Chaos and Cryptography

Frank Dachselt and Wolfgang Schwarz, *Member, IEEE*

***Abstract*—In this paper, the authors present their opinion and research results collected during the last ten years on the conjunction of chaos and cryptography. Special consideration is taken to show some general critical points and limitations of this approach and to provide some pointers to a comprehensive and careful evaluation of the cryptographical properties of chaotic systems.**

***Index Terms*—Chaotic encryption, cryptography.**

I. Introduction

THE recent research activities in the field of nonlinear dynamics and especially on systems with complex (chaotic) behavior have forced a lot of investigations on possible applications of such systems. Among them also the field of *chaotic encryption* appeared although it was not initially new at this time (see Section III). However, in most of the publications on this topic, a comprehensive and careful evaluation with respect to the cryptographical properties of the designed systems is not provided. Therefore, the reader's attention is drawn to some aspects which, in the authors' opinion, should be kept in mind when chaotic systems are considered from a cryptographical viewpoint.

Furthermore, this article attempts to establish a link to the conventional cryptography which has a wide range of appropriate design and analysis methods. An adaptation of some of these methods for the evaluation of chaotic encryption systems will not only lead to performance measures and appropriate design criteria, it will also allow to compare chaotic encryption systems with conventional ones.

While initially the publications on chaotic encryption were discussed also in the community of applied cryptography [1]–[4], this interdisciplinary debate seems to have ceased. Today, chaotic encryption is almost exclusively considered inside the nonlinear systems community. One reason may be that even the casual evaluation of the first attempts has shown a lot of potential pitfalls and inherent drawbacks. Today, there seems to dominate on one side, a negative opinion about the applicability of chaotic encryption schemes and on the other side the fear of a too strict evaluation.

However, the authors think that the discussion should be reopened on an appropriate level of mutual expectations. May the following thoughts and opinions contribute toward this step.

Manuscript received February 28. 2001; revised August 3, 2001. This work was supported in part by the Deutsche Forschungsgemeinschaft. This paper was recommended by Guest Editor M. Ogorzalek.

The authors are with the Department of Electrical Engineering, Technical University of Dresden, Dresden 01062, Germany (e-mail: dachselt@iee1.et.tu-dresden.de; schwarz@iee1.et.tu-dresden.de).

Publisher Item Identifier S 1057-7122(01)10390-9.

II. Terminology

When speaking about *conventional cryptography* we mean cryptosystems which work on discrete values and in discrete time. This involves the classical cryptography as well as all modern systems being nowadays in practical use.

The crucial points in *chaotic cryptography* are continuous-value information and the usage of continuous-value systems which may operate in continuous or discrete time. To emphasize the difference to conventional cryptography we will use the term *continuous-value cryptography* synonymously with chaotic encryption or chaotic cryptography. In our understanding, it is just a necessity to utilize nonlinearities and to force the system dynamics into a chaotic operation to fulfil basic cryptographical requirements in the continuous-value case.

III. Historical Review

Since its ancient beginnings, cryptographical methods (as far as they are consistent with the today's system concept) has been almost exclusively applied to discrete-value information. These methods range from the so-called Caesar Cipher over the well-known Vegenère Cipher up to modern encryption algorithms like data encryption standard (DES) or the asymmetrical algorithm by Rivest, Shamir, and Adelman (RSA). A comprehensive survey on these and many more conventional ciphers can be found in [5]. With the work of Shannon [6], [7] the development of cryptographical methods became a modern science which is the information-theoretic basis for all encryption systems nowadays in use.

Despite the strong relevance of the discrete-value systems, there have been attempts to apply cryptographical methods to continuous-value information. Early investigations, that were mainly inspired by the increasing research on chaotic systems can be found in [8], [1], [9], [10], [2], [11], [12]. In these applications, autonomous chaotic systems were used as pseudo-random number generators in discrete-value implementations. Thereafter, the pioneering works on chaos synchronization [13]–[18], led to a new branch of applications. Now, nonautonomous chaotic systems with continuous-value signals were used to transmit information. Several schemes have been developed which allow to transform the information signal into a chaotic waveform on the encoder side and to extract the information signal from the transmitted waveform on the decoder side. The most important among them are:

- *Chaotic Masking*: The encoder consists of an autonomous chaotic system whose output signal is added to the information signal. This sum is transmitted over the channel. The decoder uses the transmission signal to synchronize an equivalent chaotic system with the encoder system. The reconstructed chaotic signal is then subtracted from the

transmission signal which finally reconstructs the information signal. In order to guarantee synchronization on the receiver side the information signal has to be sufficiently small with respect to the chaotic signal.

- *Chaos Shift Keying*: The encoder consists of two or more autonomous chaotic systems with different parameters. According to the discrete information signal one of them is selected whose output signal is transmitted over the cannel. In the decoder the same number of chaotic systems tries to synchronize with their encoder counterparts. The parameters are adjusted in such a way that only one pair can synchronize at a time. Detecting this synchronization decodes the discrete information.
- *Chaotic Modulation* or *Inverse System*: The encoder is a nonautonomous chaotic system whose state is influenced by the information signal. The decoder synchronizes with the encoder via reconstruction of its state using the transmission signal. The information signal is recovered by applying the inverse encoder operation to the reconstructed state and the transmission signal.

All of these schemes have been investigated analytically and experimentally in continuous-time as well as in discrete-time applications [19]–[27].

From our today's viewpoint, the inverse-system approach [18] seems to be the most suitable scheme for continuous-value encryption because of its unrestricted signal structure. Furthermore, its structure corresponds to conventional self-synchronizing stream ciphers [28], [29].

What qualifies chaos for encryption purposes? The interest in this application field is mainly triggered by the obvious geometrical signal complexity and the statistical signal properties which can be observed in nonlinear dynamical systems [30]–[32]. In this way, chaotic signals can be thought of as the continuous-value equivalent of discrete-value pseudo-random sequences which are discussed in conventional cryptography.

However, the strength of some modern discrete-value cryptosystems is based on particular algebraic or number-theoretic problems (e.g., the RSA public-key algorithm uses the high computational complexity of factorizing large integers). Continuous-value systems do not offer any equivalent properties. Chaotic encryption therefore depends on "conventional" criteria such as signal statistics and their transformations. Their application is more or less restricted to the classical problem of hiding information in a symmetrical (see Section IV–A) scheme.

In what follows we will mostly restrict our discussion to discrete-time chaotic systems. Because of the same time domain this class has the most common properties with the conventional system structures. However, many of the presented results and ideas apply in a similar manner to continuous-time systems.

IV. Cryptographical Preliminaries

This section presents selected material for a cryptographical background which applies for conventional as well as for continuous-value encryption systems. We consider encryption systems complying with the primary cryptographic objective which is to prevent third parties from gaining access to a particular information, referred to as the plaintext, during transmission or storage.

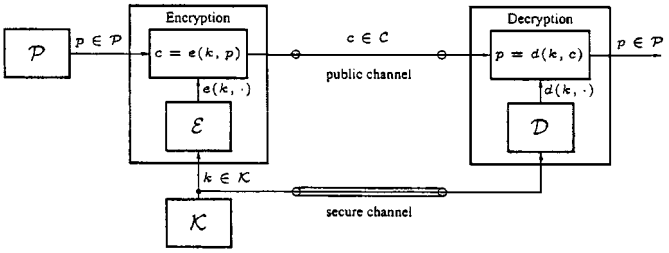

Fig. 1. Elements of a symmetrical public-channel cryptosystem.

A. The Symmetrical Public-Channel Cryptosystem

The analysis of discrete-value cryptosystems is based on a model which characterizes a cryptosystem by five sets [5]:

- the set of possible plaintexts, the plaintext space $\mathcal{P}$;
- the set of possible ciphertexts, the ciphertext space $\mathcal{C}$;
- the set of possible keys, the keyspace $\mathcal{K}$;
- the sets of possible encryption and decryption transformations, the function spaces $\mathcal{E}$ and $\mathcal{D}$.

For each key $k \in \mathcal{K}$, there exist an encryption function $e(k,\cdot) \in \mathcal{E}$ and a corresponding decryption function $d(k,\cdot) \in \mathcal{D}$ such that for each plaintext $p \in \mathcal{P}$ the condition for unique decoding, $d(k, e(k,p)) = p$, is fulfilled.

Encryption systems can be classified according to the publicly available information about the system and its operation. Cryptosystems are usually designed in such a way that its security relies on as few as possible secrets which is expressed by the following assumptions.

- *Public channel*: An opponent has access to the transmission channel such that he knows an arbitrary segment of the ciphertext c.
- *Public structure*: An opponent knows the structure of the encryption system and the a priori probability of the key k that is used. On these conditions, only the secrecy of the used key is needed. This requirement is referred to as *Kerckhoff's principle*.

In our today's understanding, the underlying chaotic operation and the resulting signal properties make a symmetrical structure for chaotic encryption systems necessary. This means, that the key k determines in a similar manner, the encryption function $e(k,\cdot)$ as well as the decryption function $d(k,\cdot)$. Prior to any encrypted transmission, both communication parties have to agree on the key k which must be transferred via an additional secure channel. Thereafter, the ciphertext can be transferred over the public channel. The key k must be kept secret by both communication partners. Such systems are also referred to as secret-key cryptosystems.

Fig. 1 illustrates the elements and their relations in a symmetrical public-channel cryptosystem. We will restrict all of our following considerations to this system class.

In the cryptographic context, the transmission channel is always assumed to be ideal. Although cryptosystems are also investigated in terms of their error-propagation behavior, they provide no means of error detection or even correction. In fact,

it can be shown that the requirements of cryptography and error detection/correction are somewhat contradictory. Thus, the proper channel coding to ensure sufficient transmission quality is a separate problem which has to be dealt with independently of the cryptographical one.

B. Security

The crucial measure for the quality of a public-channel cryptosystem is its capability to withstand the attempts of an opponent to gain knowledge about the plaintext. This measure will be called *security*. The security of a cryptosystem is evaluated by means of *attacks* which try to break the system. The maximum intention of an attack is to determine the used key which allows the opponent to find the decryption function and to decrypt arbitrary ciphertexts as long as they are encrypted with the same key.

In this sense, *perfect security* means that it is impossible to break the cryptosystem, even with unlimited computation resources. As a necessary condition to obtain perfect security Shannon [7] has shown that the number of possible keys has to be at least as large as the number of possible plaintexts. In terms of Shannon's entropy notation this reads as

$$H(P\,|\,C) = H(P) \Rightarrow H(K) \geq H(P) \Rightarrow \|\mathcal{K}\| \geq \|\mathcal{P}\|. \quad (1)$$

Obviously, perfect security is a highly impractical requirement for almost all encryption systems. Instead, *computational security* is determined, which describes the lowest bound for the expense (i.e., computation time, amount of input data, memory space) of all possible attacks. For practical reasons, only the subset of known analysis methods can be considered which results in certain upper security bounds. Thus, computational security can not be evaluated in a positive sense, instead this has to be done indirectly via an "insecurity" estimation.

In other words, while cryptanalysis can prove weakness for a given level of effort, it can not prove that there are no simpler attacks.

C. Attacks

Attacks on a cryptosystem can be distinguished according to the opponent's access to additional information. The most important which apply to symmetrical systems are

- *Ciphertext-only attack*: This is the most restricted situation for the opponent. He has access to the public channel such that he knows some segment of the ciphertext c. Furthermore, certain statistical properties of the plaintext (*a priori* plaintext information) are known.
- *Known plaintext attack*: In addition to the ciphertext segment, the opponent knows also the associated piece of plaintext.

These and other attacks correspond to certain system identification schemes where the unknown key can be considered as the system parameter which is to be identified. This should be observed especially in continuous-value systems because there exist already a great variety of identification algorithms.

Another classification of attacks is according to their strategies and methods used. The attack which defines the highest upperbound of computational effort for breaking a given cipher is the *exhaustive key search*. In this way, any cipher can be attacked with an effort that is proportional to the keyspace size $\|\mathcal{K}\|$. All other attacks try to be more effective by exploiting certain weaknesses. Examples are, differential and linear cryptanalysis, correlation attacks, divide and conquer attacks and combinations hereof.

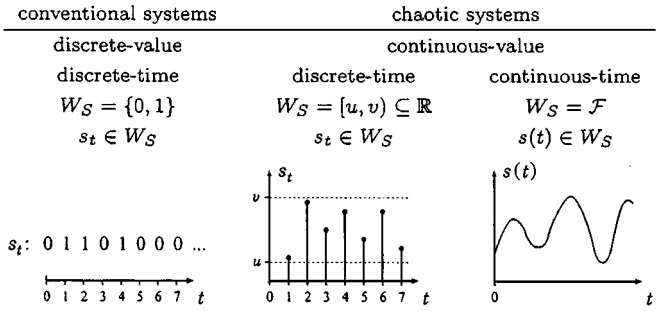

Fig. 2. Symbol-level domains and examples of signals in different classes of encryption systems.

V. Signal Classes in Encryption Systems

As already pointed out in Section II, the most important difference between conventional and continuous-value systems, is the domain of the involved elementary signals. We will call this domain the *symbol level* W_S. It describes the smallest pieces of which the stream of information is composed of. for example, a plaintext p of length L can be written as

$$p = \{s_0, s_1, s_2, \ldots, s_{L-1}\} \quad \text{with } s_i \in W_S. \quad (2)$$

Conventional systems operate on discrete signals. Plaintexts, ciphertexts and keys are elements of finite sets no matter whether these are bits, integer numbers or some kind of metasymbols. Usually the symbol level is binary. The information-theoretic approach to conventional encryption is based on Shannon's theory for discrete information sources and channels.

On the other hand, in *discrete-time continuous-value systems* plaintexts, ciphertexts and keys are in general real values. The symbol level is the real axis or an interval of it. Unfortunately, a straightforward generalization of Shannon's entropy for continuous-value variables does not lead to a compatible notation. This means that the continuous entropy can not be viewed as limit value obtained by discretizing a continuous interval on the real axis into equally-spaced subintervals and letting their number tend to infinity. However, there exist various attempts for an unified approach to discrete and continuous entropy notations [33], which may help to find also a comparable information-theoretic treatment. So far, the comparison of discrete-time systems is restricted to some compatible performance measures (e.g., algorithmic complexity, transformations of signal statistics).

While the above holds for discrete-time systems, the information-theoretic consideration of *continuous-time continuous-value systems* is even more complicated and most of the similarities with conventional systems are lost. In a formal manner we would consider the whole plaintext and ciphertext time functions as elements of the symbol level, because there is no strict mathematical way to break this information down into smaller

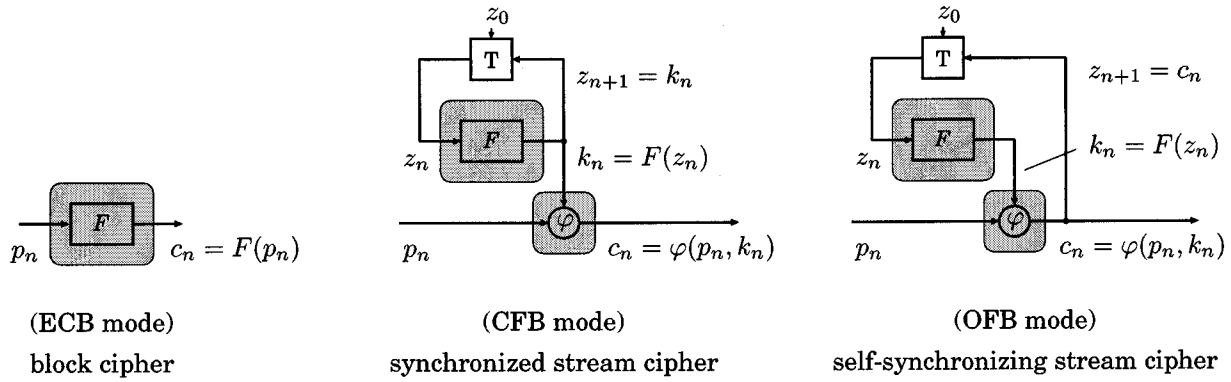

Fig. 3. Modes of operation for the static transformation F.

units. We have to consider the plaintexts as elements of a certain function space $\mathcal{F}$. However, a more semantic-oriented view may suggest to subdivide the time function into segments (usually not of equal length) of separable information elements as they appear for instance in spoken language.

Fig. 2 illustrates the symbol level domains for the above-discussed cases.

VI. Synchronization Schemes

A. Block Ciphers

From the operational point of view, encryption systems can be divided into block and stream ciphers. A block cipher is a static transformation $F_B : W_P \rightarrow W_C$, which operates on the segmented plaintext $p = \{p_0, p_1, p_2, \ldots\}$ such that each plaintext block $p_i \in W_P$ is encrypted independently of all other blocks

$$F_B : \{p_0, p_1, p_2, \ldots\} \rightarrow \{F_B(p_0), F_B(p_1), F_B(p_2), \ldots\} \quad (3)$$

W_P and W_C denote the domains of plaintext and ciphertext blocks with block lengths L_P and L_C, respectively. Thus, each plaintext block p_i consists of L_P symbols $s_{i,j} \in W_S$

$$p_i = \{s_{i,0}, s_{i,1}, s_{i,2}, \ldots, s_{i,L_P-1}\} \in W_P = W_S^{L_P} \quad (4)$$

where $s_{i,j}$ denotes symbol j in block i. Usually, plaintext and ciphertext have identical block domains, i.e., $W_P = W_C = W_S^L$ with $L = L_P = L_C$. Such ciphers are called endomorphic and can be cascaded.

There exist proposals for chaotic-encryption systems that use N-times iterated chaotic maps $c_n = F_B(p_n) = g^N(p_n)$ to transform plaintext into ciphertext blocks [10]. These systems are static transformations. They use the idea of cascading simple ciphers [34] in order to increase the complexity of the static map and gain security. Such systems try to exploit the inherent sensitivity of the chaotic map g to initial conditions, but at the same time they suffer from this property. Since there is no mechanism to avoid the propagation of errors due to initial state or parameter deviations, they are restricted to discrete-value implementations. For a discussion of this topic, we refer to Section VIII.

Static-block transformations are the central element in all cryptographical systems and used as building blocks for more complex systems. They contain all of the uncertainty about the key parameters that is involved in the encryption process and that an opponent tries to reveal.

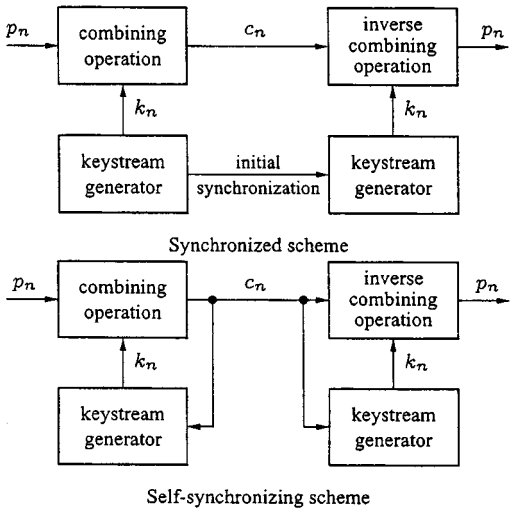

Fig. 4. Synchronization schemes.

B. Stream Ciphers

In many situations, the pure application of a static transformation, the so-called *electronic code book mode* (ECB mode), is not sufficient to fulfil strong cryptographic requirements. This especially holds if the block size is small, for instance, if the blocks are single symbols, i.e., $L = 1$, as it is done in most continuous-value applications. In order to strengthen such ciphers, they are configured as dynamical systems with memory T and a combining function $\varphi(\cdot, \cdot)$ such that the ciphertext block c_n depends on an internal state z_n and possibly on all plaintext blocks $p_0, \ldots, p_n$ appeared so far. The most important structures are the *cipher feedback mode* (CFB) mode and the *output feedback mode* (OFB) mode. These modes of operation which may also be considered for continuous-value systems are summarized in Fig. 3.

When ciphers are used in CFB or OFB mode, then synchronization becomes essential. These two operation modes correspond to two different synchronization principles, the synchronized and the self-synchronizing scheme, illustrated in Fig. 4. In order to ensure an exact information recovery in the decoder, the internal states z_n of the keystream generators have to be

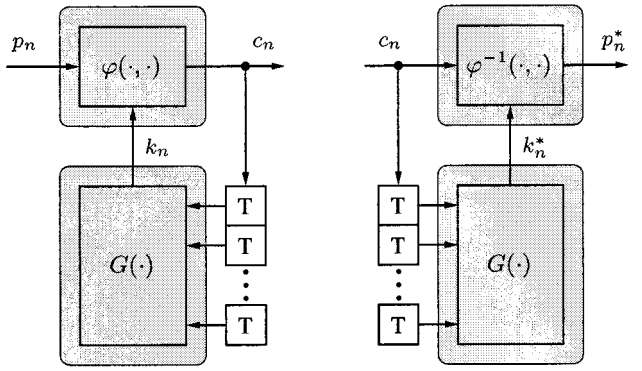

Fig. 5. Canonical structure of a self-synchronizing stream cipher.

identical, i.e., the latter must be synchronized. The synchronized scheme assumes that the keystream generator runs autonomously after an initial synchronization which is obtained via an additional channel (e.g., by setting the initial state or sending synchronization frames over the public channel). The initial state z_0 of the keystream generator may also be part of the encryption key. On the other hand, the keystream generator in the self-synchronizing scheme is nonautonomous and uses the ciphertext stream to synchronize continually.

When stream ciphers are used in conventional cryptography, then the synchronized scheme is preferred because it has a greater potential complexity and its cryptographical properties can be proven more easily [35], [29]. However, this scheme is not suitable for continuous-value implementations because the sensitivity of the nonlinear generators to initial conditions and parameter mismatch causes their states to diverge.

In this sense, chaotic masking may be thought of as the attempt to combine the original synchronized scheme with the capability of a continuous self-synchronization. On the other hand, the inverse-system approach corresponds directly to the self-synchronizing scheme. We will use this principle throughout the further discussion.

A special case of the self-synchronizing scheme in discrete-time systems is the so-called dead-beat synchronization [16]. In this case, the keystream generators are guaranteed to fully synchronize after a finite number N of time steps independently of their initial states. This means that after N steps, the current keystream symbols k_n in the encoder and k_n^* in the decoder are identical. Since no other signals are shared between encoder and decoder, these keystream symbols can only depend on the N preceding ciphertext symbols

$$k_n = k_n^* = G(c_{n-1}, c_{n-2}, \ldots, c_{n-N}) \tag{5}$$

where G denotes the static keystream map. This equation gives rise to a canonical structure of self-synchronizing stream ciphers (SSSC) which is depicted in Fig. 5. In this canonical structure, the state of the keystream generator is given by the last N ciphertext symbols, stored in an N-stage shift register. From the viewpoint of an attack, this means that the analysis of the cipher can be transformed into the pure static problem of analysing the keystream map G and the mixing map φ

$$G : W_C^N \to W_K \quad k_n = G(c_{n-1}, c_{n-2}, \ldots, c_{n-N}) \tag{6}$$

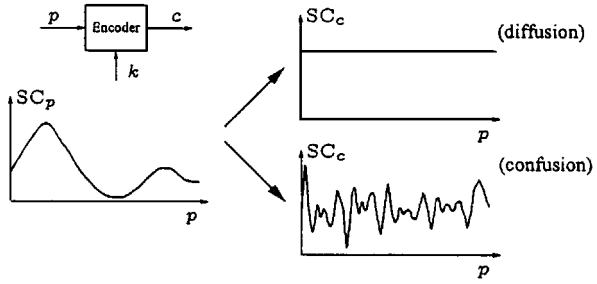

Fig. 6. The weak requirement: good transformations of information-dependent signal characteristics.

$$\varphi : W_P \times W_K \to W_C \quad c_n = \varphi(p_n, k_n) \tag{7}$$

or the combined map $F_C : W_P \times W_C^N \to W_C$ given by

$$c_n = F_C(p_n, c_{n-1}, c_{n-2}, \ldots, c_{n-N}). \tag{8}$$

In other words, the necessary property of self-synchronization implies that the encoder is fully observable. A sufficiently long piece of ciphertext reveals the current internal state of the encoder completely to an observer on the channel.

VII. Design Criteria

The conventional cryptography provides a number of criteria to support the design of encryption systems although there is no design flow which guarantees the construction of cryptosystems with desired strength. While these criteria are mainly given for discrete-value systems, in this section we try to reformulate them for the continuous-value case.

A. Statistical Properties

1) Global Requirements: First we consider the global requirements for encryption systems that correspond to the cryptographical setup given in Section IV. The statistical design criteria can be separated into two classes:

- weak requirement concerning the transformations of *information-dependent signal characteristics*
- strong requirement concerning the transformations of *information-independent signal characteristics*.

An information-dependent signal characteristic SC_p of the plaintext p is one which varies according to the information to be transmitted. Examples are the time function $p(t)$ itself for the transmission of exact values, the short term spectrum for audio signals and the short term mean for noisy measurement signals. These characteristics can be thought of as the carrier of information.

A good cryptographic transformation $\mathcal{P} \to \mathcal{C}$ is one that either makes as many as possible characteristics SC_c of the ciphertext c independent of information-dependent characteristics of p or produces highly complex dependences of SC_c on SC_p as it is illustrated in Fig. 6. These requirements are closely related to the principles of *diffusion* and *confusion* introduced by Shannon [7].

Diffusion aims to spread the influence of a single plaintext symbol over many ciphertext symbols and thus making a ciphertext symbol dependent on many plaintext symbols and consequently flattens information-dependent characteristics in c.

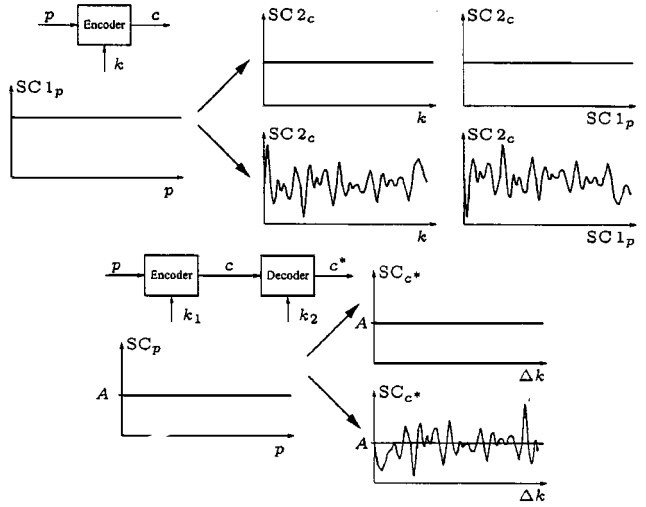

Fig. 7. The strong requirement: good transformations of information-independent signal characteristics.

On the other hand, confusion utilizes nonlinearities in order to achieve a complex correspondence between plaintext and ciphertext symbols which avoids smooth dependences of the ciphertext c on information-dependent characteristics in the plaintext p.

We will call the above criteria the *weak requirement* because it tries to avoid the unauthorized information extraction by an unwanted observer on the channel. This requirement is sometimes also referred to as *privacy*.

In contrast to the weak requirement we need the *strong requirement* in order to prevent systematic attacks on the cryptosystem from being successful at too low expenses.

For that purpose, we consider information-independent signal characteristics SC_p of the plaintext p. These are constant and belong to the a priori plaintext information for ciphertext-only attacks, i.e., they are supposed to be known by the opponent. Examples are long term statistics such as the probability density function or the power density spectrum of p. Transformations of these characteristics during encryption and decryption allow various techniques for analysing the coder system. Two subcases can be distinguished where either the ciphertext c or the decoded plaintext p^* is exploited.

1) The dependence of some signal characteristic $\mathrm{SC2}_c$ of the ciphertext c on the key k and on a known information-independent characteristic $\mathrm{SC1}_p$ of the plaintext p is used. This dependence can be determined by a potential opponent using the publicly available information and can be used during an attack in order to determine the currently used key.
2) The influence of the parameter mismatch Δk on a characteristic SC_{p^*} of the decoded plaintext p^* is considered. During an attack, this can provide a criterion for the adaptive adjustment of the key k_2 in the decoder until the characteristic SC_{p^*} equals to the known information-independent characteristic SC_p.

Fig. 7 demonstrates how the strong requirement can be met. Again, the diffusion and confusion principle can be applied accordingly. In order to make the above-mentioned analysis approaches ineffective, no dependences should exist which an opponent can easily investigate.

2) Uniform Distribution of c_n: In order to achieve a reasonable level of security, encryption systems should try to make every possible plaintext $p \in \mathcal{P}$ an equally likely interpretation of any ciphertext $c \in \mathcal{C}$. Therefore, it is common practice in conventional stream cipher design to strive for the best possible randomness properties of the ciphertext.

The maximum objective would be to produce ciphertexts whose behavior at symbol level is that of an i.i.d. sequence with uniform distribution. While this case would perfectly meet the requirements of Section VII–A–1, it is also identical with Shannon's definition of perfect security as discussed in Section IV–B. For reasons of practicability, this objective is reduced by requiring only local randomness in the sequence of ciphertext symbols. This means that up to a certain order, N the conditional probability density $f_N(\boldsymbol{c}_n \,|\, \boldsymbol{p}_n)$ of ciphertext N-tuples $c_n, \ldots, c_{n-N+1}$ is independent of all previous plaintext symbols $p_0, \ldots, p_n$

$$f_N(\boldsymbol{c}_n \,|\, \boldsymbol{p}_n) = f_N(\boldsymbol{c}_n) = \frac{1}{\|W_C^N\|} \tag{9}$$

where

$$f_N(\boldsymbol{c}_n \,|\, \boldsymbol{p}_n) = f((c_n, \ldots, c_{n-N+1}) \,|\, (p_0, \ldots, p_n)) \tag{10}$$

$$f_N(\boldsymbol{c}_n) = f(c_n, \ldots, c_{n-N+1}). \tag{11}$$

While the requirement in (9) for an Nth-order uniform probability distribution of c_n over its domain W_C holds for discrete-value systems, the corresponding notation for continuous-value systems is less strict. For the latter, any continuous and constant, i.e., information-independent, probability density function for an Nth-order independent ciphertext sequence suffices. In the simplest case one can think of an additional dynamics compression by an invertible static nonlinearity $g(\cdot)$ which is applied to the Nth-order uniform distributed ciphertext. The compressed ciphertext $\tilde{c}_n$ is transmitted and afterwards decompressed into c_n^*, i.e.,

$$\tilde{c}_n = g(c_n) \quad \text{and} \quad c_n^* = g^{-1}(\tilde{c}_n) = c_n. \tag{12}$$

Since we assume an ideal channel there is no information loss during this process. The transmitted ciphertext $\tilde{c}_n$ is still independent of order N with $f_N(\boldsymbol{c}_n \,|\, \boldsymbol{p}_n) = f_N(\boldsymbol{c}_n)$, thus an opponent can not gain more information from $\tilde{c}_n$ than from c_n. In general, this nonuniform distribution of c_n may be an inherent property of the system dynamics of the whole cryptographic transformation.

The above-described difference of discrete-value and continuous-value systems is an outcome of the incompatible notions of discrete and continuous information that were already pointed out in Section V.

Although the choice of the uniform distribution of c_n in continuous-value systems is not strictly necessary from the cryptographical viewpoint, we will prefer it, mainly for two reasons.

1) There exists a well-developed theory for a particular class of continuous-value systems which generate uniformly distributed signals, namely systems with piecewise-linear

maps. This theory provides a wide range of analytical tools for the analysis of relevant properties as there are invariant measures, ergodicity and statistical independence. It is a general principle in cryptography that those systems are preferred whose properties can be proven, even if there are systems whose properties might be stronger.

2) Although the cryptographical model assumes an ideal channel, in real applications the ciphertext will suffer from nonidealities of the continuous-value channel, whose performance is supposed to be worse than that of discrete-value channel. In this context, a uniformly distributed ciphertext shows the best error performance.

3) Balanced Combining Operation: Assuming the canonical structure of a self-synchronizing stream cipher as depicted in Fig. 5, the necessary property (9) of the ciphertext is obtained by a so-called *statically-balanced combiner*

$$c_n = \varphi(p_n, k_n). \tag{13}$$

In such a combining operation, any particular output value c_n can be produced by any value on one input (e.g., on p_n), simply by selecting some appropriate value for the other input (e.g., for k_n). In this way, knowledge of only the output value provides no information, not even statistical information, about either input. Furthermore, if either input, p_n or k_n, is uniformly distributed, then the output c_n has a uniform distribution too, independently of the distribution of the other input. The latter fact is especially required for the keystream sequence k_n

$$\begin{aligned} f(k_n) = \frac{1}{\|W_K\|} &\Longrightarrow f(c_n \,|\, p_n) \\ &= f(c_n) = \frac{1}{\|W_C\|} \end{aligned} \tag{14}$$

Balanced combining operations are also called *correlation immune* combiners [35], [29], in the context of continuous-value systems they have been referred to as *uniform distribution preserving* combiners [36].

In order to transform the ciphertext c_n into the correct plaintext p_n during the decoding process, the combining function needs to be invertible along the path $p_n \rightarrow c_n$ for all pairs $(p_n, k_n) \in W_P \times W_K$

$$c_n = \varphi(p_n, k_n) \Longrightarrow p_n = \varphi^{-1}(c_n, k_n). \tag{15}$$

The most common examples of combining functions in conventional stream ciphers are the binary XOR, the integer-addition modulo 256 (byte addition) and the addition in $\mathrm{GF}(2^8)$ (bitwise XOR of bytes). The simplest suitable combining function for continuous-value systems is the so-called *modulo addition* consisting of an ordinary addition of real numbers followed by a modulo nonlinearity. This is illustrated together with the binary XOR in Fig. 8.

All of the mentioned examples are perfectly balanced, i.e., they fully meet requirement (14). However, from an algebraic viewpoint these simple combiners are inherently linear. Some consequences will be discussed in Section VII–B–2.

The use of balanced combining functions shifts the statistical requirements of the whole encryption transformation into those of the keystream k_n. In order to fulfil requirement (9) the keystream generator has to produce an N-th order uniformly distributed keystream k_n, i.e.,

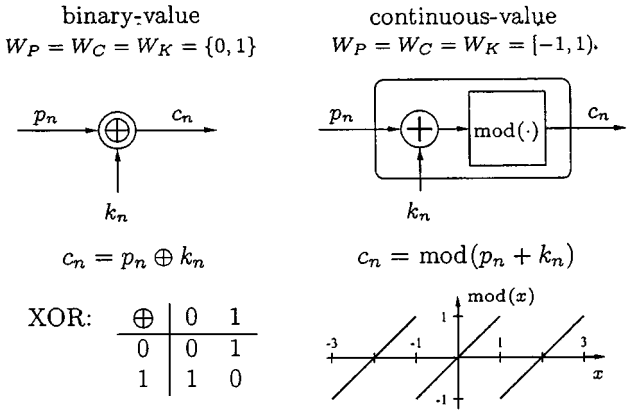

Fig. 8. Examples of balanced combining functions $\varphi(\cdot, \cdot)$ for discrete-value and continuous-value systems.

$$f_N(\boldsymbol{k}_n) = f(k_n, \ldots, k_{n-N+1}) = \frac{1}{\|W_K^N\|}. \tag{16}$$

In this way, we can construct the continuous-value equivalent for the so-called one-time pad which is known to offer perfect security. Given a unique key k which consists of a sufficiently long uniformly distributed i.i.d. sequence k_n over $W_K = [-1, 1)$. If this keystream is applied to the plaintext p_n via the modulo addition as depicted in Fig. 8, then the ciphertext c_n is a unique i.i.d. sequence with uniform distribution and thus meets the requirement (9) for any N. No further complexity is needed to fulfil the criteria of perfect security. In an alternative view, such an encryption transformation may be thought of as a totally disturbed channel along the path $p_n \rightarrow c_n$. There is no information of p_n left in c_n.

B. Complexity Considerations

As already mentioned, there exists the class of piecewise-linear continuous-value systems that offer the ability to generate uniformly distributed signals up to an arbitrary order N as it is required for the keystream k_n according to (16). In this way, chaotic systems are able to fulfil perfectly the statistical demands of cryptography. However, the following considerations show that even those continuous-value systems may possess cryptographical weaknesses in terms of complexity.

In fact, it is a celebrated property of chaotic systems that they can produce highly complex signals using relative simple systems structures. This seems a serious potential pitfall for cryptographic applications.

1) Analytical Complexity: When the order N of the uniform distribution according to the requirements (9) and (16) for a cryptographical design has been specified, then it can be shown that the keystream generator (5) needs at least N delay stages. On the other hand, with N delay stages the best signal statistics that is guaranteed is an N-th order uniform distribution of k_n.

The statistical advantages of piecewise-linear systems have forced several investigations in this class of systems for cryptographical purposes. In what follows we will use a simple structure that is an example outcome of a general design process [36].

This keystream generator uses a linear combination of the delayed ciphertext symbols to produce the keystream

$$k_n = \mathrm{mod}(a_1 c_{n-1} + a_2 c_{n-2} + \cdots + a_N c_{n-N}) \qquad (17)$$

where the coefficient vector $\boldsymbol{a} = (a_1, \ldots, a_N)$ serves as the key parameter. Using the modulo addition also for the combining operation, the encryption transformation can be written as

$$c_n = \mathrm{mod}(p_n + a_1 c_{n-1} + a_2 c_{n-2} + \cdots + a_N c_{n-N}). \quad (18)$$

This encoder structure (depicted in Fig. 9) guarantees an N-th order uniformly distributed key stream, if the coefficient a_N is an integer with $|a_N| > 1$ and the remaining coefficients $a_i, i = 1, \ldots, N-1$, are nonzero.

The self-synchronizing stream cipher (18) can be considered as a single static map (8) with $N+1$ inputs and a single output, where an opponent knows the N input signals $c_{n-1}, \ldots, c_{n-N}$ and the output c_n (see Section VI–B). Thus, for an opponent the following situation arises:

While all information about the plaintext p_n and the key $k = (a_1, \ldots, a_N)$ is completely hidden in the ciphertext and its statistics up to order N the ciphertext probability density of order $N+1$ is not necessarily uniform but uniquely determined by:

- the static map (8), i.e., by the coefficients $a_1, \ldots, a_N$;
- the first-order probability density of p_n.

To illustrate this fact we choose a two-dimensional (2-D) system example of the transformation (18) with

$$N = 2, \quad a_1 = 0.8, \quad a_2 = 2.0, \quad p_n = 0.3 \sin(n/n_0). \quad (19)$$

The probability density of that particular chosen plaintext differs notably from a uniform density. The resulting two- and tree-dimensional trajectory plots of the ciphertext are shown in Fig. 10.

Whereas the first- and second-order distribution of c_n in Fig. 10 are uniform, it is obvious that the third-order distribution of c_n reveals the map F_C completely. Using correlation techniques the estimation of the coefficients $a_1, \ldots, a_N$ is possible even without knowing exact expressions for the probability density of p_n. An example of a breaking method based on these properties can be found in [37]–[39].

The cryptoanalysis based on this property becomes impossible if the distribution of the plaintext symbols c_n is uniform (of first order) over W_C. However, if the second-order plaintext distribution is nonuniform, then the cryptoanalysis can be applied to the ciphertext distribution of order $N+2$ in a similar way. In general, any nonuniform plaintext distribution of order M will reveal information about the map F_C in the ciphertext distribution of order $N+M$.

Of course, the described behavior of the ciphertext distribution holds in the same way for self-synchronizing discrete-value systems. However, a proper design of the discrete-value keystream map and combining function can ensure, that *no* information about these maps is revealed in any higher-order distribution of c_n, unless some a priori information of p_n is known. This qualitative difference results from the inherent property of continuous maps to preserve metric information about their

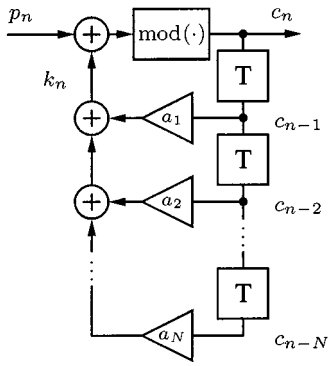

Fig. 9. 2-D example structure.

input signals in the output [40]. Discrete maps, designed at bit level, are able to completely hide this information.

2) Algebraic Complexity: Raising the complexity of the keystream function is usually done by applying combining techniques to basic structures with proven properties. Most common among them are cascading, serial and parallel compositions [34], [28]. Since these combining techniques preserve the property of self-synchronization, also the transformation of the resulting ciphers into their canonical representation remains possible. However, suitable combining may improve the ratio between complexity and realization expense of the keystream function. The following discussion considers the method of cascading for the system structure in Fig. 9.

The particular cryptoanalysis of system (18), presented in [37]–[39], is based on the statistical properties discussed in the previous section. It also uses the fact, that the keystream function is a linear combination of the previous ciphertext symbols. The latter condition is exploited for a so-called *divide-and-conquer attack*. This concept splits a overall cryptoanalysis into several parts, each of which is less complex than the total. Since the analysis expense increases with the order N of the linear combination, we introduce the following complexity measure [41]:

Linear keystream complexity $\mathcal{L}_k$: Let G be the keystream map of the canonical realization of a stream cipher as given in (5). If N is the smallest value for which this map can be realized as a linear combination of $c_{n-1}, \ldots, c_{n-N}$, then the linear keystream complexity $\mathcal{L}_k$ of this stream cipher is equal to N. If this map, for any finite N, can not be realized as a linear combination of $c_{n-1}, \ldots, c_{n-N}$, then $\mathcal{L}_k$ is equal to infinity

$$\mathcal{L}_k := \begin{cases} \min\{N\}, & \text{if } k_n = \sum_{j=1}^{N} a_j\, c_{n-j} \quad \text{with } a_N \neq 0 \\ \infty, & \text{otherwise.} \end{cases} \qquad (20)$$

When two ciphers (18) are cascaded (i.e., the output of the first cipher is connected to the input of the second which generates the final ciphertext), then, the resulting canonical representation depends on the algebraic structure which these ciphers are operating on.

If the continuous-value system (18) is considered with the parameter restriction

$$a_1, \ldots, a_N \in \mathbb{Z} \qquad (21)$$

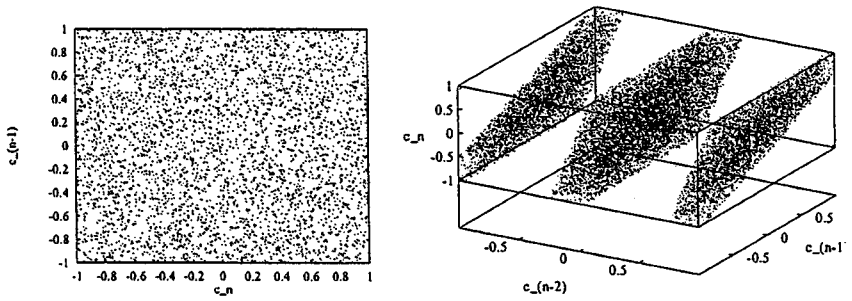

Fig. 10. 2-D and 3-D trajectory plots of c_n for the example system (19).

then, the encryption transformation can be modeled as a linear operator acting on a $\mathbb{Z}$-module [42] with the characteristic polynomial

$$g(x) = 1 \ominus a_1 \odot x \ominus a_2 \odot x^2 \ominus \cdots \ominus a_N \odot x^N. \quad (22)$$

Here, $\ominus$ and $\odot$ denote $\mathbb{Z}$-module subtraction and multiplication, respectively. For $a_N \neq 0$, the linear keystream complexity is $\mathcal{L}_k = \deg(g(x)) = N$. Cascading two systems with $\deg(g_1(x)) = N_1$ and $\deg(g_2(x)) = N_2$, respectively,

$$\begin{aligned} y_n &= \mathrm{mod}(p_n + a_{1,1} \cdot y_{n-1} + \cdots + a_{1,N_1} \cdot y_{n-N_1}) \\ c_n &= \mathrm{mod}(y_n + a_{2,1} \cdot c_{n-1} + \cdots + a_{2,N_2} \cdot c_{n-N_2}) \end{aligned} \quad (23)$$

results in a system structure with a characteristic polynomial

$$g_C(x) = g_1(x) g_2(x) \quad \text{with } \deg(g_C(x)) = N_1 + N_2 \quad (24)$$

and a linear keystream complexity $\mathcal{L}_{k,C} = N_1 + N_2$. Thus, there exists a canonical representation

$$c_n = \mathrm{mod}(p_n + a_1 \cdot c_{n-1} + \cdots + a_{N_1+N_2} \cdot c_{n-N_1-N_2}) \quad (25)$$

describing a system with identical input-output behavior for suitable chosen initial states. The new linear coefficients $(a_1, a_2, \ldots, a_{N_1+N_2})$ can be uniquely determined from the coefficients of the original systems.

If restriction (21) is replaced by the general assumption

$$a_1, \ldots, a_N \in \mathbb{R}, \quad (26)$$

then, the canonical form of the cascaded system (23) is no longer guaranteed to have a linear keystream map. Instead, the new system is described by

$$c_n = \mathrm{mod}\left(p_n + \sum_{j=1}^{N_1+N_2} a_j c_{n-j} + H(\boldsymbol{c}) \right) \quad (27)$$

where

$$H(\boldsymbol{c}) = 2 \sum_{j=1}^{N_1} a_{1,j}\, \mathrm{sd}\left(-c_{n-j} + \sum_{l=1}^{N_2} a_{2,i} c_{n-j-l} \right) \quad (28)$$

with $\mathrm{sd}(x) = 1/2(x - \mathrm{mod}(x))$, is a nonlinear map which can not be resolved as a linear combination of $c_{n-1}, \ldots, c_{n-N}$ for some finite N. According to definition (20), the linear keystream complexity of system (27)–(28) is $\mathcal{L}_{k,C} = \infty$.

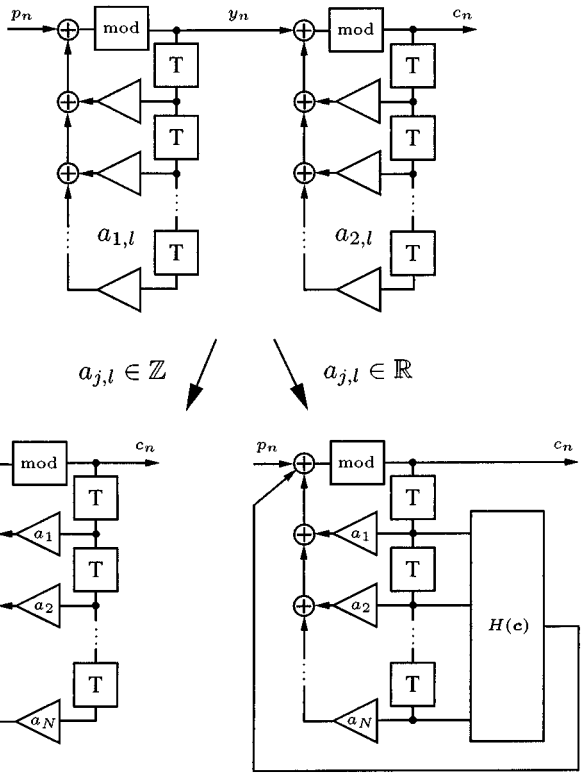

Fig. 11. Cascading two continuous-value ciphers with linear keystream maps.

The described transformations of two cascaded continuous-value ciphers into their canonical representations are illustrated in Fig. 11. As for the case (21) with continuous-value systems, an algebraically similar situation arises for discrete-value systems, when they are operating on certain algebraic structures. As an example we can consider a binary self-synchronizing cipher based on linear feedback shift registers over GF(2). Cascading such ciphers leads to the same complexity properties as for the above-described $\mathbb{Z}$-module systems. These results on cascading are summarized in the following table:

	Binary SSSC over GF(2)	Chaotic SSSC $a_1, \ldots, a_N \in \mathbb{Z}$	Chaotic SSSC $a_1, \ldots, a_N \in \mathbb{R}$
$\mathcal{L}_{k,C}$:	$N_1 + N_2$	$N_1 + N_2$	∞

In this way, continuous-value systems may have advantages in terms of complexity with respect to discrete-value systems when a suitable set of parameters is used as keyspace. On the

other hand, if a cascaded system (23) is designed with the keyspace (26) for $\mathcal{L}_{k,C} = \infty$, then a certain subset of so-called *weak keys* exist (see also Section VIII), namely those given by (21). They result in particular weak encryption conditions, i.e., a finite value of $\mathcal{L}_{k,C}$, which may be exploited for a cryptoanalysis.

VIII. Implementation Aspects

In the previous sections, the authors have restricted their consideration to those continuous-value systems which are capable of continuous-value implementations. The reasons are explained below.

A. Continuous-Value Versus Discrete-Value Realization

It seems to be an attractive idea to implement chaotic systems as discrete-value realizations in order to avoid problems with the inherent sensitivity to parameter and signal deviations, thus for instance making the synchronized scheme applicable for chaotic systems.

Discrete-value realizations are used for computer simulations of chaotic systems all the time. It is an accepted fact, that suitably implemented simulations provide correct results for the statistical properties of chaotic signals. This may also work for cryptographical applications, but besides of signals statistics the performance of encryption systems strongly depends on further system properties.

B. Binary Sublevel

Encryption systems are designed on symbol level, which is chosen according to the smallest separable entities of information in the plaintext. In fact, modern secret-key cryptosystems are almost exclusively designed at bit level [43]. This ensures at best, that no unwanted behavior exists in other signal levels.

The design of continuous-value system is based on a symbol level which is given by real numbers. But implementing such systems digitally creates a binary sublevel depending on the discrete-value representation of the real signal values. Furthermore, all mathematical operations of the continuous-value system can be subdivided into a corresponding set of operations on the binary sublevel. The design of a chaotic encryption system is completely unaware of this sublevel and its properties. This may cause serious weaknesses which an opponent can use for attacks at this sublevel.

The following discussion collects some of these noteworthy properties.

C. Weak Keys

Any discrete-value implementation becomes necessarily a finite-state machine which may effect the system behavior. In this context, the periodicity of generated sequences is the most important problem for cryptographical applications. This holds for autonomous generators in synchronized schemes as well as for the nonautonomous keystream generator in self-synchronizing systems when the plaintext is constant or periodic. Keystream and ciphertext sequences with short periods are considered as cryptographically weak. Since the continuous-value design does not care about periodicity, sequences with short periods may occur accidentally. In order to avoid such situations in would be necessary to find all those keys which may lead to short periods and exclude them from the keyspace. However, there exist no systematic approach to find these *weak keys* other than an exhaustive simulation.

D. Floating-Point Implementation

Because of its wide dynamic range the floating-point implementation seems to be most appropriate for discrete-value realizations of continuous-value encryption systems. However, floating-point numbers are not uniformly distributed over any given interval of the real axis [44]. This means that the distribution of number occurrences in the ciphertext does not correspond to the distribution of the ciphertext, which the continuous-value system was designed for. If a uniform distribution has been designed, then there are some numbers with a relatively high probability of occurrence and a larger set of numbers with a relatively low probability of occurrence. From the communications viewpoint this may be seen as an information loss.

Furthermore, the existence of redundant number representations has to be observed. Due to the normalized calculations in floating-point arithmetic some floating-point numbers represent the same real signal value. On the other hand, not all binary patterns can occur at the output of the encoder and the decoder. As a consequence of this fact the floating-point implementation of continuous-value encryption systems is not suitable for the transmission of arbitrary binary patterns.

E. Realization Expense

In order to compensate the above-mentioned side effects of discrete-value realizations it would be necessary to carefully investigate to underlying operations at the binary sublevel and the continuous-value design is certainly needed to be modified. However, it is our considered opinion that these modifications would degenerate most parts of the continuous-value design obtained before. Finally, the cryptographic properties would mainly depend on the binary sublevel operations. In this way it seems always preferable to design discrete-value encryption systems using conventional methods. They will certainly increase the cryptographic performance and decrease the realization expense.

IX. Application Areas

With the restriction to continuous-value implementations and the general cryptographic weaknesses of continuous-value systems as shown in preceding discussion, chaotic encryption may in our today's opinion not be suitable for applications with high security demands. However, since a number of cryptographic requirements can be met to a certain extent chaotic systems may nevertheless be useful in some specific situations.

We do not expect applications in those areas, where also digital hardware or software could be used, because conventional encryption serves for any cryptographic demand more efficiently. However, there may be information systems where continuous-value signals are essential, i.e., where the information source produces and the information sink demands continuous-value signals and the information processing is

done on a continuous-value base. Such implementations can include for instance SC- and SI-circuits where no digital channel is required. Adding digital processing hardware exclusively for encryption including the necessary components such as A/D- and D/A-converters may cause additional costs or other unwanted side effects. In these situations also continuous-value encryption can provide a reasonable level of privacy and even a certain amount security against simple attacks.

A further branch of possible applications may arise in such situations where identification is based on cryptographic methods and individual secrets are stored in widely distributed places. Such systems, e.g., smartcard-based solutions, are subject to active tampering. Integrated digital hardware and software are known to be easily analysable at layout level and digitally stored parameters can be revealed by certain techniques. Thus, identification attributes can be reproduced and systematically changed by sufficiently equipped attackers. Using additional analog hardware components and continuous-value signals within such systems may provide more tamper-resistant schemes by raising the costs for parameter detection and reproduction. The usage of biometric-based identification schemes has been triggered for the same reason. While this situation is beyond the cryptographical scope of this article as described in Section IV, the requirements for the essential signal properties remain unchanged.

X. Conclusion

Discrete-time continuous-value systems are able to fulfil the statistical requirements of cryptographic transformations, even perfectly. They can be designed in similar ways as conventional self-synchronizing stream ciphers. There exist equivalent subsystems such as balanced combining operations and keystream generators producing N-order uniformly distributed keystreams.

In terms of complexity, continuous-value systems suffer from the inherent continuous nature of the involved maps. In conjunction with system structures of low algebraic complexity they may lead to considerable weaknesses.

In the authors' opinion applications of continuous-value encryption systems may be found in situations which require only moderate levels of security. Especially, these systems could be useful when continuous-value signal processing is es sential.

References

[1] D. D. Wheeler, "Problems with chaotic cryptosystems," *Cryptologia*, vol. XIII, no. 3, pp. 243–250, 1989.

[2] E. Biham, "Cryptanalysis of the chaotic-map cryptosystem suggested at EUROCRYPT'91," in *Advances in Cryptology—EUROCRYPT'91*, D. W. Davies, Ed. New York: Springer-Verlag, 1991, vol. 547, pp. 532–534.

[3] T. Beth, D. E. Lazic, and A. Mathias, "Cryptanalysis of cryptosystems based on remote chaos replication," in *Advances in Cryptology—CRYPTO'94*, Y. G. Desmedt, Ed. New York: Springer-Verlag, 1994, vol. 839, pp. 318–331.

[4] W. G. Chambers, "Comments on "chaotic digital encoding: An approach to secure communication"," *IEEE Trans. Circuits Syst. II*, vol. 46, pp. 1445–1447, Nov. 1999.

[5] D. R. Stinson, *Cryptography, Theory and Practice*. Boca Raton, FL: CRC, 1995.

[6] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 623–656, 1948.

[7] ——, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, pp. 656–715, 1949.

[8] R. Matthews, "On the derivation of a "chaotic" encryption algorithm," *Cryptologia*, vol. XIII, no. 1, pp. 29–41, 1989.

[9] D. W. Mitchell, "Nonlinear key generators," *Cryptologia*, vol. XIV, no. 4, pp. 350–354, 1990.

[10] T. Habutsu, Y. Nishio, I. Sasase, and S. Mori, "A secret key cryptosystem by iterating a chaotic map," in *Advances in Cryptology—EUROCRYPT'91*, D. W. Davies, Ed. New York: Springer-Verlag, 1991, vol. 547, pp. 127–140.

[11] J. M. Carroll, J. Verhagen, and P. T. Wong, "Chaos in crytography: The escape from the strange attractor," *Cryptologia*, vol. XVI, no. 1, pp. 52–71, 1992.

[12] R. Anderson, "Chaos and random numbers," *Cryptologia*, vol. XVI, no. 3, p. 226, 1992.

[13] L. M. Pecorra and T. L. Carroll, "Synchronization in chaotic systems," *Phys. Rev. Lett.*, vol. 64, no. 8, pp. 821–824, 1990.

[14] T. L. Carroll and L. M. Pecorra, "Synchronizing chaotic circuits," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 453–456, Apr. 1991.

[15] L. M. Pecorra and T. L. Carroll, "Driving systems with chaotic signals," *Phys. Rev. A*, vol. 44, no. 4, pp. 2374–2383, 1991.

[16] A. de Angeli, R. Genesio, and A. Test, "Dead-beat chaos synchronization in discrete-time systems," *IEEE Trans. Circuits Syst. I*, vol. 42, pp. 54–56, Jan. 1995.

[17] P. Celka, "Synchronization of chaotic systems through parameter adaptation," in *Proc. Int. Symp. Circuits and Systems (ISCAS'95)*, vol. 1, 1995, pp. 692–695.

[18] U. Feldmann, M. Hasler, and W. Schwarz, "Communication by chaotic signals: The inverse system approach," *Int. J. Circuit Theory Appl.*, vol. 24, no. 5, pp. 551–579, 1996.

[19] C. W. Wu and L. O. Chua, "A simple way to synchronize chaotic systems with applications to secure communication systems," *Int. J. Bifurcation Chaos*, vol. 3, no. 6, pp. 1619–1627, 1993.

[20] D. R. Frey, "Chaotic digital encoding: An approch to secure communication," *IEEE Trans. Circuits Syst. II*, vol. 40, no. 10, pp. 660–666, Oct. 1993.

[21] S. Papadimitriou, G. Pavlides, A. Bezerianos, and T. Bountis, "Chaotic systems of difference equations for real-time encryption," in *Proc. Workshop Nonlinear Signal and Image Processing (NSIP'95)*, 1995, pp. 145–149.

[22] K. Kelber, T. Falk, M. Götz, W. Schwarz, and T. Kilias, "Discrete-time chaotic coders for information encryption—Part 2: Continuous- and discrete-value realization," in *Proc. Workshop Nonlinear Dynamics of Electronic Systems (NDES'96)*, 1996, pp. 27–32.

[23] A. Leuciuc and V. Grigoras, "Multi-parameter chaos modulation of discrete-time filters," in *Proc. Workshop Nonlinear Dynamics of Electronic Systems (NDES'97)*, 1997, pp. 81–86.

[24] S. Papadimitriou, A. Bezerianos, and T. Bountis, "Secure communication with chaotic systems of difference equations," *IEEE Trans. Comput.*, vol. 46, p. 27, JAN. 1997.

[25] H. Zhou and X. T. Ling, "Problems with the chaotic inverse system encrytion approach," *IEEE Trans. Circuits Syst. I*, vol. 44, pp. 268–271, Mar. 1997.

[26] A. Leuciuc, "Information transmission using chaotic discrete-time filter," *IEEE Trans. Circuits Syst. I*, vol. 47, pp. 82–88, Jan. 2000.

[27] Z. He, K. Li, L. Yang, and Y. Shi, "A robust digital secure communication scheme based on sporadic coupling chaos synchronization," *IEEE Trans. Circuits Syst. I*, vol. 47, pp. 397–403, Mar. 2000.

[28] U. M. Maurer, "New approaches to the design of self-synchronizing stream ciphers," in *Advances in Cryptology—EUROCRYPT'91*, D. W. Davies, Ed. New York: Springer-Verlag, 1991, vol. 547, pp. 458–471. of *Lecture Notes in Computer Science*.

[29] R. A. Rueppel, "Stream ciphers," in *Contemporary Cryptology*, G. J. Simmons, Ed. Piscataway, NJ: IEEE Press, 1992, ch. 2, pp. 65–134.

[30] L. Kocarev, G. Jakimoski, T. Stojanovski, and U. Parlitz, "From chaotic maps to encryption systems," in *Proc. Int. Symp. Circuits and Systems (ISCAS'98)*, vol. IV, 1998, pp. 514–517.

[31] K. M. Roskin and J. B. Casper. (1998) From Chaos to Cryptography. [Online]. Available: http://xcrypt.theory.org/

[32] T. Yang, C. W. Wu, and L. O. Chua, "Cryptography based on chaotic systems," *IEEE Trans. Circuits Syst. I*, vol. 44, pp. 469–472, May 1997.

[33] G. Jumarie, *Relative Information*. New York: Springer-Verlag, 1990.

[34] S. Even and O. Goldreich, "On the power of cascade ciphers," *ACM Trans. Comput. Syst.*, vol. 3, no. 5, pp. 108–116, 1985.

[35] R. A. Rueppel, *Analysis and Design of Stream Ciphers*. New York: Springer-Verlag, 1986.

[36] M. Götz, K. Kelber, and W. Schwarz, "Discrete-time chaotic coders for information encryption—Part I: Statistical design approach," *IEEE Trans. Circuits Syst. I*, vol. 44, pp. 963–970, Oct. 1997.

[37] F. Dachselt, K. Kelber, and W. Schwarz, "Chaotic coding and cryptoanalysis," in *Proc. Int. Symp. Circuits and Systems (ISCAS'97)*, vol. II, 1997, pp. 1061–1064.

[38] ——, "Discrete-time chaotic coders for information encryption—Part III: Cryptographical analysis," *IEEE Trans. Circuits Syst. I*, vol. 45, pp. 983–988, Sept. 1998.

[39] W. Schwarz, M. Götz, K. Kelber, A. Abel, T. Falk, and F. Dachselt, "Statistical analysis and design of chaotic systems," in *Applications of Chaotic Electronics to Telecommunications*, P. Kennedy, R. Rovatti, and G. Setti, Eds. Boca Raton, FL: CRC, 2000, ch. 9, pp. 253–305.

[40] F. Dachselt and W. Schwarz, "Discrete versus continuous maps—A cryptographical comparison," in *Proc. Workshop Nonlinear Dynamics of Electronic Systems (NDES 2000)*, 2000, pp. 66–70.

[41] F. Dachselt, K. Kelber, J. Vandewalle, and W. Schwarz, "Chaotic versus classical stream ciphers—A comparative study," in *Proc. Int. Symp. Circuits and Systems (ISCAS'98)*, vol. IV, 1998, pp. 518–521.

[42] F. Dachselt, M. Götz, and W. Schwarz, "Discrete-time chaotic systems homomorph on modules," in *Proc. Workshop Nonlinear Dynamics of Electronic Systems (NDES'99)*, 1999, pp. 17–20.

[43] J. Daemen and V. Rijmen. (2000) The Rijndael Block Cipher. American National Institute of Standards and Technology (NIST). [Online]. Available: http://csrc.nist.gov/encryption/aes/

[44] D. E. Knuth, *The Art of Computer Programming*. Reading, MA: Addison Wesley, 1998, vol. 2.

Frank Dachselt received the Diplomingenieur (M.Sc.) degree in electrical engineering from the Technical University of Dresden, Germany, in 1996. He is currently working toward the Ph.D. degree at the Department of Electrical Engineering of the Technical University of Dresden, Dresden, Germany.

His research interests are in the field of nonlinear dynamical systems, especially including statistical aspects and nonlinear signal processing.

Wolfgang Schwarz (M'1992) received the Dipl.-Ing. (M.Sc.), the Dr.-Ing. (Ph.D.) and the Dr.-Ing. habil. (D.sc.) degrees from the Technical University of Dresden, Dresden, Germany, in 1965, 1969 and 1976 respectively.

From 1965 to 1969, he was a Lecturer and Senior Lecturer at the Faculty of Electrical Engineering at the Technical University of Dresden. From 1969 to 1974, he was an Assistant Professor at the Engineering College, Mittweida, Germany, where he condcted lectures in control engineering, and conducted research on Robot Control Systems. From 1974 to 1977, he was a member of the R&D Staff of Starkstromanlagenbau, Chemnitz, Germany, where he worked in the development of machine tool CNC systems. From 1977 to 1983, he was a Professor of information engineering and Head of Department of Information Electronics at the Engineering College Mittweida, Germany. In 1974 and 1977, he was an Invited Professor at the Moscow Telecommunication Institute, Moscow, U.S.S.R., where he gave lectures on stochastic signals and dynamical systems. From 1983 to 1992, he was a Professor of electronic circuits at the Technical University of Dresden, and since 1992, he has been full Professor for fundamentals of electrical engineering and electronics at the same university. In 1992, he was Invited Researcher at the University of California, Berkeley. Dr. Schwarz was Founding Chairman of the Institute of Fundamentals of Electrical Engineering and Electronics at the Technical University of Dresden from 1990 to 1993. He teaches fundamentals of Electrical Engineering and Electronics and Electronic Circuits. His research interests are in the fields of nonlinear dynamic systems and circuits. Together with professor A.C. Davies, King's College, University of London he initiated the International Workshop on Nonlinear Dynamic Electronic Systems (NDES) in 1993. He was the local organizer of the V. EUROCHIP Workshop on VLSI Design Training in Dresden, Germany. In 1999, he was Technical Chair of the international workshop on Nonlinear Dynamics in Electronic Systems (NDES 1999) in Roenne, Bornholm, DK and in 2000 the General Co-chair and local arrangements chair of the International Symposium on Nonlinear Theory and its Applications (NOLTA 2000) in Dresden, Germany.