Chaos-Based Symmetric Key Cryptosystems

Christopher A. Wood
Department of Computer Science, Rochester Institute of Technology, Rochester, New York, USA

Abstract— *Chaos theory is the study of dynamical systems that are highly sensitive to initial conditions and exhibit seemingly random behavior. From the perspective of cryptography and information security, randomness generated from entirely deterministic systems is a very appealing property. As such, the application of chaos in modern cryptography has been a topic of much research and debate for over a decade.*

This paper presents an overview of chaotic dynamics and their role in symmetric key chaos-based cryptosystems from both a theoretical and practical perspective. It is argued that chaos-based ciphers are not likely to succeed until a valid and accepted definition of discrete chaos in finite domains is established, the inefficiencies of chaos-based cipher implementations are improved, and thorough security analysis reveals them to be comparable to standardized cryptographic primitives.

Keywords: Chaos Theory, Cryptography, Dynamical Systems, Symmetric Key Cryptosystems

1. Introduction

Ever since the discovery of chaotic behavior in the mathematical models of weather systems by Edward Lorenz in the early 1960s [1], chaos theory has found its way into many different fields of science, including physics, economics, biology, and even philosophy. In recent years its influence has begun to spread into cryptography. The sensitivity to initial conditions and seemingly random behavior produced from deterministic equations caught the eye of cryptographers as they tried to incorporate these properties into cryptographic primitives, including symmetric key ciphers, hash functions, and pseudorandom number generators.

Although chaos-based symmetric key cryptosystems are very appealing at a theoretical level, they don't provide the same cryptographic assurances that come with standardized cryptosystems like the Advanced Encryption Standard (AES). Based on research efforts throughout recent years, chaos-based symmetric key cryptosystems are not likely to succeed and thrive into the future unless the following conditions are met:

1) A valid and accepted definition of discrete chaos in finite domains is established, likely stemming from the discrete Lyapunov exponent
2) The inefficiencies of cipher implementations that are based on real-value chaotic maps are improved
3) Thorough security analysis reveals that chaos-based symmetric key cryptosystems are comparable to standardized cryptosystems from a diffusion and confusion perspective

This paper explores the history of chaos-based symmetric key cryptosystems and discusses the shortcomings of past and present research efforts. We attempt to explain the properties that attribute to their lack of success and acceptance by commercial applications.

2. Chaos Dynamics

Chaos is best known as a sensitivity to initial conditions exhibited by dynamical systems described by differential equations or iterated mappings. In the case of chaos-based symmetric key cryptosystems, we will focus on chaotic systems that are defined by iterated mappings as they are the more likely candidates for actual implementation. It is important to note that these iterated mappings are really just recurrence equations derived from their differential equation counterparts.

For such systems, a sequence of points created by recursive iterations $f^n(x)$ of some initial value x_0 of the phase space, which is the domain of the map, is defined as a phase trajectory, or simply a trajectory. These systems must be sensitive to initial conditions, have a dense collection of points with periodic orbits, and be topologically mixing in order to be deemed chaotic [2].

Periodic orbits are recurring sequences of elements in trajectories produced by chaotic maps. In dynamical systems, a collection of points is dense if at any point $x \in X$, where X is the phase space, x either belongs to a subset $A \subseteq X$ or is a limit point of A. One can see that if such a collection of points exists then all possible values in the phase space will be generated arbitrarily closely.

Topological mixing is a form of mixing that may be defined without appeal to a measure (or size) of the system. Formally, a system F possesses the mixing property if, for any two measurable sets A and B, there exists an integer N such that for all $n > N$ the relationship in (1) is satisfied [3].

$$f^n(A) \cap B \neq \emptyset \tag{1}$$

Sensitivity to initial conditions is formally defined as a characteristic of dynamical systems where two significantly close points will rapidly diverge under f^n to produce very different trajectories as they are iterated by the map. This

characteristic is satisfied if the chaotic system has a positive Lyapunov exponent, which gauges the rate of separation of infinitesimally close initial trajectories. Specifically, two trajectories in a system's phase space with initial separation δX_0 diverge according to:

$$|\delta X(t)| \approx e^{\lambda t} |\delta X_0| \tag{2}$$

$$\lambda = \lim_{t\to\infty, |\delta X_0|\to 0} \left(\frac{1}{t} ln \frac{|\delta X(X_0, t)|}{|\delta X_0|} \right) \tag{3}$$

In other words, the difference between two initial trajectories will exponentially increase after a very short time depending on the magnitude of the Lyapunov exponent λ. Based on this rate of separation, a system is deemed chaotic if $\lambda > 0$. Conversely, the system is deemed "regular" if $\lambda \leq 0$.

One final important piece of chaotic dynamics is the notion of attractors and robust chaos. A chaotic attractor is a set of elements in the phase space towards which trajectories of the system evolve over time. The elements of trajectories produced by the system will remain in the bounds of the attractor as they are recursively generated. Furthermore, two arbitrarily close trajectories within an attractor will exhibit different and unrelated behavior within the bounds of the attractor as they are recursively iterated over time.

Robust chaotic systems are those that have an attractor even when parameters in the system undergo small changes. This is an ideal property for chaotic systems that are used for cryptography because any change in initial conditions will cause trajectories to remain within the same attractor, thus making it difficult to predict any outcome without knowing the initial conditions of the system and the iteration count.

3. Chaos Theory and Cryptography

At a theoretical level, chaotic systems have unique characteristics that have potential applications in cryptography. These characteristics can be related and subsequently mapped to properties of cryptographic primitives. For example, consider the mixing property of chaotic systems. By definition, chaotic maps with strong mixing properties will recursively generate regions of elements that will eventually cover the majority of the phase space and start to overlap as the system evolves over time. Cryptographic primitives possess a similar property known as diffusion, which is defined as the process by which the influence of a single plaintext digit is spread out over many ciphertext digits.

Another similarity between chaotic systems and cryptographic primitives lies in the relationship between discrete time-based system iterations and encryption rounds. Consider the following discrete time-based system:

$$x_{n+1} = r x_n (1 - (x_n)) \tag{4}$$

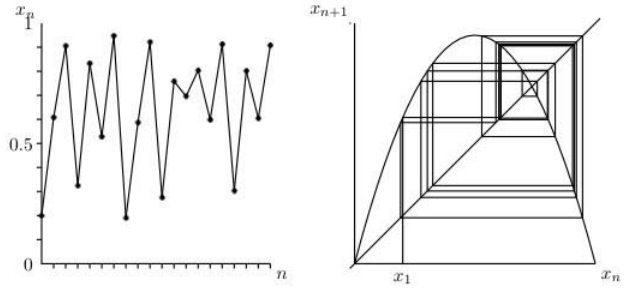

Fig. 1: The figure on the left shows a single trajectory of the logistic map as it is recursively iterated n times. The figure on the right shows a plot of the points (x_n, x_{n+1}) for this same trajectory, which in turn depicts the chaotic attractor of the logistic map.

This is the recurrence relationship for the logistic map, which is derived from the differential form of the logistic equation ($\frac{dx}{dt} = rx(1-x)$). The value r is a positive constant that is commonly referred to as the control parameter [4]. At first glance it might seem that this recursive map is very simple. However, after further analysis one can see that it is capable of very complicated behavior depending on the parameters of the system. Figure (1) shows one chaotic trajectory of the logistic map.

Each iteration of the logistic map produces new points in the phase space. Depending on the level of mixing within the chaotic map, this value will diverge and cover more elements in the phase space. These iterations are very similar to rounds in a cryptographic cipher, where each round serves to transform the internal state of the cipher towards the final ciphertext.

The last most significant similarity between chaotic systems and cryptographic primitives is the relationship that exists between system parameters and cryptographic keys. In a general sense, the system parameters for a chaotic map and a cryptographic key serve the same purpose, which is to determine the functional output of the system or cipher. Since chaotic maps and cryptographic ciphers are both deterministic, the system parameters and keys determine exactly what the output will be in such a way that it is statistically infeasible for an attacker to guess the output without knowledge of such values. It is typical in chaos-based ciphers for both the initial conditions and the iteration count to remain secret in order to maintain the security properties and pseudorandomness of the map.

Although the properties of chaotic maps and cryptographic primitives have similar characteristics that make them appealing to work together, there is one significant property of chaos to take into account when considering its application in symmetric key cryptosystems. Encryption schemes traditionally operate on finite sets of integers, whereas the chaotic principles discussed above usually occur on a (sub)set of real

numbers as chaotic behavior only truly exists in continuous domains. Therefore, the level of chaotic behavior exhibited by systems that operate on real numbers is closely tied to the amount of precision with which those numbers are represented.

4. Chaos-Based Cipher Design

Modern symmetric key ciphers usually consist of the four operations that make up AES: key addition, S-box substitution, permutation, and linear mixing. Chaotic ciphers usually attempt to replicate these four elements using space-discretized versions of chaotic maps that approximate real-valued systems. The term discretized is very important here. Chaotic systems work in two dimensions: time and space. For example, the logistic map is discretized in the time dimension (i.e. values occur at fixed points in time or at fixed iteration counts). The phase space is still based on the field of real numbers, meaning that it is continuous in the space dimension.

One reason that these maps are space-discretized is that chaotic behavior is normally observed over the set of real numbers. However, even in real intervals such as $[0, 1]$, the number of elements $x \in R$ is uncountable. This implies that it is impossible to represent all of the values within this interval and, by induction, any interval on a continuum. Therefore, given the finite representation of modern computing devices we must limit the precision with which we represent elements in the phase space.

Another reason for further space-discretization of chaotic maps is for performance. By limiting the amount of precision for phase space elements for chaotic maps the amount of data that has to be stored and computed is reduced.

The approximation of a chaotic map is important in the design of chaos-based ciphers. Given the computational and memory limitations of modern computing systems there should be an efficient mapping scheme between elements on a continuum to elements of a finite set when discretizing chaos-based ciphers. The most common approach has been to partition the phase space of a continuum into a finite number of blocks [5]. The size of such blocks correlates to the degree of precision available for implementation. The more accurate the representation can be, the smaller such partition blocks can become, which in turns increases the size of the system and its phase space.

The confusion and diffusion properties of these chaotic maps must also be considered in terms of both the Euclidean geometry and Hamming distances [6]. This requirement goes back to the Lyapunov exponent measure of chaotic maps that gauges the rate of separation for trajectory elements. Approximated chaos for discrete systems must have a high rate of separation and input/output differences. Otherwise, information about the contents of the system parameters may be leaked if patterns begin to emerge through frequent periodic behavior exhibited by the map.

5. Cipher Evaluation

Security, from a cryptography perspective, is measured from both the theoretical and practical levels of cryptographic primitives. At the theoretical level, cryptographic primitives are deemed secure if they possess "randomness increasing" and "computationally unpredictable" characteristics. A complete description of these properties is beyond the scope of this paper, but a brief discussion is warranted to make this paper self inclusive.

By definition, randomness increasing implies that the cryptographic primitive must increase the entropy of the system over which it operates. However, it is impossible in classical information theory for a deterministic function or map applied to a probability distribution P to increase entropy [5]. In practical implementations, however, where computational power and resources are limited, an increase in entropy may be possible. The reason for this is that given a mapping $G : S_1 \rightarrow S_2$ (S_i are finite sets) that is applied to P, where P is a PDF for each set S_i, the result $G(P)$ may be similar enough to approximate another distribution Q. Due to the limits of modern computing power it may be infeasible to differentiate Q from P, and if the entropy of Q is greater than that of P, then we can say that the mapping G is computationally randomness increasing. This loophole is exploited during the construction of modern chaos-based ciphers so as to hinder the application of cryptanalysis techniques to break the primitive.

One way to increase the entropy of chaos-based ciphers is to modify the order of the key and plaintext/ciphertext space. In such systems the size of these sets is directly proportional to the amount of entropy. Specifically, the entropy of a system with a key space of K keys is approximately $log_2 K$. Clearly, as the order of the key space increases, then the entropy increases as well.

The initial conditions of a chaotic system also play a significant role in its entropy. Consider, for example, the bifurcations of the logistic map. As the value of r is varied the number of unpredictable trajectories of a given initial value x_0 changes dramatically [4]. It is important to note that these initial parameters must be chosen such that the map both exhibits chaotic behavior. Furthermore, these should be chosen such that they have secure properties that allow it to avoid predictability and improve the pseudorandomness of trajectories.

The notion of being computationally unpredictable is a bit more sophisticated. Its roots lie in complexity theory, and the reader is referred to [5] for a more detailed discussion.

From a practical perspective, cryptographic primitives are deemed secure if they are resistant to known attacks. The two most common forms of cryptanalysis attacks are differential and linear cryptanalysis. Other forms of attacks specific to chaos-based ciphers include trajectory-based, loss of information, and memory attacks.

The probability of a successful differential or linear cryptanalysis attack depends on the statistical attributes of the cipher, or in this case, the internal chaotic map. If it is easy to predict values of the map after any iteration then it is obvious that these attacks will be simple to implement. However, as with any chaotic map, it is computationally difficult to perform such accurate predictions.

For example, consider the logistic map (4). If we partition the phase space of the region of the attractor into M equal subsets and calculate the number of times a trajectory visits each subset m_i for a large number of initial values and iterations we obtain a probability distribution of the map. The number of visits associated with each m_i is the probability p_i of that space in the phase space. It has been shown that the probability distribution of truly chaotic systems has no dependence on the system's initial value [7], which implies that the probability measure is unchanged by the dynamics of the system (i.e. invariant probability measure). If we build the logistic map with initial parameters such that its corresponding Lyapunov exponent $\lambda = 4.0$ the probability distribution is given by equation (5). This is the ideal distribution for chaotic maps that are used in chaos-based cryptosystems, as the probability of each phase element occurring after an iteration of the map is the same as any other element. This property increases the difficulty of an effective differential or linear cryptanalysis attack.

$$P(X) = \frac{1}{\pi\sqrt{X(1-X)}} \tag{5}$$

The number of iterations of a chaotic map also impacts the security of chaos-based ciphers. Since chaotic maps are deterministic, the final value can be easily computed given the initial conditions. However, by making the number of iterations for the map unknown, determining the initial conditions based solely on the output trajectory element becomes more difficult.

One must also consider the size of the blocks of data encrypted and decrypted by chaos-based cryptosystems. Larger data blocks means the attacker will have a harder time sifting through the data to find patterns and correlations. However, this improved security comes at the cost of performance, especially when considering chaos-based cryptosystems. Given the complexity of floating point operations on traditional processors and the requirements for the cipher, it might not be feasible to support larger data blocks.

This leads to another aspect of chaotic maps to consider when implementing a chaos-based cipher: the set of elements in the phase space. A direct translation of chaotic systems over the set of real numbers to a running cipher results in the use of high precision floating point operations. This results in very inefficient code. In addition, different processor architectures might handle floating point operations differently depending on their capabilities, which makes them susceptible to reproducibility problems.

6. Case Studies

Many different chaos-based ciphers have been designed and proposed in recent years. This section is devoted to four of those cipher designs. Namely, the Simple and Advanced ciphers, Chaotic Feistel cipher, and Rabbit cipher. The internals for each of these cipher designs are discussed along with their relative security properties.

6.1 The Simple and Advanced Ciphers

The Simple and Advanced ciphers, proposed by Roskin and Casper [8], are two very basic applications of chaotic maps in block ciphers. They are based on the unpredictability of the logistic map (4). The general idea is to encrypt bytes of plaintext as the final trajectory elements obtained by a variable number of iterations of the logistic map. In this application, both the initial value and the number of iterations of the chaotic map vary.

The complete Simple cipher algorithm is outlined as Algorithm (1). f is the logistic map (4) with initial parameter $r = 3.9$ that is used to generate trajectories of some initial value x_0. M_1 is a mapping function between elements in the key space to the domain of elements in the logistic map (namely, the real interval $[0, 1)$). Similarly, M_2 is the inverse of M_1 in that it maps elements in the domain of f to the set of integers between 0 and 255.

Algorithm 1 Simple cipher encryption

Generate the key schedule $\{k_0, k_1, k_2, ..., k_n\}$ from the 256-bit secret key K
for $i = 0$ to $n - 1$, where $|P| = n - 1$ **do**
  $x_0 \leftarrow M_1(k_i)$
  $t \leftarrow k_{i+1} + 16$, where t is the number of iterations
  $x_t \leftarrow f^t(x_0)$, where f is the logistic map with $r = 3.9$
  $c_i \leftarrow M_2(x_t) + p_i$
end for

The security of this cipher comes from the initialization of the chaotic map. Specifically, two successive values in the key schedule are used to generate the initial value for the map (x_0) and the number of iterations. If an attacker were to obtain the key schedule, decryption would be simple. It seems that if one does not know the key, it would be difficult to reconstruct the original plaintext from the ciphertext.

To test the security of the cipher, the authors used it to encrypt image data so as to gather a visual measure of the amount of information leakage. They found that the cipher generated data in a periodic fashion. In other words, the pads that are produced by the cipher formed a series that created a pattern of displacement in the ciphertext. The reason for this is that the pad depends entirely upon the key, thus giving the cipher a period equal to the size of the key.

To avoid the periodic behavior of the Simple cipher, the authors implemented a feedback mechanism into its design

so as to vary the pad by both the key values and the output of the previous ciphertext byte. This created a feedback chaining model, as shown in figure (2), and gave the cipher very good statistical properties. It was shown that a change in a single bit in the encryption key changed, on average, 49.6% of the bits in the corresponding ciphertext. Ideally, a change in a single key bit will change 50% of the corresponding ciphertext, so this modification works well.

One problem with this cipher lies in the size of the set of all initial points for the logistic map $S_I = \{x_0, x_1, x_2, ..., x_n\}$. Since the initial points for the logistic map are generated by the mapping M_1 using the iteration keys and previous trajectory values (which are of size 2^8 and between the interval $[0, 1)$, respectively), S_I will have an order of 2^8. Theoretically, this makes the initial points and trajectories of the logistic map susceptible to exhaustive search attacks. To work around this shortcoming the authors could have increased the size of the plaintext/ciphertext blocks and iteration keys to 256 bits to match their alleged key size, which would have increased the size of S_I. However, this also implies that the amount of precision at which these initial points were represented would need to increase. This modification would have an obvious impact on the performance of the cipher.

The authors made an attempt to work around this problem by introducing variability in the number of iterations of the logistic map. The current scheme is to use the sum of an iteration key, previous ciphertext value, and the constant 16 to generate the iteration count. One can deduce that the value 16 was chosen to provide a minimum number of iterations used to generate a pseudorandom value from the chaotic map. However, while it does introduce some pseudorandomness for the number of iterations, this number would provide more security assurances if it was generated from both successive iteration key values. For instance, the values of k_i and k_{i+1} could be XOR'd together and the resulting value could be incremented by 16 to produce the final iteration count. This new scheme should introduce more key-dependent variability which would strengthen the overall security of the cipher if the privacy of the key is maintained.

6.2 Chaotic Feistel Cipher

One recently proposed chaotic block cipher is the Chaotic Feistel cipher by Masuda et. al. [6]. The general structure of the cipher is shown in figure (3), where each round processes a 128-bit block of data.

For this cipher the authors propose a number of different options for the chaos-based mixing transformation, including a 1-D chaotic map, 2-D cat map, and even 4-D torus map [6]. For each chaotic map, the authors analyzed its security from both a dynamical systems and cryptographic perspective. Specifically, they focused their analysis on the Hamming distance between input and output values and the actual

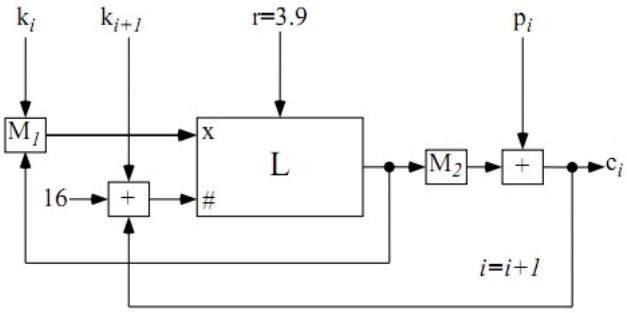

Fig. 2: Block diagram of the Advanced cipher which clearly shows the feedback mechanism used to further randomize the chaotic mappings produced by the logistic map [8].

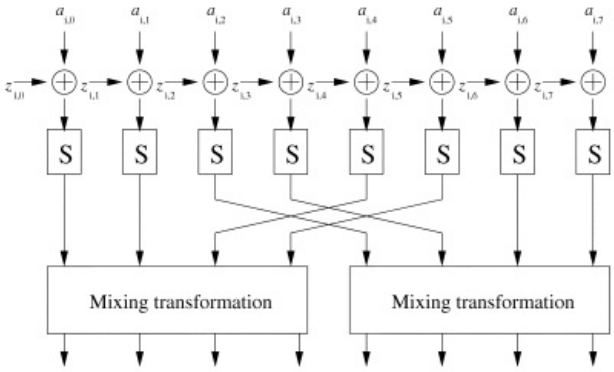

Fig. 3: Block diagram of the Chaotic Feistel cipher proposed in [6]. Each $a_{i,k}, 0 \leq k \leq 7$ is a byte of plaintext that is fed into the cipher.

Euclidean distance between two elements generated by the chaotic maps.

The cipher also relies on chaos-based S-boxes for its non-linear round transformations, which are built from a custom discretized version of the skew tent map. This map was specifically crafted to guarantee small differential and linear probabilities, which are measures of the cipher's susceptibility to differential and linear cryptanalysis attacks, respectively. Each S-box is a one-to-one transformation defined as follows:

$$S_A(X) = F_A^n(X + 1) - 1 for A \in K_S \quad (6)$$

where F_A^n is the discretized tent map for 256 elements built by the key A and consisting of n iterations. K_S is the set of keys available to build the S-box with sufficient security.

The problem with this S-box is that it is based on a discretized chaotic map that does not necessarily preserve the chaotic behavior of its real-valued counterpart. For this reason, the authors approached its analysis using various assumptions about its lack of algebraic structure and S-box input bytes. Through this analysis they were able to

numerically generate an approximate lower bound for the differential probability of the S-boxes, which effectively eliminated potential key values from K due to the high differential probability they produced. Specifically, the key space was reduced in size from its original length of 256 to 64, which resulted in the set K_S.

The small size of this key space is alarming from a security perspective, as block ciphers usually strive to maximize the order of this set. However, given that the differential probability of any possible key from K_S was less than 2^{-4}, the S-boxes were deemed secure. The authors can argue that the variability in the S-box keys introduces randomness that improves its security, but their relatively small order may still make one wonder about their susceptibility to exhaustive search attacks.

Perhaps the most lacking part in their analysis of the S-boxes lies in their approach to measure the linear probability. The authors only state the use of numerical computation methods to determine the S-boxes' susceptibility to linear cryptanalysis attacks. Their work would have benefited from an algebraic analysis in order to determine the exact correlation between transformations of input and output bytes. Although the S-box substitution is non-linear by definition, poor construction of such a transformation could lead to potential linear correlation attacks on the entire cipher.

6.3 Rabbit Cipher

Rabbit is a relatively new stream cipher that was inspired by the random behavior of chaotic maps. Briefly speaking, it is constructed using a chaotic system of coupled non-linear maps that exhibits secure cryptographic properties in its discretized form. It is designed to work with 128-bit data blocks, as both the key and output data are 128 bits in length. Additionally, its internal data structure consists of eight state variables and eight counters. Its design is very similar to the counter mode of operation for traditional block ciphers in that the secret encrypted values can be precomputed and XOR'd with the plaintext for encryption.

The algorithm for the cipher can be broken down into the following four main components: key setup, next state (round) function, counter system, and extraction scheme [9]. The key setup scheme is responsible for initializing the eight individual state variables and counters of the cipher. The mapping between the state variables and counters is a one-to-one correspondence defined by splitting the bits of the key value into 8 individual partitions with some additional manipulations. In order to decrease any statistical correlation between the initial variables and the key, the system is iterated four times using the following next-state function:

$$x_{j,i+1} = g_{j,i} + (g_{j-1,i} <<< 16) + (g_{j-2,i} <<< 16) \quad (7)$$

where, $g_{j,i}$ is defined as:

$$g_{j,i} = ((x_{j,i} + c_{j,i})^2 \oplus ((x_{j,i} + c_{j,i})^2 >>> 32)) \quad (8)$$

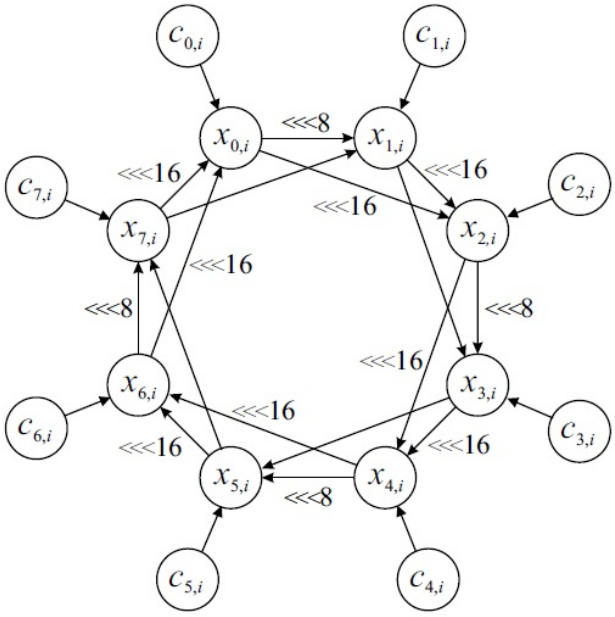

Fig. 4: The next-state function, comprised of eight coupled, non-linear, chaotic maps, used in the Rabbit cipher [9].

Note that all operations are done in modular arithmetic 2^{32} and the index calculations are done in modular arithmetic 8 (since there are 8 state variables). This system of equations can be seen graphically in figure (4).

This map simulates chaotic behavior over the finite domain of integers. However, the cardinality of the field of elements is 2^{32}, which is significantly larger than traditional block cipher fields (i.e. GF(2^8) used in AES). What is unique is that instead of operating over the domain of real numbers (meaning that implementation requires floating point operations), the domain is scaled up by 2^{32} to translate real numbers into integers.

The counter dynamics are expressed using a system of equations similar to the state variable map. The phase space for the counters is the same as the state variables. The counter values are incremented before each system iteration using predefined constants such as $0x4D34D34D$ and $0xD34D34D3$ and carry-over bits from previous iterations. Just as with the next-state function, the addition done to increment the counters is modulo 2^{32}.

After each iteration of the system, the bits of the internal state are extracted and XOR'd with the plaintext to encrypt (or ciphertext to decrypt) the data. This mode of operation is important because no inverse for the next-state function was defined, so it is treated as a one-way function as the cipher operates in a counter-mode.

A thorough security analysis of the operations that set up the secret key (key expansion, system iteration, and counter modification) shows that Rabbit is quite promising. Perhaps the most interesting security property is that the

next-state system iteration function ensures that after only two iterations all of the state bits are affected by all key bits with a probability of approximately 0.5. This value is ideal from a security perspective, but just to be safe the authors have chosen to give a safety margin of four iterations to increase this probability.

Additionally, the counter modification process is very difficult to invert without knowledge of the internal state variables. However, there is the possibility that counter values will be repeated for different keys (periodic behavior exhibiting a pattern), which is detrimental to the overall security of the cipher. Since the counter space is very large, predicting values of the counter is relatively easy since the increment function is based on simple addition operations. In particular, the least significant bit of each counter value has a probability of 1.0 to change, whereas the most significant bit has a probability of 2^{-255} of changing. Despite these drawbacks, the fact that the counter bits carry over into subsequent increment operations means that each bit will have an equal period length.

Further analysis work revealed that the cipher held up well against algebraic and statistical attacks. Their algebraic analysis examined the Hamming distance for the g function. Through simple manipulation of the individual bytes for g, the authors were able to determine that each byte of y in $g(y)$ has an entropy of approximately 7.99, meaning an acceptable level of diffusion was obtained. Unfortunately, the influence of the counter value in g was ignored. If these bytes were included in the byte-wise manipulation of the equation, the dependence results would have been slightly different and more complex, which would almost certainly lead to different results for the Hamming distance.

Also, the authors approached the security of Rabbit mainly from a cryptographic viewpoint, not a dynamical systems one. Simple numerical tests could have been performed to approximate the discrete Lyapunov exponent for the entire chaotic system, thus indicating the actual measure of trajectory divergence for the next-state function. However, given the fact that the chaotic system is comprised of multiple non-linear maps, analyzing the Euclidean distance for trajectories of the 1-D system becomes a matter of measuring the distance for all possible element pairs of the individual maps. Also, the size of the phase space (2^{32}) makes measuring Euclidean divergence difficult, but still an important part of the analysis. While this would certainly increase the complexity of the security analysis, it might reveal characteristics about the system not touched upon by algebraic analysis.

7. Conclusion

It has been argued that a lack of definition for discrete chaos in finite domains based on a discretized version of the Lyapunov exponent plays a large role in the development and security of chaos-based symmetric key ciphers. Many of the modern chaos-based symmetric key cryptosystems suffer from lack of truly chaotic behavior when their internal chaotic maps are discretized for implementation. Furthermore, the inefficiencies of these implementations have had a significant impact on both the performance and design of chaos-based ciphers. Manipulating elements in real-valued systems consists of expensive operations that significantly impact the overall efficiency of the cipher.

Chaos-based ciphers also suffer from a lack of thorough security analysis efforts that critique their design and implementation from both a dynamical systems and cryptographic perspective. It is not enough to consider one paradigm of security for these ciphers, as flaws in one may be enough to reveal a fundamental weakness in the other. Furthermore, analysis efforts should consist of both numerical and algebraic analysis techniques. Given the difficulty of implementing discrete chaos in cryptosystems, both forms of analysis are necessary to uncover potential weaknesses that may lead to successful differential or linear cryptanalysis attacks.

Overall, however, there are certainly elements of chaos theory that make it theoretically applicable to cryptography. For this reason, there has been and will probably continue to be significant research done in chaos-based symmetric key cryptosystems. However, given the loose connection between these two fields thus far, it is difficult to tell if these research efforts will be successful when compared to today's standardized cryptographic primitives and the emerging usage of elliptic curves and other number theoretical concepts in cryptography. Perhaps as chaos theory evolves this connection will become clearer and pave the way for more appropriate cryptography applications. Until then, however, traditional number theory cryptosystems will continue to lead the way into the future.

References

[1] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of the Atmospheric Sciences*, vol. 20, pp. 130–141, 1963.

[2] E. W. Weisstein. Chaos. From MathWorld, A Wolfram Web Resource. [Online]. Available: http://mathworld.wolfram.com/Chaos.html

[3] I. P. Cornfeld, S. V. Fomin, and Y. G. Sinai, "Ergodic theory," *Springer*, 1982.

[4] E. W. Weisstein. Logistic map. From MathWorld, A Wolfram Web Resource. [Online]. Available: http://mathworld.wolfram.com/LogisticMap.html

[5] L. Kocarev, "Chaos-based cryptography: A brief overview," *Circuits and Systems Magazine, IEEE*, vol. 1, pp. 6 – 21, 2001.

[6] N. Masuda, G. Jakimoski, K. Aihara, and L. Kocarev, "Chaotic block ciphers: From theory to practical algorithms," *IEEE: Transactions on Circuits and Systems*, vol. 53, pp. 1341 – 1352, 2006.

[7] L. Shujun, M. Xuanqin, and C. Yuanlong, "Pseudo-random bit generator based on couple chaotic systems and its applications in stream-cipher cryptography," in *Progress in Cryptology Ů INDOCRYPT 2001*, ser. Lecture Notes in Computer Science, C. Rangan and C. Ding, Eds. Springer Berlin / Heidelberg, 2001, vol. 2247, pp. 316–329.

[8] K. Roskin and J. Casper, "From chaos to cryptography."

[9] M. Boesgaard, M. Vesterager, T. Pedersen, J. Christiansen, and O. Scavenius, "Rabbit: A new high-performance stream cipher," *Proc. Fast Software Encryption*, vol. 2887, 2003.