Aristocrat—An Intelligence Test for Computers

BY H. CAMPAIGNE

~~*Confidential*~~

The solution of monoalphabets was demonstrated on BOGART. This demonstration was interesting because it shows the power of BOGART, and because it forges another link in the chain of techniques needed for total automation.

Among puzzle addicts it is admitted that monoalphabets are the aristocrats of puzzles. In fact, a particular type of monoalphabet has come to be called an *aristocrat*, distinguished by its short text, spaces between words, and a bizarre vocabulary.

In cryptanalysis, monoalphabets are encountered in many places. They occur in busts when some changing element fails to change. Many cryptanalytic procedures have the solution of a simple substitution as a final step, the previous more sophisticated steps leading to an unknown wiring of a wheel or plugboard. And finally, monoalphabets are interesting in themselves as the simplest of all ciphers.

These two interesting aspects of substitutions become fascinating when viewed in the light of another recent development, the exploration of the flexibility of computers. Digital computer applicability to all kinds of problems is highly touted, but little is known about its limitations. For ten years claims have been made for language translation on computers, but only recently have any translations appeared. The quality of these translations is a matter of discussion; since there are no objective standards for accuracy and smoothness of literary works, these are described variously as "miserable," "usable," and "all that one could ask." So it is still not known how effective the computer will be. It is very rare to find a problem which it is known that a computer cannot solve; in most cases it is thought the computer could produce answers if it were programmed. Of course a computer cannot play tennis, nor weed the garden, nor do other obvious things. But the boundaries of computer ability have yet to be found.

The use of machines to aid cryptanalysis has been extensive at NSA. In some cases, problems have been carried almost from intercept copy to plain text entirely by machine. But of all the mechanization very little is on simple substitution. This is partly because people have not needed help, and partly because mechanization is difficult, surprisingly more difficult than for other "more sophisticated" operations.

The question posed here is: *How effectively can a machine solve simple substitutions?* which resolves immediately to: *How can a computer program be written to solve simple substitutions?*

This question I have attacked through a program called "Aristocrat." This was an adventure with many interesting aspects.

There are many versions of the problem. Will the cryptogram be spaced into words? If not, will it be spaced into groups?

The techniques to be applied can depend on the kind of plain text underlying the messages. Success is heavily dependent upon one's ability to predict text. What kind of plain text will we have in our problems? A series of messages about a military operation can be very redundant, full of "arrivals" and "departures," "reconnoitering" and "attacking." On the other hand, puzzles rely on having the most unpredictable text; "veal sables salute snooty ladies." Aristocrat was aimed at doing the problems in *Military Cryptanalytics, Part I*, specifically those with one hundred letters of text.

The number of techniques for attacking cryptograms is very large.

Aristocrat could easily become a major project. As it exists now, it suffers from many arbitrary restrictions imposed to save time or memory.

In summary then it seems that my shortcomings as a programmer rather than those of Bogart as a data manipulator have been probed. Aristocrat can read some of the cryptograms in Friedman and Callimahos. It could be made more flexible and more powerful, and I know how to do it if there were time, and that is by providing for a number of additional contingencies.

[redacted]

THE ARISTOCRATIC PLAN

[redacted]

[redacted]

CRITIQUE

[redacted]

(b)(1)
(b)(3)-50 USC 403
(b)(3)-18 USC 798
(b)(3)-P.L. 86-36

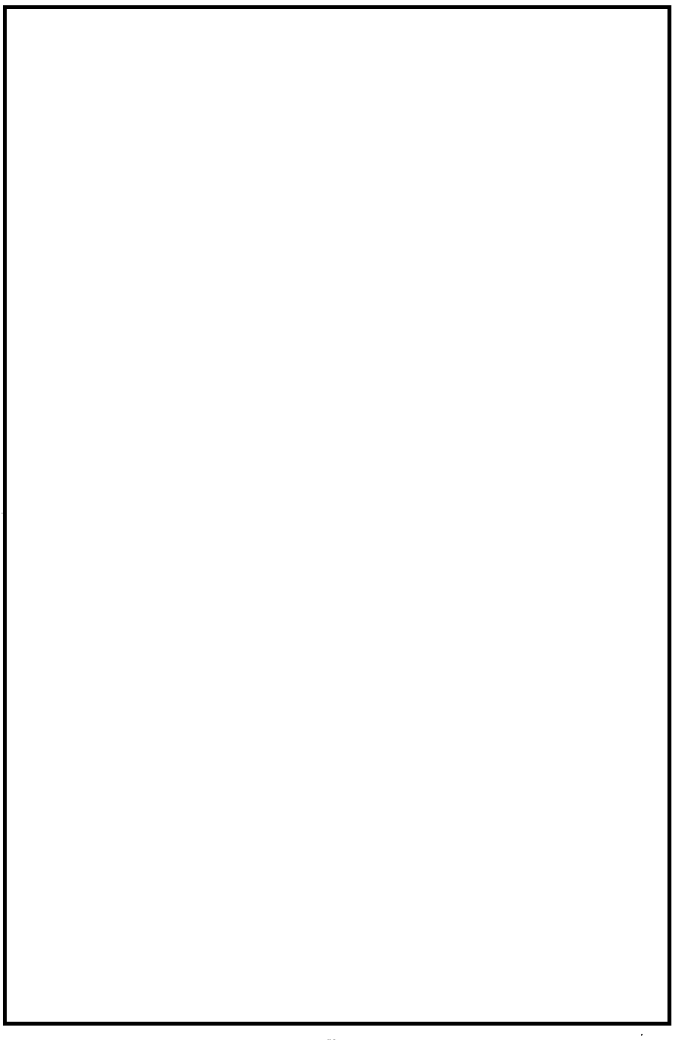

Fig. 1.