully demarcate them. A closely related point is that data standards is concerned not only with the pure gold of the Data Element, but also with the way one *names* a Data Field and the *code* or *abbreviation* one gives it. This is all spelled out in the SIGINT directive that governs the standardization program. There is a standard way to generate a field-name code or abbreviation. The author of "Data Standards Without Tears" is right when he says the Data Element is not really the "thing" itself but the descriptive "name...of a 'set of things.'" Where he gets into difficulties is in not distinguishing carefully between a Data Element and its "use identifier." Data Use Identifiers really don't have separate *data items* of their own; only true Data Elements have data items.

A practical problem arises with the DED/D. Where do you put the "good" Data Field names (i.e., the Data Element/Data Use Identifier combinations that have already been standardized)? As [redacted] says, the run-of-the-mill Data Fields that John Jones used in his favorite file will appear in the Directory. If many other people use some of the same field name/abbreviations he does, we may have a clue as to something that needs looking at as a potential standard. We agree, but let us hope that the dictionary designers will not forget about the gold we already have, the standard field name/abbreviations referred to in the previous paragraph. Conceivably they could be stored in the DED itself, as long as the designers remember that these are not in themselves Data Elements. There is a considerable economy of storage here. You only have to store (in computer memory) the *data items* for a given Data Element *once* for each identifiable Data Element.

A related problem has to do with Data Elements not yet standardized, or not capable of being standardized. For example, Case Notation has developed over the years into something so complex that it now defies any attempt to standardize it. We can, however, give it a "reserved uniform code" (CASN) and encourage file sponsors to use this in preference to one of their own invention. The NDSC has an online glossary of such Data Elements commonly seen in SIGINT files. Many are labelled "potential" data standards, but it may be quite a while before they can be introduced into the standardization process.

"...the case of a file or software system which exists before the standard is set up, where the effort required to change it is unacceptable."

Usually a sponsor cries "unacceptable" just because he does not want to go to the trouble of reprogramming. It is more a matter of convenience than operational necessity. A standard is not adopted until thorough discussion and coordination throughout the affected Agency elements have shown the NDSC that *all* users are able to implement it. The article merely supports parochialism by letting personal whim or convenience get in the way of implementing standards. The complaint about the "unacceptable effort" required to conform to an approved standard is often accompanied by one, or both, of the following statements:

> "Standards are fine, as long as they don't conflict with those we've already set up in the project."
>
> "I'll support standards 100% — so far as I possibly can."

"To sum up, standards cannot be created in a vacuum. They must be developed from current usage..."

Standards *are* created from a demonstrated need, not just dreamed up by the NDSC. We try to look at the needs of the entire Agency as regards a particular proposal and not just at the usage that has happened to evolve. Being able to identify current usage is important, though, and the coming DED/D should be very helpful in this area.

P.L. 86-36

"There are two ways of tackling standardization: the easy way and the impossible way."

Yes, at the NDSC we sometimes feel that our job is impossible. We deal with abstract concepts which are often exasperatingly hard to pin down. It would be great to find an easier way. We will be happy to see the DED/D emerge as an electro-mechanical friend who can give us a hand. It will be nice to have the DED/D document the "real world" and the standards world. I suspect, though, that there will still be a lot of blood and sweat, even without the tears. (U)

NSA-crostic No. 23

By Arthur J. Salemme, A.E.
(Acrostician Emeritus)

The quotation on the next page was taken from the published work of an NSA-er. The first letters of the WORDS spell out the author's name and the title of the work.

DEFINITIONS

WORDS

[redacted]

(Solution next month) A.J.S (U)

P.L. 86-36

Fairbanks on English

Some of the Agency's best writing on writing can be found in the essays written by Dr. Sydney Fairbanks while he was the editor of the NSA Technical Journal, *from 1956 to 1959. Most are as timely today as they were two decades ago. Here, from the October 1957 issue, is the opening salvo in his battle against "English as she is wrote in the Agency."*

We have decided that an editorial should not be mere persiflage. It should initiate reforms, strike blows for freedom, speak for the oppressed,—that sort of thing;—provided always that the Editor sticks to what concerns him. This matter of English as she is wrote in the Agency is something that inevitably concerns him. We have therefore purchased a small red flag, and are planning a series of manifestoes.

The other day a D/F [*Disposition Form, a long-defunct form for interoffice correspondence*] crossed our desk. It has been said that everything in Government is done by a D/F, but you have to be here a year or two to appreciate what a d.f. he is. This, however, is beside the point. The D/F in question was highly practical and intelligent, and it bore a rubber-stamp signature of an altitude that virtually guaranteed that the signatory neither wrote it nor read it. Nevertheless *someone* must have written it, and it is to be hoped, or feared, that someone read it. The third paragraph runs: "It shall continue nailed to the skull, however it will be removable with patience and a corkscrew." Or at least...perhaps we should explain that tact has prompted us to alter everything but the sentence structure, the comma, and the "however." It is these that we wish to discuss.

Of course there would be no point in such a discussion if the error in question were not extremely common. A friend who has to waste a large part of his time revising reports and letters written by subordinates tells me that he expects to meet it at least once a day, and wonders why this particular comma splice is preferred above all others.

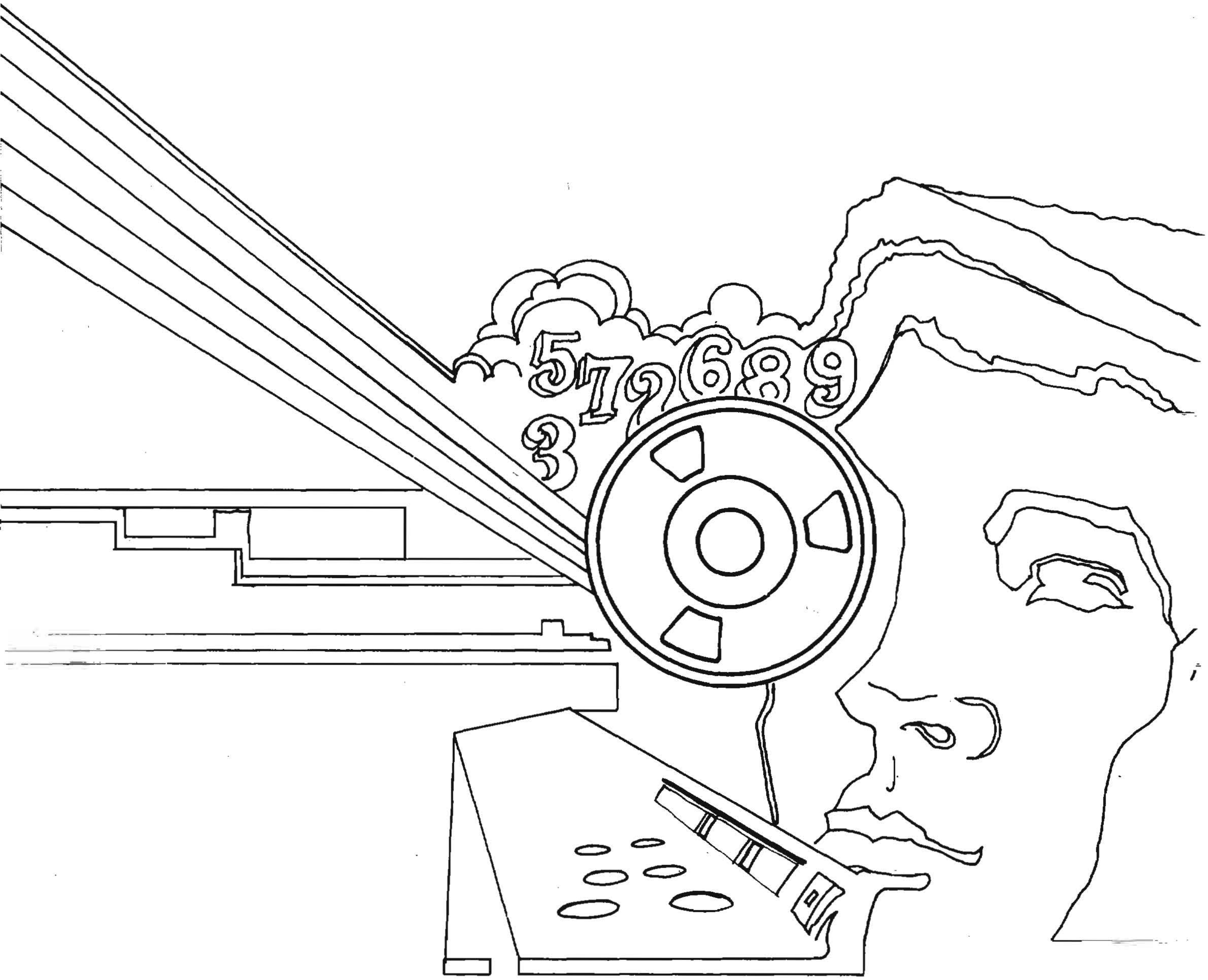
Alas, the answer is fairly clear. The sentence in question reads perfectly well if "but" is substituted for "however," and the question boils down to why the typical composer of D/F's says "however" when he means "but." He does it for the same reason that he says "presently" when he means "now." All you have to do is to count the syllables. If—and such things have happened—he wants to tell people to stop using long words in their letters, he will write, "discontinue the employment of ultra-lengthy terms in the correspondence presently emanating from your organization," without a qualm. Nothing less than a time-tested trisyllable is an adquate figleaf for his literary modesty, and the demand has created the supply.

Instead of working against nature, by trying to substitute the short word for the long, the general tendency of those who edit has been to modify the punctuation: "...nailed to the skull. However, it will be removable..."; thereby producing something that is merely clumsy. There is a legitimate use for "however" at the beginning of a sentence, where the essentially contrasting nature of what follows is to be not merely indicated but emphasized. There may even, conceivably, be an appropriate occasion for starting a sentence with "Therefore," although it is roughly equivalent to entering a room by flinging the door open with a crash and stamping on the threshold. But some deep and inscrutable instinct, like that which drives the lemmings to commit suicide, urges the D/F writer to begin every sentence with one of these two. Given the idea: "It is strong enough, but it is too large; better try something else," he can be counted on to express it: "It is strong enough. However, it is too large. Therefore, you should try something else."

If we were—fond, impious thought— one having authority, saying to one man Spell, and he spelleth, and to another Punctuate and he punctuateth, we would issue a D/F decreeing—in appropriate terms,of course— that in future no sentences would start with the words "however" or "therefore",—and then sit back and listen in grim glee while the electric typewriters ground to a halt and silence settled in the corridors. Some mute inglorious Milton would then discover for himself the possibility of writing, "We have, however..." and "It is, therefore..." and presently everything would start humming again. But the quality of the product would be, to our mind, appreciably improved.

Selah.

(U)


3
5
7
2
6
8
9