FROM CHAOS TO CRYPTOGRAPHY

K.M. ROSKIN AND J.B. CASPER

ABSTRACT. A new and unexplored area in the field of cryptography is the use of chaotic systems to encrypt data. The purpose of this paper is to present how the principles of chaos theory can applied to the field of cryptography. A cipher that uses chaos dynamics is presented that has very good statistical properties.

1. INTRODUCTION

One of the characteristics of chaotic dynamics is a sensitivity to initial conditions; i.e., two relatively close initial values will diverge as the system evolves. We use a chaotic function that is initialized with a key to encrypt data. We hope that the chaotic behavior is too difficult to predict by analytical methods without the secret key being known. This would reduce a potential attack to one category, that of a brute force attack, in which all possible keys are tested against the encrypted data. Brute force attacks are rarely successful because any attempt to crack the key depends directly upon how long the key is; an n-bit key can be any of 2^n possibilities. In our cipher we use a 256-bit key, which provides so much protection that Bruce Schneier, a noted cryptologists, estimates it would take the energy output of most of the suns (the stars, not the computers) in our universe to power a computer to just count that high, let alone make a brute force attempt at decryption [Sch96].

2. CHAOS THEORY AS IT APPLIES TO CRYPTOLOGY

2.1. **Sensitive Dependence in Chaotic Algorithms.** One of the fundamental principles of chaotic functions is sensitive dependence, or sensitivity to initial conditions. A small difference in the starting values of the function will, after the function is iterated many times, lead to a great divergence in the produced behavior. For example, if a chaotic equation is started once at 52 and another time at 52.001, after 1000 iterations the value of the first equation might be 45, while the value of the second equation would be 160.934.

2.2. **Sensitive Dependence in Cryptography.** Sensitive dependence is a very valuable property for cryptographic algorithms because one of the desired features of a cryptographic algorithm is that if the initial conditions used to encrypt data are changed by just a small amount, one bit for instance, the encrypted text should be wildly different. This sensitivity ensures that if an opponent (a person who tries to crack cryptographic systems is commonly called an opponent) tries to decrypt the data by trying different initial conditions and looking for patterns, it won't work. Even if the initial conditions that the opponent tries are very close to the ones used to encrypt the data, the opponent will still get gibberish as output.

The research group is deeply indebted to Prof. Emeritus Ralph Abraham, of the University of California at Santa Cruz, for sharing with us the wonders of chaos theory that led to this paper.

2.3. **The Logistic Map.** The chaotic function that we use is the well known logistic map:

$$x_{n+1} = rx_n(1 - x_n)$$

When $r = 3.9$, the logistic map exhibits chaotic behavior, and hence the property of sensitive dependence. The reasons that we selected the logistic map instead of another chaotic algorithm for our project are simple. The map is one dimensional, which is good because we want scalars to do the encryption, and the chaotic properties of the logistic map are well known [Str94].

3. The Basic Principles of Cryptographic Algorithms

A cipher is another name for a cryptographic algorithm. The purpose of a cipher is to take unencrypted data, called the *plaintext*, and produce an encrypted version of it, called the *ciphertext*. There are two classes of ciphers: block ciphers and stream ciphers.

3.1. **Stream Ciphers.** A stream cipher encrypts one bit of the plaintext at a time. The simplest example of a stream cipher is where a bit of data is merged, one bit at a time, with another block of data, called the pad. This requires that the pad be as large as the plaintext. Most stream ciphers use a secret key or password to generate the pad bits as they are needed. In the One Time Pad, the pad is the key; this has some unique properties, for more information see [Sch96]. To decrypt, generate the same pad and do the inverse of the merge operation on the ciphertext. Stream ciphers work very well for real time data such as voice and video, where only small pieces of data are known at a time.

3.2. **Block Ciphers.** Block ciphers operate on blocks of plaintext that are large than one bit. The difference is largely artificial since a stream cipher can be viewed as a block cipher with a block size of 1-bit.

The security of an algorithm should entirely on the secrecy of the key. If you rely on the secrecy of a algorithm, you are either a master cryptologists or a fool. We don't clam to be either, so we use a key and submit our algorithm to peer review.

3.3. **Cryptographic Keys.** Keys increase the degree of security because well known, off the shelf, time tested algorithms can be used. An *encryption pair* is a key and the encryption system that the key is used with. Thus in an encryption pair, only the key has to be secret. Every encryption pair can be thought of as different key-less encryption algorithm.

Much of the cryptography research centers around showing that the encryption pair of a given system cannot be reconstructed with having the key. In other word, the security of the the encryption system rests only in the key. If this is true (something that no one has been able to prove for any encryption system except for the One Time Pad, mentioned above), then the only way to attack the system is by brute force.

The best way to defend against a brute force attack is to use a large key. The problem is that large keys are often too large to use in the cipher directly. The solution is to break the key down into sub-keys called *session keys*. Session keys are small enough to be used in the cipher. The system we use is to divide the 256-bit key into 32 subkeys of 8 bits each.

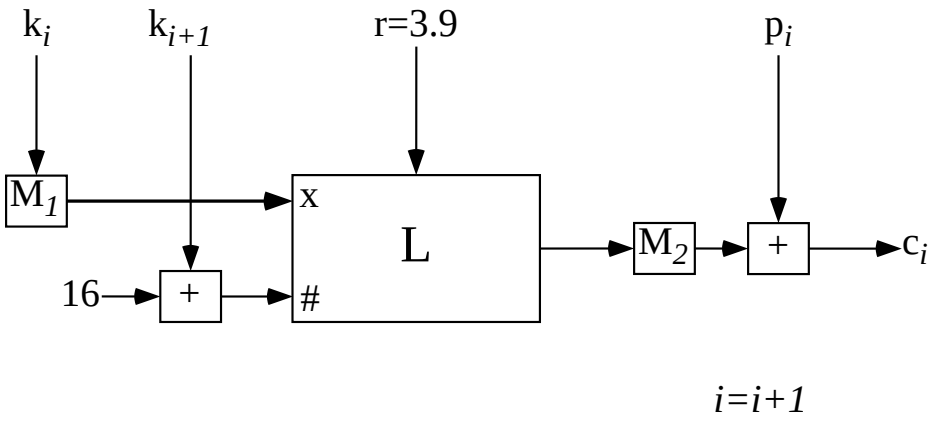

FIGURE 1. Diagram of the Simple Cipher

4. Applying Principles of Chaos to Cryptology

4.1. **The Simple Cipher.** We chose to call our first encryption algorithm the "Simple Cipher."[1] The Simple Cipher is a simple block cipher with a 256-bit key. The key is used to generate a pad that is then merged with the plaintext a byte at a time, in other words the block size is 8-bit. The specifics of how we generate the pad are a little more complex, however. As diagrammed in Figure 1, we use two successive session keys, k_i and k_{i+1}. Instead of adding them directly to the plaintext we instead use them as initial conditions in a chaotic map.

The box M_1 represents a mapping from the session key space, all integers between 0 and 255, into the domain of the logistic map, all reals in the interval $[0, 1]$. The sum of the next session key and the number 16 is used as the number of iterations of the logistic map. M_2 uses fuzzy set membership to map the domain of the logistic map, $[0, 1)$, back into the interval $[0, 255]$. As we encrypt each new block, i, the counter used to keep track of the current session key, is incremented. The output of the logistic map is then merged with the plaintext to give the ciphertext. Decryption is very simple, the same pad is generated but this time un-merged with the ciphertext to retrieve the plaintext.

4.2. **Analysis of the Simple Cipher.** An analysis of the Simple Cipher found that it has some nice statistical properties. Although the algorithm looked good statistically, there is more to the story. For more insight into how well the algorithm worked, we tried a visual comparison of the plaintext and the corresponding ciphertext. This comparison was made easy because we elected to encrypt pictures with the algorithm, an idea we borrowed from Fridrich [Fri97]. The terms plaintext and ciphertext no longer seemed quite applicable, so we coined the terms *plainimage* and *cipherimage* to refer to the unencrypted and the encrypted image data respectively. When we ran the cipher on the raw data of an image of the Eiffel Tower, as shown in Figure 2, the Simple Cipher produced the cipherimage shown in Figure 3.

[1] All the ciphers presented in this paper are available as C++ source code online at *http://xcrypt.theory.org*.

FIGURE 2. The plainimage.

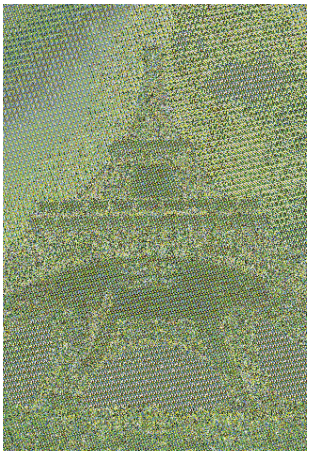

FIGURE 3. Encrypted with the Simple Cipher.

As can be seen from Figure 3, a lot of information about the plainimage is easily visible in the cipherimage, as evidenced by the ghost of the plainimage that is still visible in the cipherimage. The problem is that the pad produced by the cipher depends only on the key and therefore has a period equal to the key size. This means that the same series of pads is repeatedly used and a pattern of displacement is created. Patterns are easy for the eye to distinguish, so even though the colors in the image change, it is still possible to see what the picture is. This can be seen more clearly, if we look at the changes in the cipherimage if we change one bit in the key. In 4, the black pixels represent a change at that location in the cipherimage caused by the bit change in the key. The *difference map* clearly demonstrates the periodicity of the cipher.

4.3. **The Advanced Cipher.** Periodic behavior in a cipher is a clear sign that the cipher is not secure. To solve this problem we incorporated a feedback mechanism into the cipher, as seen in Figure 5. Feedback is a very common principle in

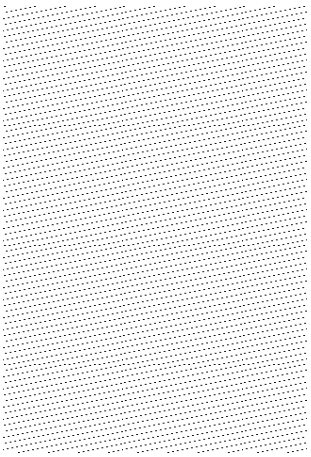

FIGURE 4. Difference map of an image encrypted with keys that differs by one bit using the Simple Cipher.

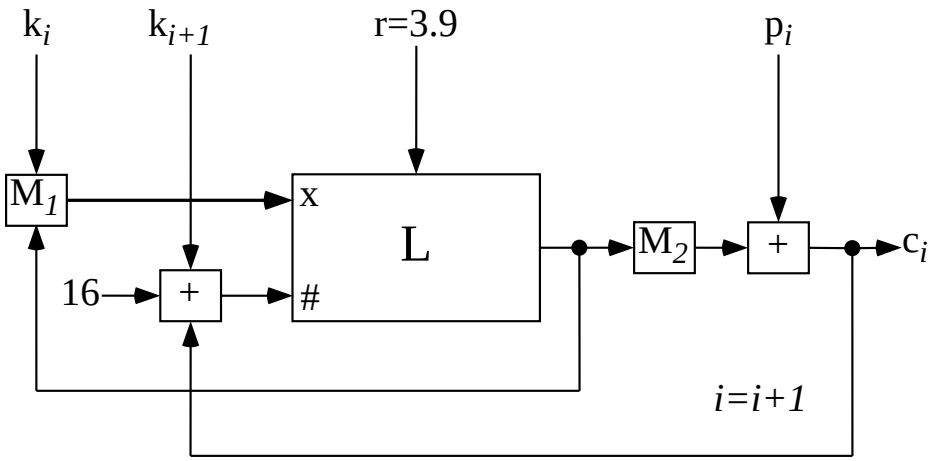

FIGURE 5. Diagram of the Advanced Cipher

cryptographic algorithms. The basic concept is that the encryption of each part of the plaintext depends not only on the key, but also on the previous ciphertext. In Figure 5 the algorithm begins exactly like the Simple Cipher, but very quickly diverges. To encrypt the next block of the plaintext, the output of the chaotic function from the previous encryption is added to get the new x. Furthermore, the numerical value of the previous block of ciphertext is added to the number of iterations. This use of feedback and the corresponding increase in complexity is what led to our renaming of the algorithm. Instead of the Simple Cipher, we now use the auspicious title of Advanced Cipher.

4.4. **Analysis of the Advanced Cipher.** The Advanced Cipher does a much better job of encrypting the image, which can easily be seen in Figure 6. The ghost of the Eiffel Tower, seen after the simple encryption, is all but gone after the advanced encryption. There are still some residual dark spots that appear around

FIGURE 6. Encrypted with the Advanced Cipher.

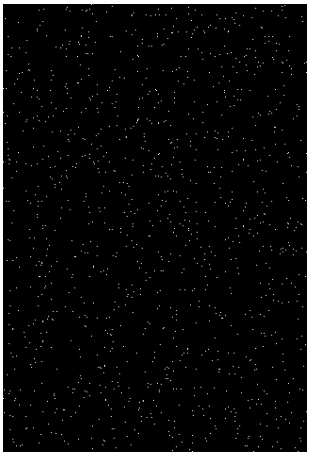

FIGURE 7. Difference map of an image encrypted with keys that differs in one bit using the Advanced Cipher.

high contrast area of the plainimage, but they give away much less information about the structure of the plainimage. Furthermore, these dark spots can be hidden by compressing the plainimage, as any encryption system that incorporates the cipher would do instead of just applying the cipher directly as we have done.

The reason that the advanced chaos cipher does a much better job of encryption is that it does not suffer from the same periodicity as the Simple Cipher. This is the direct result of using feedback in the algorithm; there can be no simple periodicity because the series of pads used depends on the cipherimage itself instead of solely upon the key. This can be seen in Figure 7, the same kind of difference map mentioned above but using the Advanced Cipher. As can be seen, a one bit change in the key causes the cipherimage to diverge very quickly.

Another desirable effect of the feedback it to ensure that any changes in the plainimage are cascaded forward throughout the cipherimage, which means that

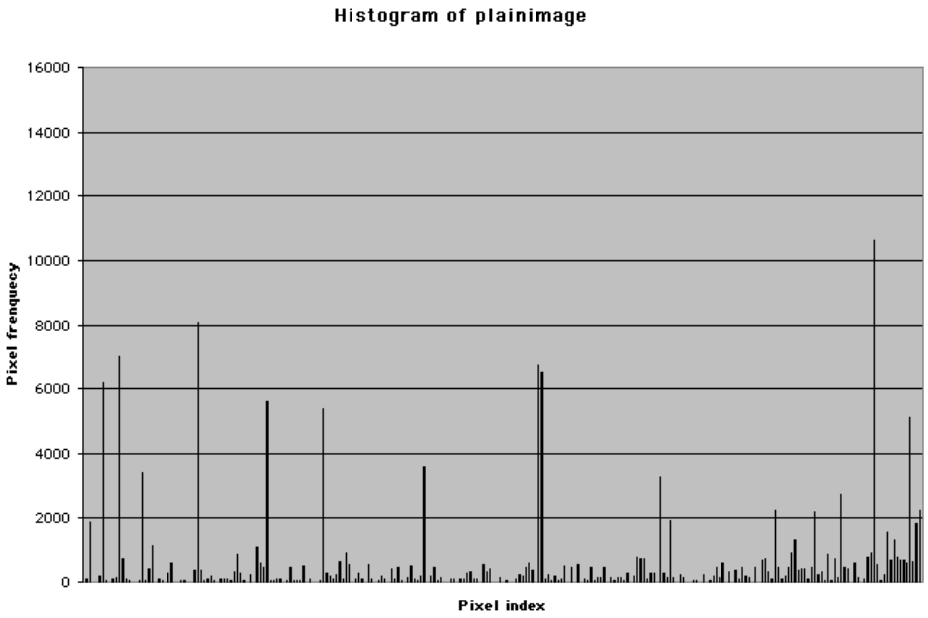

FIGURE 8. Histogram of the plainimage.

two almost identical images will encrypt to completely different cipherimages. This sensitivity to the plaintext is a good sign for the security of the algorithm, because it gives an opponent less information to work with.

To prevent the leakage of information to an opponent, it is also advantageous if the cipherimage bears little or no statistical similarity to the plainimage. To analyze the statistical distribution, we created a histogram of the plainimage, which is shown in Figure 8. Large spikes can be seen around the black and blue colors, corresponding to the high amount of those colors in the picture. The histogram of the cipherimage, Figure 9, on the other hand, is much more uniform and bears little statistical resemblance to the plainimage. Note that the scale of the cipherimage histogram is much larger than the scale of the plainimage histogram. If the two images were shown to scale, the cipherimage histogram would be no more than a low level of noise compared to the peaks of the plainimage histogram.

Another nice statistical property is that the advanced cipher is highly sensitive to the key used. By statistical analysis, we found that changing one bit in the encryption key caused 49.6% of the bits to change in the corresponding cipherimages. Optimally, fully half of the bits should change, in order to hide any information about the key from leaking. It is clear that the Advanced Cipher is almost optimal in this respect.

5. CONCLUSION

We have demonstrated that the Advanced Cipher has some good fundamental properties for cryptography, but we have not yet actually tried to crack our algorithm. This is a very critical part of our research, particularly in light of the fact

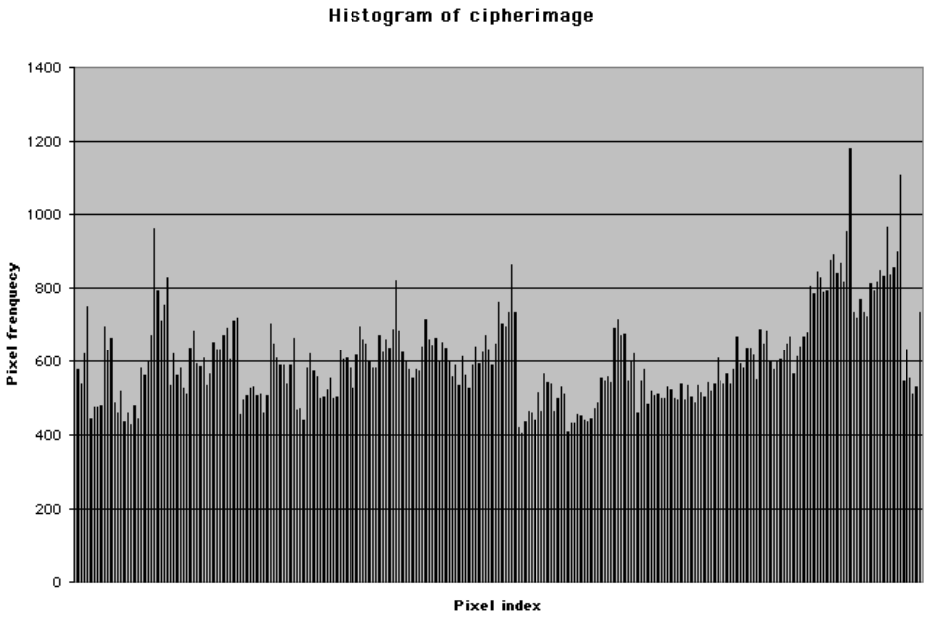

FIGURE 9. Histogram of the cipherimage, encrypted with the Advanced Cipher.

that our original encryption algorithm didn't work very well. We need to make sure that when we introduced feedback, we didn't just hid any cryptographic weakness from ourselves. Another area of potential exploration is the use of other algorithm concepts, like the permutation concept introduced in Fridrich's article [Fri97] and the time targeting [GDK+96, Hei99]. And there is still much work to be done on connecting chaos and cryptology [KJSP98, Fri98].

And there is still much more to be done with the analysis of the Advanced Cipher. Only further research can tell if it deserves its title.

REFERENCES

[Bap98] M.S. Baptista, *Cryptography with chaos*, Physics Letters A **240** (1998), no. 1-2, 50–4.

[DKS97] F. Dachselt, K. Kelber, and W. Schwarz, *Chaotic coding and cryptoanalysis*, Proceedings of 1997 IEEE International Symposium on Circuits and Systems. Circuits and Systems in the Information Age (New York, NY, USA), vol. 2, IEEE, IEEE, June 1997, pp. 1061–4.

[DKSV98] F. Dachselt, K. Kelber, W. Schwarz, and J. Vandewalle, *Chaotic versus classical stream ciphers—a comparative study*, Proceedings of the 1998 IEEE International Symposium on Circuits and Systems (New York, NY, USA), vol. 4, IEEE, IEEE, May 1998, pp. 518–21.

[Fri97] J. Fridrich, *Image encryption based on chaotic maps*, 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation (New York, NY, USA), vol. 2, IEEE, IEEE, October 1997, pp. 1105–10.

[Fri98] J. Fridrich, *Symmetric ciphers based on two-dimensional chaotic maps*, International Journal of Bifurcation and Chaos in Applied Sciences and Engineering **8** (1998), no. 6, 1259–84.

[GDK+96] D. Gligoroski, D. Dimovski, L. Kocarev, V. Urumov, and L.O. Chua, *A method for encoding messages by time targeting of the trajectories of chaotic systems*, International Journal of Bifurcation and Chaos in Applied Sciences and Engineering **6** (1996), no. 11, 2119–25.

[GG96] Jonathan B. Gallagher and Jeremy Goldstein, *Sensitive dependence cryptography*, Web page: `http://www.navigo.com/sdc/`, 1996.

[Hei99] Tim Heilman, *An implementation of encryption by time targeting of the trajectories of the henon map*, Research project at the the University of California at Santa Cruz., March 1999.

[KJSP98] L. Kocarev, G. Jakimoski, T. Stojanovski, and U. Parlitz, *From chaotic maps to encryption schemes*, Proceedings of the 1998 IEEE International Symposium on Circuits and Systems (New York, NY, USA), vol. 4, IEEE, IEEE, June 1998, pp. 514–17.

[PS95] F. Pichler and J. Scharinger, *Finite dimensional generalized baker dynamical systems for cryptographic applications*, Computer Aided Systems Theory — EUROCAST '95. A Selection of Papers from the Fifth International Workshop on Computer Aided Systems Theory (Berlin, Germany) (F. Pichler, R. Moreno-Diaz, and R. Albrecht, eds.), Springer-Verlag, May 1995, pp. 465–76.

[Sch96] Bruce Schneier, *Applied Cryptography: Protocols, Algorithms, and Source in C*, second ed., John Wiley & Sons, Inc., New York, 1996.

[Str94] Steven Henry Strogatz, *Nonlinear dynamics and chaos: With applications to physics, biologym chemistry, and engineering*, first ed., Addison-Wesley Publishing Company, Reading, Massachusetts, 1994.

[TLBCM98] Yang Tao, Yang Lin-Bao, and Yang Chun-Mei, *Cryptanalyzing chaotic secure communications using return maps*, Physics Letters A **245** (1998), no. 6, 495–510.

[YWC97] Tao Yang, Chai Wah Wu, and L.O. Chua, *Cryptography based on chaotic systems*, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications **44** (1997), no. 5, 469–72.

[ZLY97] Hong Zhou, Xie-Tang Ling, and Jun Yu, *Secure communication via one-dimensional chaotic inverse systems*, Proceedings of 1997 IEEE International Symposium on Circuits and Systems. Circuits and Systems in the Information Age (New York, NY, USA), vol. 2, IEEE, IEEE, June 1997, pp. 1029–32.

E-mail address: `KRISH@CATS.UCSC.EDU`
E-mail address: `JCASPER@CATS.UCSC.EDU`