The Quest for Capable Cryptography:

Horst Feistel's Groundwork for the Art of Encryption

Levi Seibert

00086809

Lseiber1@my.athens.edu

CS 484 Research Paper

The Quest for Capable Cryptography: Horst Feistel's Groundwork for the Art of Encryption

In a May 1973 edition of the *Scientific American* magazine, Horst Feistel submitted an article entitled "Cryptography and Computer Privacy" (Feistel, 1973 throughout). Although this manuscript is nearing its half-centennial, it presents cryptographical ideologies that are still applicable today. In fact, the focus of the paper was the idea of individual privacy, a concept that arguably is more important in today's society than ever before. Feistel predicted that it would "soon be feasible to compile dossiers in depth on an entire citizenry", meaning that one would be able to compile a complete profile on an individual, based on information that is in one place. This is indeed the case using today's World Wide Web, where personal data abounds, partially at the fault of the affected individuals and partially at the fault of those collecting the data. How can this privacy crisis be mitigated? Horst Feistel argues that "a computer system can be adapted to guard its contents from everyone but authorized individuals by enciphering the material in from highly resistant to cipher-breaking." Feistel then explains different variations of encryption algorithms and determines how successful and useful they are at providing data confidentiality.

Horst Feistel was a cryptographic trailblazer. His work in creating encryption algorithms paved the way for several of today's modern cryptographic methodologies, like DES. The Lucifer cipher, which he briefly explains in "Cryptography and Computer Privacy," although now considered unsafe and immature, was revolutionary when he released it. His objective was to properly implement Claude Shannon's concepts of diffusion and confusion in an effective cipher that could be decrypted only with the knowledge of the private key. These ideas led to Feistel's focus on using substitution and permutation boxes.

THE QUEST FOR CAPABLE CRYPTOGRAPHY

In “Cryptography and Computer Privacy,” Horst Feistel begins by looking at cryptography from the bare minimum: simple, single alphabetic substitution. As he demonstrates, while messages enciphered using this methodology would add some confusion, it does nothing to deal with letter frequencies. Anyone could look at a sample of ciphertext and observe which letters appear most frequently. Based on their knowledge of the frequency of letters appearing in the English language (or any language that is being dealt with), they could easily start matching ciphertext characters to plaintext letters.

Feistel then shifts focus to discuss encryption using binary digits rather than English characters. He begins this section with an oversimplification of adding a key string consisting of bits from Key 0 and Key 1. Key 0 bits are marked with a 0 and Key 1 bits are marked with a 1. Key 0 keeps the bit as they are, and Key 1 flips the value of the bit. Some sequence of binary 0s and 1s makes up the actual key and the key is then added to the message to encrypt it. This process actually just performs binary addition modulo 2, which is simply just the XOR operation. For example, if the plaintext message was the bit sequence 01011101 and the selected key was 11000011, then each bit from the two sets is added together (and the modulus 2 is taken). So, adding the two example sequences, the result is 10011110. This algorithm, simple as it is, actually can be impenetrable. As long as the key bits are chosen completely randomly, there is no real way of cracking this cipher. The problem with this algorithm is that it often is unfeasible. In order for it to work, both the sender and the receiver must have the exact same key and this key must be the same length as the message (the keys are loaded on to two identical tapes that hold the exact same key). This leads to so much overhead, that although the enciphering is excellent, is not reasonable for the average user. That being said, because of the strength of the cipher, this is the basis for the Vernam system, also called the “one-time pad,” used by

government agencies in several countries. Its incredible strength does make it a viable choice for delivering top-secret messages, as long as the hardware is able to support it.

Since the bit-for-bit randomized key methodology is not a valid solution in most situations, Feistel offers a solution that would enable the end-users to use two (much shorter) tapes to produce the keys. The two tapes have to be either of prime or co-prime length. This restriction ensures there are no matching common factors, so when the two lengths are multiplied, there are no repeated digits. This allows two tapes to be used to create a unique set of key digits that do not repeat for the product of the lengths of the two tapes' digits. Although still unrealistic for common users, this does make the implementation of the algorithm more feasible. In addition, Feistel points out a major flaw in Vernam-style ciphers: there is not any avalanche effect. That is, a single bit difference in the enciphering only changes one bit of the final ciphertext (or plaintext). One might think that this would be useful to the cryptographer; a simple mistake will not ruin the message. However, this is a flaw and can lead to easier discovery of the plaintext. As will be explained below, one key feature of strong encryption is diffusion, which is built upon the goal of promoting avalanches. Therefore, a lack of avalanche effect leads to a weaker cipher.

Feistel goes on to explain that this flaw is best mitigated not by modifying the stream algorithm itself, but instead by looking at encryption in a new way. Rather than encrypting each bit of data in sequence, the best method of cryptography considers blocks of message bits. Block ciphers, which convert *n* plaintext bits into *n* ciphertext bits, can be very effective, and can be created in a way in order to increase the effects of any changes in the text or key (a successful avalanche effect). The first way Feistel presents block encryption is through use of simple substitution and permutation boxes. Substitution boxes take a given length input and produce a

predefined output. This output is of the same length as the input; however, the number of bits that are set is not one-to-one: an input consisting of three 1s might produce an output of one 1. One beauty of this system is that it is actually, as Feistel puts it, "nonsystematic", that is, one connection in a substitution box gives no information regarding any other connections. However, there is an issue regarding frequency analysis with substitution boxes. With a fair amount of observation, an attacker could observe patterns among blocks of characters and may be able determine which substitution boxes were used. Substitution boxes can be quite large and handle many inputs, creating a broader range of possible outputs and thus discouraging frequency analysis attack. However, in order to be truly effective, there would have to be so many inputs that the number of internal terminals would be simply unfeasible for any normal machine. Therefore, although in a perfect world, substitution boxes could produce the perfect answer to the encryption problem, in the real world, it cannot do so.

Permutation boxes are similar to substitution boxes in that they accept input of a specific length and produce a given output based on that input using terminal switches. However, they differ in that permutation boxes only shuffles the inputted bits. Each binary input is wired to a specific binary output. This is a linear system and can be solved without much difficulty. All an attacker must do is input a single bit into the box and observe its result. This process can be repeated multiple times (if the there are n inputs, then the key can be discovered in $n - 1$ attempts). Obviously, this method of enciphering is insufficient.

If permutations, substitutions, and bit-by-bit stream ciphers do not work, is there any hope for secure encryption? Thanks to the brilliant minds of men like of Claude Shannon, the cryptography culture turned to the idea of "mixing transformations" or creating product ciphers. These ciphers consist of two or more ciphers that are combined in such a way that they are

stronger than either of the ciphers on their own. The goal of such transformations is to create an encryption algorithm that emulates a sufficient substation cipher without the need for unattainable technologies. The balance of these two ideas is built upon Shannon's ideas of infusing "confusion" and "diffusion." The binary message is confused using a (relatively small) number of substitution boxes and diffused using permutation boxes. Merging these two methodologies into one encryption cipher may result in the solution to the cryptography problem.

Feistel, as he illustrates in the article, developed a rudimentary system that provided uncommon cryptographic strength while still being realistic and attainable. The Lucifer cipher, which he worked with IBM to create, makes use of substitution boxes that take inputs of the length of four bits. For the permutation boxes, the system uses inputs of larger lengths (either 64 or 128 bits). The encryption process works by alternating substitutions and permutations for a given number of rounds. As illustrated in the magazine article, this system obfuscates the message (in his illustration, an input of single 1 and fourteen 0s results in eleven 1s and four 0s, using six fourteen-bit permutation boxes and five three-bit substitution boxes). In the actual implementation, the message is encoded even more. Since the S boxes are nonlinear and can increase the number of 1s in the message, a random avalanche of 1s may pervade the system (which is an integral factor in proper cryptography).

Although the Lucifer cipher is an excellent step in the right direction for encryption, it is far from perfect. Feistel himself goes on to add a few suggestions to the system in order to make it more secure. For example, he suggests that the substitution and permutations boxes be made permanent and that the substitution boxes have two potential outputs, with the actual result being decided by a binary key. At the time of his writing in 1973, this method made the Lucifer cipher

impenetrable by analysis. In today's world, this is no longer true and the cipher is considered immature and inadequate.

For its time, however, the Lucifer cipher was a masterpiece and provided cryptographic strength without unfeasible requirements. In fact, Feistel concludes the article by illustrating how his cipher could be used to ensure the authenticity of data messages. Using the system the combination with two synchronized password generators, a system could allow two users to securely send data to one another while ensuring the data had not been tampered with.

Horst Feistel, as illustrated in his 1973 *Scientific American* article, "Cryptography and Computer Privacy" was ahead of his time in creating useful encryption systems. The article concludes with his well-known Lucifer cipher; however, the age of encryption hardly concluded with that cipher. In actuality, the Lucifer algorithm really is not that strong of a cipher (although it definitely was for the time). The concept behind it, though, lives on. The concept of the block cipher is the basis of several modern-day symmetric encryption schemes, some still used today (like AES) and others since deprecated (like DES, which Horst Feistel helped create). While most of Feistel's work is now considered ancient and archaic, he paved the path for the field of encryption and charted a course for the quest of capable cryptography.

References

Feistel, H. (1973). Cryptography and Computer Privacy. *Scientific American*, *228*(5), 15–23. https://www.jstor.org/stable/24923044