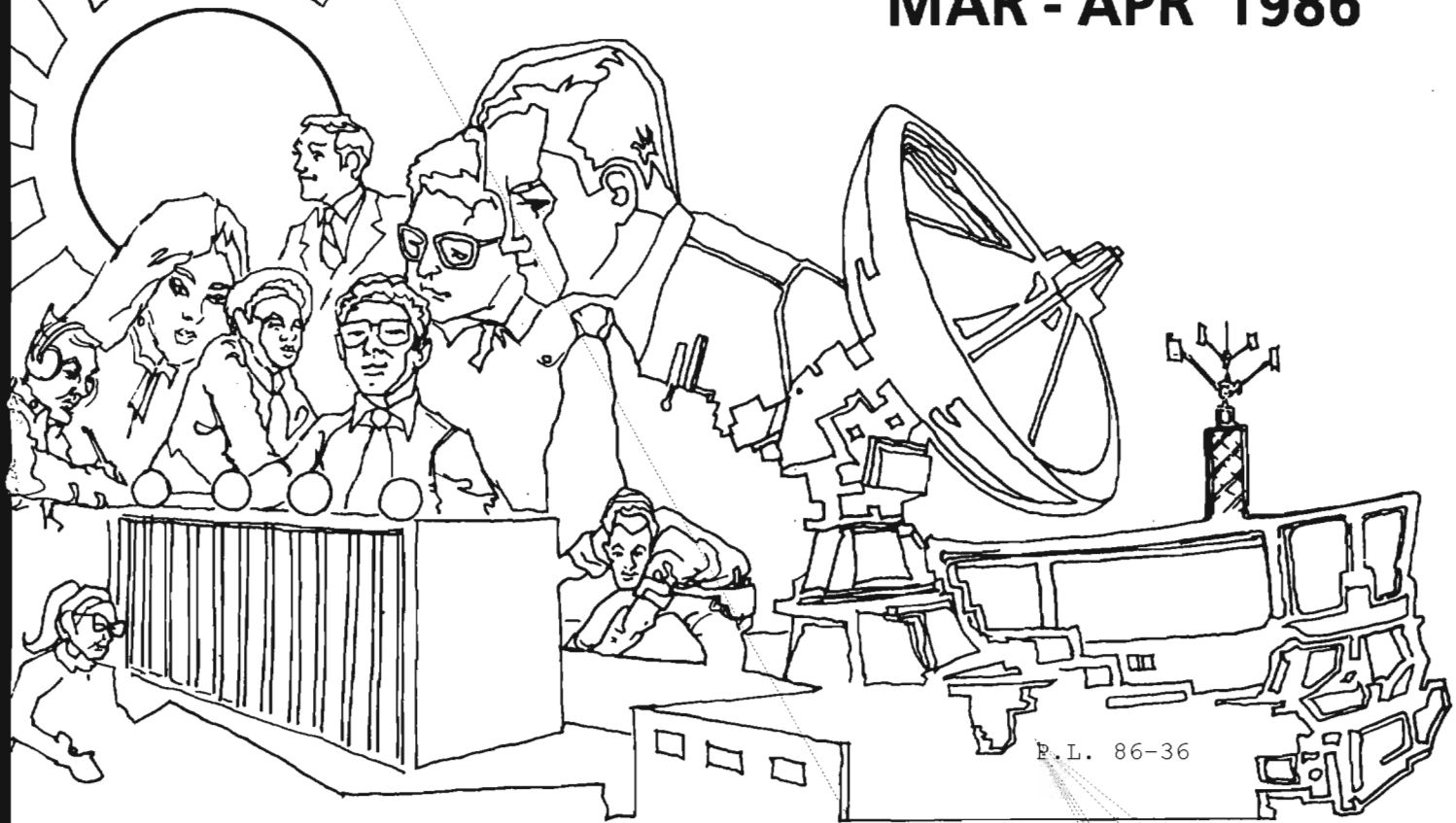
~~TOP SECRET~~

NATIONAL SECURITY AGENCY
FORT GEORGE G. MEADE, MARYLAND

CRYPTOLOG

MAR - APR 1986



~~HANDLE VIA COMINT CHANNELS ONLY~~

~~TOP SECRET~~

~~NOT RELEASABLE TO CONTRACTORS~~

~~CLASSIFIED BY NSA/CSSM 123-2~~
~~DECLASSIFY ON: Originating Agency's Determination Required~~

CRYPTOLOG

Published by P1, Techniques and Standards

VOL. XIII, Nos. 3-4 ..... March-April 1986

PUBLISHER .....

To submit articles or letters by mail, send to:
Editor, CRYPTOLOG, P1

If you used a word processor, please include the mag card, floppy or diskette along with your hard copy, with a notation as to what equipment, operating system, and software you used.

via PLATFORM mail, send to:
cryptolg at bar1c05
(bar-one-c-zero-five)
(note: no 'o' in 'log')

Always include your full name, organization, and secure phone number.

For Change of Address
send name and old and new organizations to:
Editor, CRYPTOLOG, P1

TIGHTENING OUR BELT (U)

Like everything else in the federal government, CRYPTOLOG has been affected by budget cuts, but to an even greater extent. As a Class 1 publication in the Defense Department we have been ordered to cut by 55%.

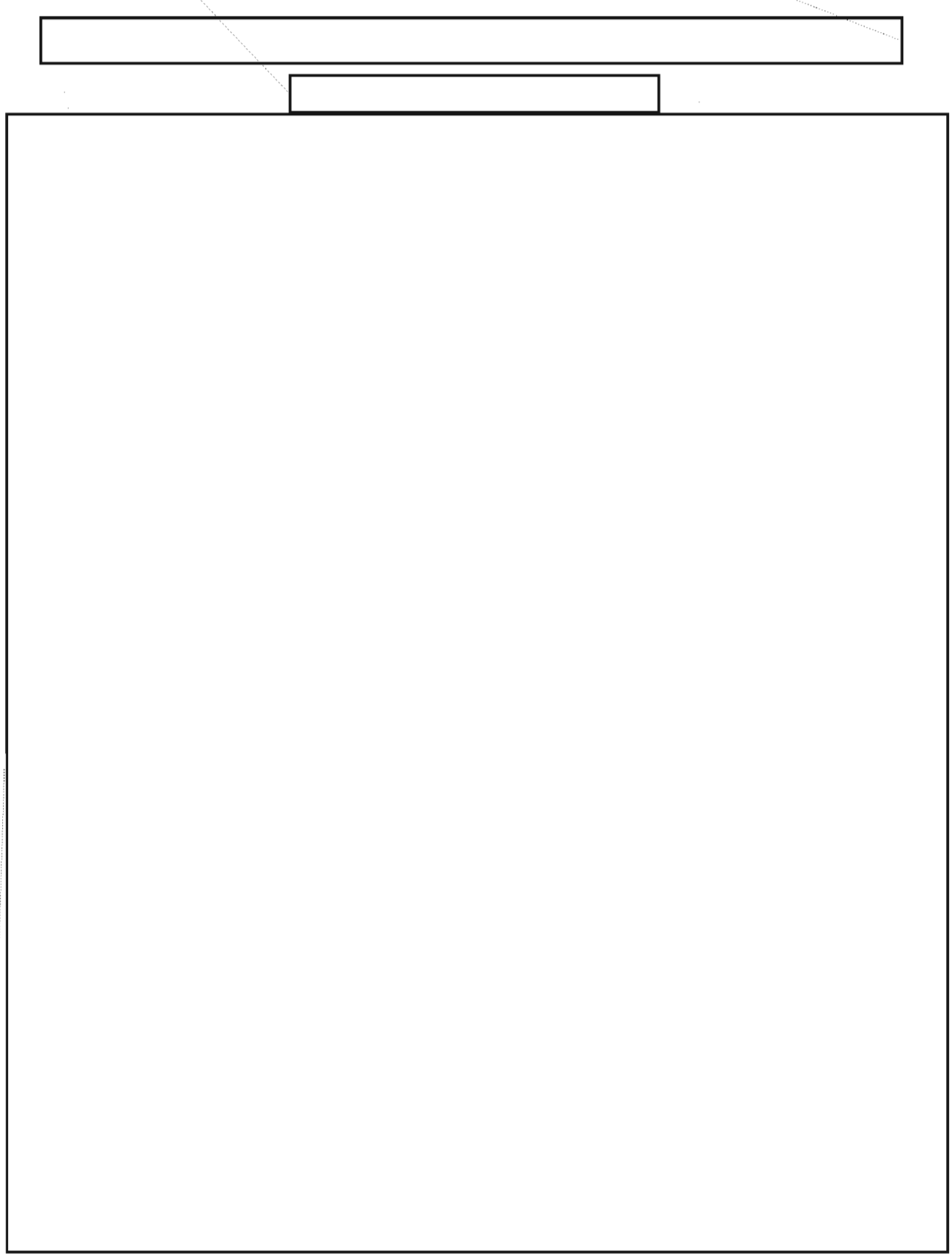
That's a lot for a spartan periodical with nary a frill, ruffle or furbelow to do without. We don't use color, the paper is minimum quality usable on a press, and the format is the most cost-effective. Even so, in the past year we've done what we could to bring costs down:

- For word processing we've moved to the Xerox Star from the Unix Editor on the BARDOLPH TSS, resulting in impressive savings in labor costs, especially in making printouts and in doing layouts;
- To avoid retyping we're using the Data Conversion Center for converting floppies made on other word processors;
- We've gone over to the high-speed Solna press which is more economical but prints only multiples of 16 pages;
- We're publishing half as many issues;
- We are down to a one-person staff (the editor) who has other duties as well;
- We are trimming the distribution list.

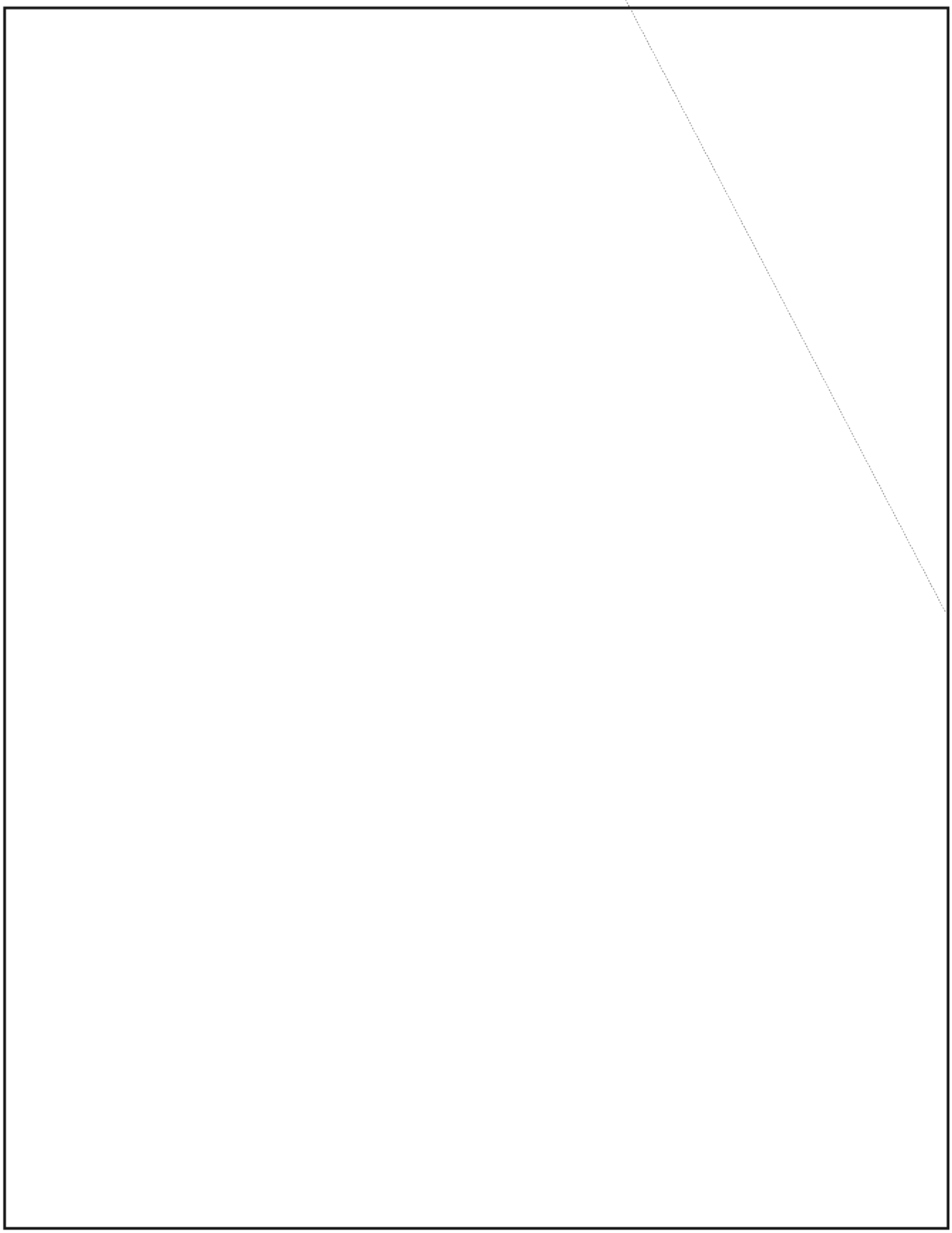
We can use your help on this last item:

- If you are going to the field or into retirement, let us know so that we can remove your name from distribution. (Organizational, not personal copies, are sent to the field. You can have a personal copy again when you return.)
- If your organization is now smaller than it was, request fewer copies.
- As the number of personal copies in your element increases, eliminate an appropriate number of organizational copies.

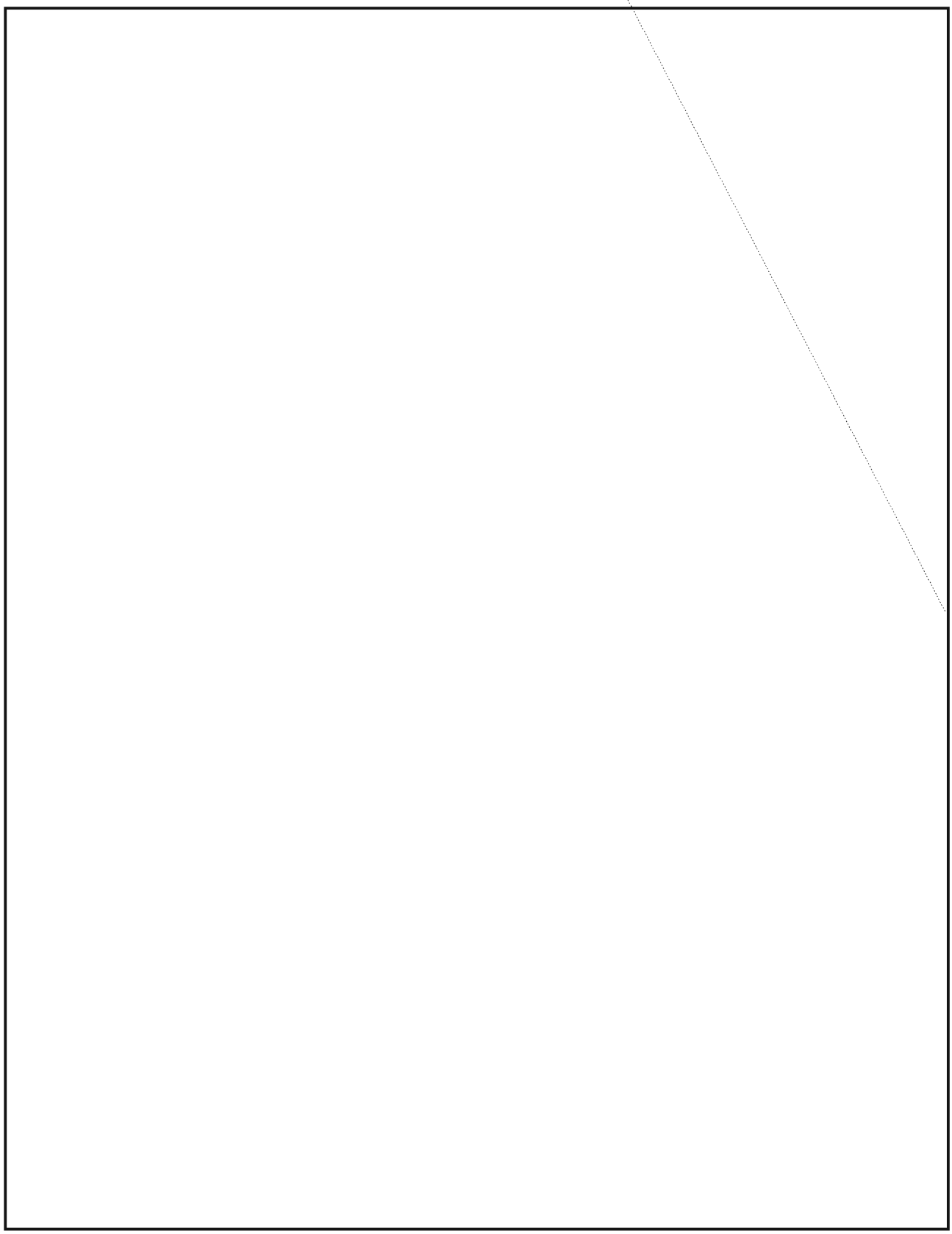
Thanks.



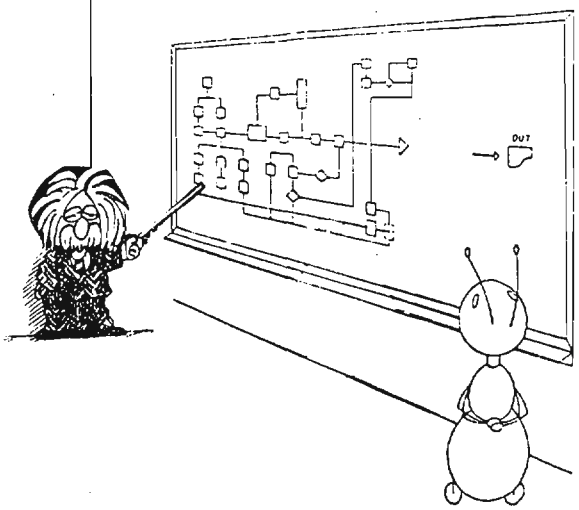
~~SECRET~~



~~SECRET~~



P.L. 86-36



EXPERT SYSTEMS--FOR NSA?
(u)

One of the more profitable advertising gimmicks in computer software today is the use of the word "expert." Numbers of products claim to be "expert systems" or to have "expert" capabilities. After we cut through the hype, do we find that expert systems actually have any value? When are they useful? Can we use them at NSA?

WHAT CAN AN EXPERT SYSTEM DO?

In general, an expert system starts with a set of facts, obtains more facts from the user through interaction, and,using a database of knowledge about the application, produces a reasoned conclusion. The system can, in most cases, explain its conclusion by displaying the logic chain.

An expert system is very good at applying heuristic rules (rules of thumb) to a set of loosely related data, and deriving a conclusion -- much like an expert does. It has been said that an expert is one who jumps to the right conclusions, because the expert picks just the few facts that are important to a conclusion, often bypassing complex computations, and arrives at a conclusion while other people are still sifting facts.

An expert system on a computer can be used to record the heuristic rules that a particular expert uses -- to "clone" the expert. This can be used to spread the expertise around or to substitute for an expert who is leaving. It can also be used to train new experts -- to organize the expertise so that it can be taught. The expert system leads the student through a correct logic chain with sample data, an improvement over traditional computer-aided instruction systems that merely quiz a student repeatedly.

Putting rules into a knowledge base can help to organize knowledge that was not well understood or not centrally located. Books and papers can also be used for these purposes, but the user interaction with the expert system gives it an advantage. Also, the expert system actually tries to apply the knowledge; it does not just serve as a repository for that knowledge.

An expert system could be used to explain rules and regulations, particularly those that have accumulated over time and are not logically consistent. Expert systems have been used most successfully to diagnose problems in expensive, complex mechanisms.

WHAT IS AN EXPERT SYSTEM?

I use the term "expert system" narrowly, to refer to software that separates the application domain knowledge from the program's control structure and that can explain its reasoning to a user. The repository of domain knowledge is commonly called the "knowledge base" (as in "data base") and the program control structure an "inference engine." This is a specialized program to read the knowledge base and process it logically to derive some conclusion or conclusions.

The most common class of expert system building tools, or shells, on the market today (there are dozens of them) use a knowledge base of rules in the form of "IF some condition or conditions THEN some condition or conditions." The conditions usually are expressed as a parameter/value combination, and may refer to other rules, which are determined to be true or false in a chain of logic going back to data entered by the user or sensed by the system externally. This chaining usually goes backward from a "goal

rule" to the original data, but may go forward from a set of data to one or more conclusions (see box). Key to the usefulness of the expert system is the fact that the sequence of rule use is data-driven and may vary drastically in different runs. The inference engine selects rules, as needed, and chains through the knowledge base in the most efficient manner.

Most expert systems have some ability to handle uncertain data. A piece of data can be true, not true, or somewhere in between: probably, possibly, not likely, almost certainly, almost certainly not, etc. Data that is "probably true" pushes the system in the direction of a positive conclusion, and data that is "probably false" pushes the system in the opposite direction.

In addition to dealing with uncertain data, most expert systems have some mechanism for handling unknown data, by assigning a medium value, or extrapolating from known data, or by ignoring the rules involving unknown data.

Some expert systems have mechanisms for representing knowledge in some form other than simple rules; e.g., complex data objects that can have different kinds of values, or attributes, and can inherit attributes from certain other objects.

PROBLEMS WITH EXPERT SYSTEMS

Some knowledge simply doesn't need to be organized in a randomly arranged data base. The steps follow each other in a purely sequential fashion, and the solution can be programmed more simply and efficiently in a traditional programming language.

The structure of knowledge representation is very important. Considerable thought needs to be given to this, just as you would give considerable thought to the design of an important and complex computer data base.

Inexact reasoning is a black hole. Although it is possible to combine values that are inexact, no one has proven that one way of doing this is better in a general sense than any other way. It appears that some methods are better for some applications, some for others, but the selection of a method is not well understood.

Conflict resolution can be tricky. The inference engine matches patterns of rules to find the best rule to apply in any particular instance. If two or more rules match the pattern, some conflict resolution technique must be used. There is not a good understanding of how to resolve these conflicts.

Expert systems tend to use up large amounts of computer resources -- both time and memory. Most of the smaller systems are severely limited in the number of rules they can process (one of the most successful expert systems, DEC's XCON, uses 8000 rules; a system which runs on an IBM PC might be limited to as few as 100 rules).

SAMPLE RULES: KNOWLEDGE BASE

1. if SCREEN is BLANK
   then POWER is OFF (0.5)

2. if KEYBOARD is not WORKING
   then PROGRAM is HUNG (0.4)

3. if DISPLAY is ERROR
   then PROGRAM is HUNG (1.0)

4. if PROGRAM is HUNG
   then FIX is CTL-ALT-DEL (0.5)

5. if POWER is OFF
   then FIX is SWITCH POWER ON (1.0).

6. if FIX is KNOWN
   then DIAGNOSIS is COMPLETE.

Forward Chaining: The system tries rule 1; if the screen is blank, then rule 1 concludes that the power is probably off (confidence factor of 0.5). The system then looks for (chains forward to) a rule which is true if the power is off (rule 5). Rule 5 then says that the fix is to switch the power on. Once the fix is known, rule 6 concludes the diagnosis. If the screen is not blank (rule 1), the system tries rule 2, and then rule 3, until it either exhausts the knowledge base or finds a rule that works (whose premise is true). Any chain of rules which reaches the goal rule (rule 6) ends the program.

Backward Chaining: The system starts with the goal rule (rule 6). If the fix were known, it could stop. It therefore looks for a rule which, if true, would make the fix known (rule 4 or rule 5). The system picks rule 5 and checks its premise (is the power off?). Rule 5 chains backwards to rule 1; if the screen is blank, the power is probably off, and the fix is to switch the power on. If rule 1 is false, the chain fails, and the system must try another chain; it tries rule 4, which chains to either rule 2 or rule 3. The system continues backtracking through the knowledge base until it either completes a valid chain or exhausts the rules.

EXAMPLES OF EXPERT SYSTEMS

Until recently, all expert systems were built to run on mainframe or mini computers. Only in the last couple of years have PC-based tools entered the market.

One of the oldest expert systems is MYCIN, developed at Stanford University to diagnose infectious diseases and to prescribe antibiotics.

Although MYCIN has never been used operationally, its techniques have been copied by many commercial expert system packages.

One of the most successful expert systems is DEC's XCON, which configures DEC computer systems. Using 8000 heuristic rules, XCON takes a customer's order and identifies all the parts that are needed to fill the order, making sure that they all fit together. DEC estimates that XCON saves the company at least $18 million a year. Before XCON was built, the human experts found the task boring most of the time and therefore made numbers of mistakes. Traditional programs had been tried but didn't work; the rules kept changing, the knowledge was scattered, and the totality of the logic was not well understood.

More recently, and on a smaller scale, the Campbell Soup company built a small expert system on a PC to diagnose breakdowns in their giant soup cooking machines. A few hundred rules enabled a non-expert to diagnose most problems effectively.

The General Electric company built an expert system to diagnose GE locomotive problems. The improvement in repair time was so dramatic that GE's market share increased by a large margin.

Autometric, an NSA software contractor, is developing an expert system to diagnose software problem reports. They estimate that the prototype system, with about 200 rules, correctly identifies a problem module about fifty percent of the time, saving about four hours of diagnostic time each week. Eventually, the system should enable a relatively inexperienced programmer to solve most software problems. One of the chief reasons for developing this system was that the experts who had designed and built the software didn't want to maintain it, and staffing became a problem.

Burroughs has a prototype expert system to diagnose personal computer problems for users calling a Burroughs hot line. This hot line has been staffed by experts, who are bored to tears by the mundane problems posed by most callers. The expert system will enable an unskilled person to answer the hot line and effectively diagnose most user problems. The experts will then work only on the more challenging problems.

[redacted] in R8, has developed a PC-based expert system to identify the classification of cryptanalytic information, according to a set of rules published by P1.

R53 researchers are working on a number of expert systems, including one to support DEFSMAC. Other expert system experiments are underway in G4, P1, W1, A2, and T3.

BUILDING YOUR FIRST EXPERT SYSTEM

Pick a problem. Use the criteria described in "Selecting Your First Application," and add other criteria if they are important to you. Pick a problem you know well, one that looks too small to bother with (because it probably will take more work than you thought), but pick a problem that matters to someone -- preferably your boss.

Pick an expert. You want someone who really is THE expert, and who wants to help. If you are the expert, so much the better.

Develop a prototype. Set some relatively short period of time, and see what you can do in that period. The schedule depends on your problem, of course, but three to six months should be enough to accomplish something that can be evaluated.

Evaluate the performance of your prototype system. Does it make a good decision or provide satisfactory advice? Evaluate the reasoning process: do those right answers come from a valid line of reasoning? Is the user interface acceptable for operational use? Does the system run efficiently enough to be usable? Will it still be usable when you get the full system (more rules) built? Will the system be cost-effective when compared to the current (manual) way of doing it?

Proceed with incremental development of an operational system, if your evaluation produces acceptable answers to your questions. If the answers don't look good, you need to fix whatever is wrong, or go back and try a different application (or, possibly, you need a different expert system building tool).

Prepare the users for using the system. This is the most important item, if you are building an operational system. DEC's XCON almost failed, after three years of research, because the people doing the work manually thought (erroneously) that the system would replace them and they would lose their jobs.

SELECTING YOUR FIRST APPLICATION

Your application doesn't have to match all of these criteria, but the more items it matches, the better your chances of success.

- **Traditional technology doesn't work.** Before DEC hired CMU to build R1, they tried three times to write the program in FORTRAN.
- **The knowledge domain is narrowly defined and primarily involves logical deduction.** One way of looking at it is to ask if a human expert could solve the problem over the telephone.
- **Recognized experts know how to solve the problem.** You will have trouble teaching a machine how to solve it if you don't know how yourself.
- **The task involves a few minutes to a few days of an expert's time.**
- **A traditional approach to solving the problem requires combining a large number of steps in a large number of complex sequences.**
- **Wrong answers will not result in loss of life, or other unacceptable consequences.** Because of the heuristic rule approach, optimal results are not guaranteed.

- **The resulting system will have a high payoff.** You don't want to put a lot of effort into a system that doesn't really do anything.

[redacted] In hardware or software support, an expert system might help diagnose problems in large, complex computer systems, or analyze network performance problems. An expert system might assist with computer language translation when the two languages involved have different structures (for example, from FORTRAN to Ada).

GETTING STARTED

If you are interested in developing an expert system, look around; there is probably someone not too far away who is experimenting with the technology and can help you. If you don't find someone close at hand, call [redacted] He is chairman of the NSA Artificial Intelligence Working Group, and he can put you in touch with a number of people to talk to. □

BULLETIN BOARD

[redacted]

Dave Fitzpatrick E42 968-8418

[redacted]

Bill Patterson B5093 963-3490

[redacted]

SOLUTION TO NSA-CROSTIC No. 62

[Arthur J.] Salemme, "Joys of Plural Dropping," *CRYPTOLOG*, January 1978.

"When our daughter and a friend of hers were both five and were eating lunch at our house, my wife gave them each a sandwich and a slice of American cheese cut in strips. The little friend particularly [enjoyed] the latter and asked what they were. `Cheese,' she was told. `Well, then,' she asked, `can I please have another chee?' "

AN INTRODUCTION TO AI (U)

This is an abridgement of an article that appeared in the February 1986 issue of the Cryptanalysis Intern Bulletin.

Artificial intelligence (AI) comprises elements of many disciplines. It was once explored only by academics; now it has become a topic of vital interest to both government and private industry. It is worthwhile for the cryptanalytic community to familiarize itself with developments in AI because of potential applications to cryptanalytic tasks.

The history, characteristics, and applications of artificial intelligence provide a fascinating exploration into the essence of the interaction between man and machine that characterizes contemporary human existence.

DEFINITION AND HISTORY

Although the discipline known as AI has been in existence for over thirty years, no single definition is accepted by all the experts. A survey of definitions given by leaders in the field, however, has yielded these concepts in common:

1) AI seeks to enable a machine to perform

- tasks traditionally thought to require human intelligence,

- in a manner approaching the working of the human mind.

This pursuit includes research in the cognitive sciences, in an effort to understand more fully the human capabilities upon which intelligent machines are to be modeled.

2) AI is distinguished from conventional computer science by its focus on symbolic rather than numeric (qualitative rather than quantitative) processes.

New capabilities for enabling machines to imitate man have been created as a result of fundamental developments in diverse fields, including computer science, the psychology of learning, physiology, and information science. The work of cryptanalyst Alan Turing (hypothesizing a machine that could perform mathematical operations in a binary format and developing a test to evaluate whether a disguised correspondent was machine or human), Claude Shannon (applying Boolean algebra to electronic circuitry), and Norbert Wiener (cybernetics) are only a few examples of contributions that made AI possible.

The term "artificial intelligence" appeared concurrently with the first major event to bring together from a variety of academic backgrounds those who shared a common interest in enabling machines to perform intelligent tasks. The event was the 1956 Dartmouth Conference, organized by John McCarthy and attended by others who were to become prime movers in the field in subsequent years. Although the Dartmouth Conference did not yield the results predicted, it did set the atmosphere for interdisciplinary cooperation and challenge. In addition, two conference participants, Allen Newell and Herbert Simon, both connected with Carnegie Tech (now Carnegie Mellon) and the RAND Corporation, came to the conference with a working program demonstrating the intelligent processing of information that conference participants had discussed and sought.

During the 1950s and 1960s, expectations were high that machines could be programmed to imitate human thinking in the realm of common sense, that is, everyday acts of reasoning that are performed by most people without conscious recognition of the processes involved. As the passing years revealed the unexpected complexity of this task, AI in the 1970s was marked by a philosophical and methodological shift toward specialization. Programs were written that would enable a particular machine to perform intelligently within a very limited domain. In addition, whereas once the goal was to equip a computer with a general set of theories that would enable it to make a wide variety of decisions, the

focus shifted to providing an impressive array of facts about a given topic.

Since the late 1970s, a major trend in AI has been commercialization. What was once a topic that concerned academics and a limited number of research corporations has now become an industrial and a governmental fad of sorts, especially within the US Department of Defense. Moreover, AI is an integral part of an emerging technical war between Japan and nations of the West. In 1982, Japan announced a massive computer research initiative to take place during the next ten years. Entitled the "Fifth Generation of Computer Systems," the project aims to put Japan in the forefront of world computer technology by the 1990s. Many of the project's goals fall into the area defined as AI. Response to the challenge was not slow in coming: Britain, Europe and the United States initiated efforts related to those of the Japanese. AI is included in DARPA's (Defense Advanced Research Projects Agency) ten-year development program and in the plans of MCC (Microelectronics and Computer Technology Corporation), a corporation of 21 companies under the leadership of former NSA Director Bobby Inman.

AI PROCESSES AND APPLICATIONS

AI software has developed in channels largely unknown to conventional programming languages. Instead of using essentially linear reasoning, AI programs endeavor to imitate the flexibility of the human mind. LISP, developed in 1958, is the most popular programming language used by AI experts in the United States. Short for List Processing, LISP allows the organization of data in network-like structures. "Items" of data may be descriptive statements, lists, or many other data groupings. "Frames" and "scripts" are two popular ways of storing information. Frames contain slots into which data is inserted; scripts have a similar structure, including a series of scenarios into which the data might be expected to enter.

The hardware associated with AI falls into two categories: computers designed especially for AI usage, and AI adaptations of other computer systems. Advertisements describing AI systems reveal AI-related innovations: use of "windows" and a "mouse" pointer to allow greater flexibility in examining the relationships between data sets, the adaptation of computer graphics to capture the essence of symbolic processing, and a variety of developmental tools to assist the programmer in formulating programs that exploit all of the networking potential of AI languages. Many of these features are being used in areas other than those associated specifically with AI.

One AI application that is now receiving much attention is the development of expert systems. This is described in another article in this issue.

Other areas of artificial intelligence are also the objects of intensive research efforts. Natural language processing is the ability to program and receive feedback in a conversation-like context. Computer vision and speech recognition are receiving attention from DARPA for their defense potential. In addition, robotics, educational computer interactions, and a variety of office workstation uses for AI are currently under investigation, and in some cases, implementation.

PROSPECTS FOR AI

It soon becomes apparent that scientific opinion is divided regarding the ability of AI to substantiate its claims. The early AI pioneers were premature in declaring their ability to convert the mysteries of human intelligence to machine form. Although the people currently at the forefront of AI development communicate a more cautious and realistic outlook, the recent transformation of AI into a prime business venture has subjected the discipline to the exaggerations of commercial advertising.

While AI is still in development, significant contributions have already come from AI research and technology. AI professionals are quick to recognize that the potential of machine capabilities has barely been tapped. The next decade should prove critical to the question of whether AI is able to prove itself to the rest of the scientific community, as the concerted efforts of government, universities and industry seek to overcome the current limits of computer technology.

ACKNOWLEDGMENTS

A major source of information was Agency personnel who are currently active in AI research. [redacted] were all generous in sharing their time and personal resources in helping to explain AI.

P.L. 86-36

Books and Articles Used

[redacted] "Artificial Intelligence Expert Systems," NSA briefing,1985.

McCorduck, Pamela. *Machines Who Think: A Personal Inquiry into theHistory and Prospects of Artificial Intelligence.* San Francisco: W.H. Freeman, 1979.

Mishkoff, Henry C. *Understanding Artificial Intelligence.* Dallas, Texas: Texas Instruments, 1985.

Waldrop, M. Mitchell. "Artificial Intelligence." *Science* reprint (from 24 February 1984, 23 March 1984, 27 April 1984, 15 June 1984, 10 August 1984 and 14 September 1984). □

★ LEVELS OF ★ CLASSIFICATION (U)

★ [redacted] P05/SAO

(U) Recognized classification levels and the factors involved in establishing them are as follows:

~~(TS-CCO)~~ **TOP SECRET** shall be applied only to information or material when its unauthorized disclosure could reasonably be expected to cause exceptionally grave damage to the national security.

~~(S-CCO)~~ **SECRET** is used for information or material the unauthorized disclosure of which could reasonably be expected to cause serious damage to the national security.

(U) **CONFIDENTIAL** is the appropriate classification when the unauthorized disclosure of the information or material in question could reasonably be expected to cause identifiable damage to the national security.

(U) **UNCLASSIFIED** material will not normally be marked or stamped **UNCLASSIFIED** unless it is essential to advise the recipient of the material that it was examined in order to assign a classification, and was then determined not to require one.

"FOR OFFICIAL USE ONLY"

(U) The caveat **FOR OFFICIAL USE ONLY (FOUO)** is a unique marking in that, although it can be used to preclude public dissemination of certain information, it is not in itself a level of security classification. The rules governing the use of this handling instruction can be quite confusing, so we often decide, or are told by others, to mark an item **FOUO** without understanding its meaning, the impact of its use, or whether it may be legally used in a given context.

(U) Certain information that has not been given a security classification under the criteria for assignment of **CONFIDENTIAL, SECRET**, or **TOP SECRET**, may be marked **FOUO** and withheld from the public domain under Freedom of Information (FOI) Exemptions. Additionally, at NSA, such unclassified information is generally covered by Public Law 86-36, the National Security Act of 1959. Section 6 of this act states, " ... nothing in this act or any other law ... shall be construed to require the disclosure of the organization or any function of the National Security Agency, of any information with respect to the activities thereof, or of names, titles, salaries, or number of the persons employed by such Agency." Thus, this Act provides **FOR OFFICIAL USE ONLY** protection of our organizational designators, unclassified missions and functions, budgetary information, and personnel strength. Only information that has not been previously available to the public may be marked **FOUO**.

(U) Included among the FOI exemptions permitting **FOUO** protection are:

(1) Regulations, orders, manuals, directives and instructions relating to the internal personnel rules or practices of a DoD component, if such publication or inspection would not affect the general public, or if release to the public would hinder effective performance of DoD functions;

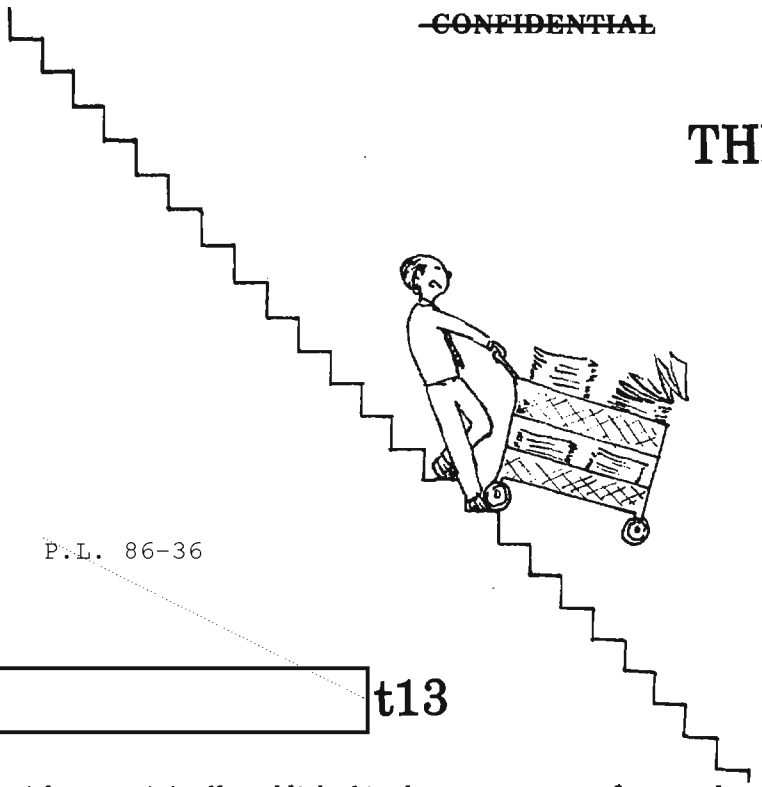
(2) Internal advice, recommendations, and subjective evaluations, as contrasted with factual matters, that are reflected in records pertaining to the decision-making process of an agency or DoD component;

(3) Information in personnel and medical files, and personal information in other files that, if disclosed, to the public, would result in clearly unwarranted invasion of personal privacy;

(4) Investigative records compiled for the purpose of enforcing civil, criminal, or military law.

(U) Please remember that, properly used, the **FOR OFFICIAL USE ONLY** caveat can afford us a great deal of protection against the public disclosure of unclassified information that is not in the public interest. □

THE 39 STEPS (U)



P.L. 86-36

t13

This article was originally published in the December 1980 issue of the T15 Technical Bulletin.

(U) In 1957 NSA was scattered around the Washington Area: at Arlington Hall, Virginia, at Nebraska Avenue in D.C., and in barracks at Ft. Meade, Maryland. The Agency had acquired a new building (today OPS 1) but it was not quite ready for occupancy. The Agency had also acquired a new computer - a second IBM 704. The decision was made to install the new machine at Ft. Meade rather than at Arlington Hall because that complex was scheduled to move. Having installed the machine it seemed a prudent thing to run it.

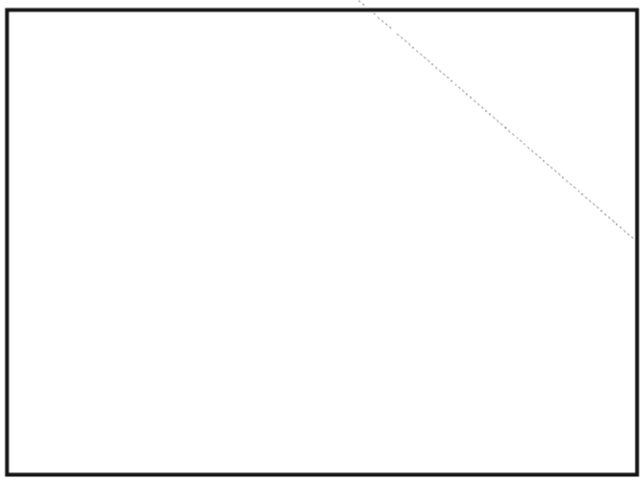
One Friday in October, 1957, at 11:30 p.m.

(U) The Baltimore-Washington Parkway was unusually dark that night. The fog masked the lights of the few southbound cars and even the warm glow from BINDERS (which at this time had yet to succumb to the arsonist) was not to be seen. The old Studebaker seemed especially reluctant to pursue its course to Route 32. The scene was something out of Rod Serling's THE TWILIGHT ZONE, though in this case more aptly it was the MIDNIGHT ZONE. Eventually the Studebaker groaned to a stop in front of Gatehouse 3. Carrying his brown bag and thermos, the operator passed by the nodding Marine guard and descended into the basement. The eve shift delayed its departure only long enough to tell its replacement that his partner on the shift had called in sick. The promise of a long lonely night was about to be fulfilled.

(U) The 704 occupied some portion of space which today belongs to the CAP area. The area was a restricted area and in fact at the time was one of a very few populated areas in the new building. A limited amount of production was running on the system since all customers, programmers, etc., were located at Arlington Hall and Nebraska Avenue. The 704 was one of the most powerful computers on the market. From an operational view it was a simple machine. It ran one program at a time. The operator's function was to load the program (it was on cards) into the card reader, set any necessary sense switches (there were 6), mount and ready any required tapes, check the printer, and press the program load button.

(U) The computer room was a sterile place. In addition to the computer, its furnishings consisted of two government issue gray desks, (one of which was placed in front of the machine console), a gray table, a gray coat rack, and a few gray chairs, all of which sat on a gray raised floor. This exotic color combination was offset by a pile of brown burn bags and a brown Frieden Calculator (circa 1812) which sat on the gray table. The atmosphere always convinced the operator that it was raining. He was constantly amazed when at the end of the shift he would emerge into sunshine.

(U) It was 1 a.m. The only sound in the room was the aching drone of the air handler punctuated by a periodic clacking of the printer. Sitting in front of the console, it was hard for the operator not to be mesmerized by the lights. He had heard of an operator who got some erotic pleasure from the console lights; but hard has he tried, he could receive only soporific sensations. He was running KNOLL IXC. Actually, the machine was running KNOLL IXC. It was a production job and it had been running for weeks. It was the only production job or any other kind of job available.



(U) It was too early for 'lunch' yet the emptiness of the room induced a similar sensation in his stomach. The bologna sandwich, potato chips, and the coffee consumption took 30 minutes. It was now 2 a.m., only 6 hours to go. There were three projects available. One, track down the cat that everyone agreed was somewhere under the false floor. Previous efforts at this hunt had proven fruitless. The operator decided against the project. The cat, he was sure, had its own secret entrance to this secure area.

(U) The second project was the repair of the Frieden Calculator. This electro-mechanical marvel had not worked since the day it came over from Arlington Hall. The fall from the truck probably had something to do with its condition. The operator and his partner had decided to repair it and in so doing had ended with 15 parts left over. Its present condition, however, was no worse, i.e., it still didn't work. The operator vetoed this task since the hammer had been missing the last two nights and such a tool was needed for the delicate repair procedure.

(U) The third project was do-it-yourself programming.

(U) In retrospect, the 50's were an extraordinary time for learning about computers. The Agency was in the forefront in the acquisition of hardware. ABNER, ATLAS, and the IBM 700, 705 and 704 were on board. But where were the people, the mechanics, who knew how to use them? They did not exist. As a result, the Agency embarked on a course of do-it-yourself. It provided its populace with a million-dollar machine, a manufacturer's manual, and said, "Learn It!" and they did.

(U) The operator's partner had spent some time looking at a dump of the KNOLL IXC program. He had determined where the program read the cards. He produced changes which read the pattern cards from tape rather than the card reader. The effect was, that after doing a "load card to tape," the operator never had to service the program for the entire shift, except for the printer. Not to be outdone, this operator, too, had been looking at the dump and had produced changes to put the printer output on tape, also. At 4 a.m. he was ready to implement his change. He loaded the program. It began. It read the pattern cards from tape. It cycled. It produced output ... ON THE CARD PUNCH !!!

(U) The operator recovered. The original version of KNOLL IXC was back in production. His version was in the burn bag. Tomorrow would be better. It was time to shut down.

(U) While there was a great deal of compute time available, the number of applications programmed were not enough to justify weekend work on overtime. The Agency was thus in the ideal posture for processing on demand, a posture never to be seen again.

(U) Since it was the weekend, Saturday morning, the operator began the shut down procedure. He powered down the tape drives, the printer, the main frame. He placed all tapes, all cards, all listings in the oversize shopping cart. He secured the area and pushed the cart down the deserted roadway to the elevator.

(U) At the time, when the building had yet to be approved for "beneficial occupancy," there were very few secure areas. In the North end (first floor) there was established a vault room for the computer operations activity. It had a 3-way combination lock.

(U) The elevator refused the operator's call. Try as he would, it would not come. What to do? The operator was glad he had been lifting weights. Pulling the loaded cart up the 39 steps to the first floor was not in his job description but he was not enough of a PMM lawyer to know it. It was a thing that had to be done.

(U) It was now 8 o'clock. The darn combination didn't work. Here he was, the fate of the free world in his shopping cart. He couldn't open the vault. Once again, "my age, my father's age, the playboy centerfold's age." Would it work? It worked.

(U) He left. It was not raining, Depositing his empty brown bag and his empty thermos beside him, he urged the Studebaker to start. It did. As the car staggered down the B.W. Parkway, the operator wondered if he had done any good.

(U) KNOLL IXC made a major contribution to what the Agency is all about. It is a fact of life, then, as now, that our operators are prime movers in successes which they know not of. □

CRYPTOLOG invites readers to submit vignettes of historical interest

PUZZLE PAGE

MORAL DISQUISITION

Find the five-letter anagram to fill all blanks

" ----- in Deo," so the saying goes

(One cannot ----- it, though the sense one knows.)

However, after "mon ----- " one day,

When ----- and port had both been cleared away,

I fell to musing, would God ----- a Cain

Who ----- the heads of men, instead of grain?

At jousts the knight with ----- to win a prize

----- his opponents down to half their sizes

With him is damned the conqueror

To murder, also he who ----- and burns.

●●●●●●●

Reprinted from the
ASA *Review*, Vol. 1. No. 4, July-August 1950

Solution to NSA-CROSTIC No.61

[A. J.] Salemme, "[Guide to Russian] Technical Translation," NSA, 1974.

> "The translator must ... determine the precise type of [rotary-wing] aircraft ... being referred to in Russian as ВЕРТОЛЕТ so that his rendition will conform to precise English usage. He must not ... avoid the issue by translating it in formal context as "chopper" or "eggbeater."

Solution to

CRYPTARITHM

(Jan-Feb 1986)

```
1 2 3 4 5 6 7   1 1 0 0 1 0
K R Y P T O S   3 2 4 0 6 7 5
+S O C I E T Y +5 7 9 8 1 6 4
C P K I I K C   9 0 3 8 8 3 9
```

Number the columns left to right, 1-7. Note that no two letters refer to the same digit.

Column 1 shows that **C** is greater than **S**. Hence column 7 shows that **Y = C-S**. Now column 1 shows that there must have been a carry from column 2, so that **1 + K + S = C**, so **Y = K + 1**.

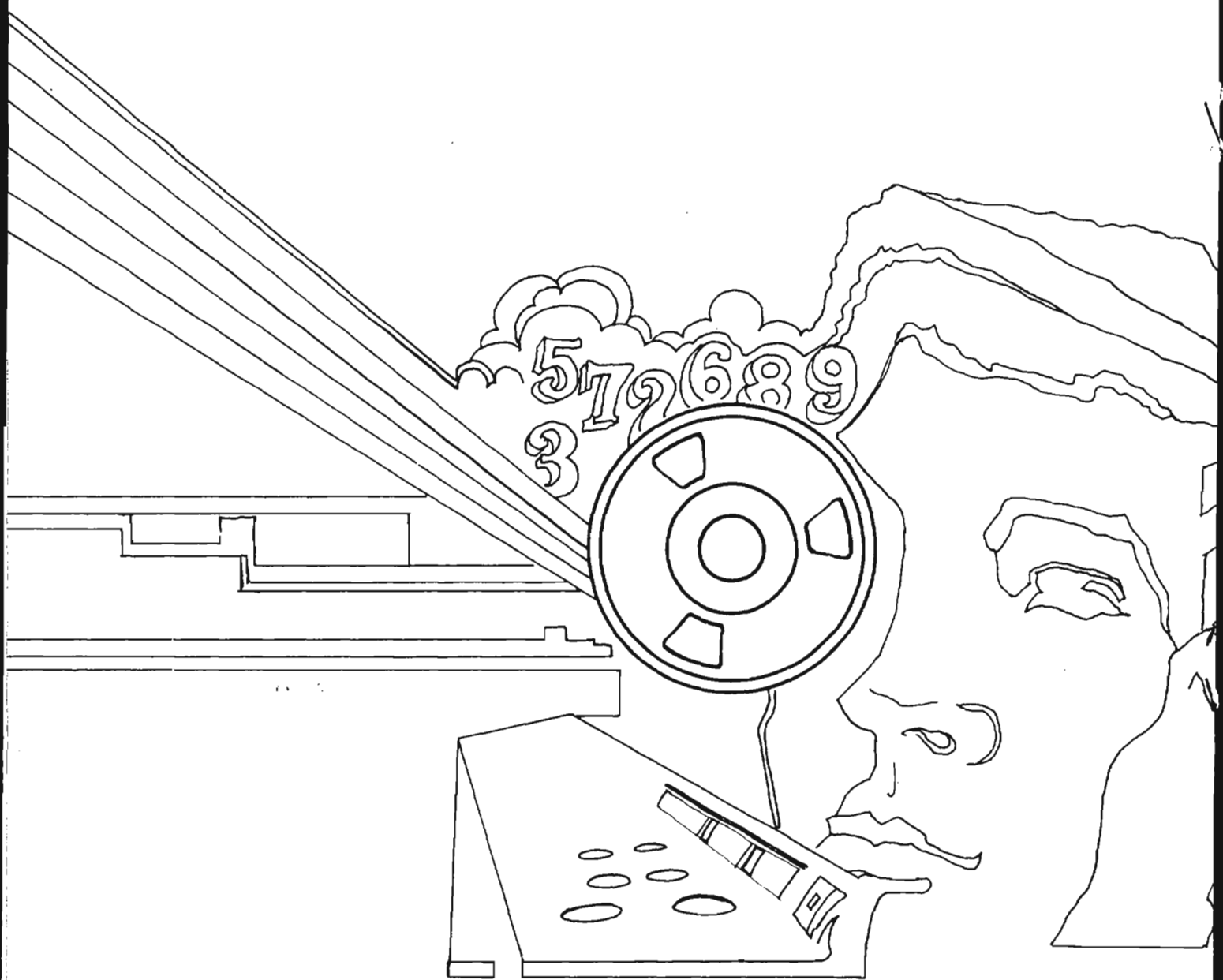
Since column 2 produces a carry, **P** in that column cannot be **9**. Therefore column 4 shows that **P** equals **0**.

Column 4 introduces no carry into column 3; since **Y = K + 1**, **C** must equal **9**. There must be a carry into column 2, so **R + O = 9**. And column 1 informs that **S + 8-K**.

Hence **K** cannot equal **0, 4, 8** or **9**. An assumption of any of the remaining six values for **K** gives tentative assumptions **Y** and **S**.

Letter **O** cannot equal **0, 9, K, Y, S**, or **9-S**, so there remain four possible values after a **K** assumption. Then **R = O-9**; column 6 gives **T = K-O**; column 5 produces no carry so **E** is less than **I** and must be the smaller of the two digits still unaccounted for.

The six **K** assumptions and the four **O** assumptions make for just 24 things to try, far fewer than the 10! possible permutations of ten letters. Continued analysis or a computer program leads to the unique solution **PERKY STOIC**. Alternatively, knowledge that the correct arrangement constitutes a phrase would quickly eliminate wrong assumptions; e.g., if **K** equaled **1**, the phrase sould have the unpromising beginning **PK**.


5726893