Communicating with noise: How chaos and noise combine to generate secure encryption keys

Ali A. Minai[a] and T. Durai Pandian
Complex Adaptive Systems Laboratory, Department of Electrical & Computer Engineering and Computer Science, University of Cincinnati, Cincinnati, Ohio 45221-0030

(Received 8 December 1997; accepted for publication 11 March 1998)

An approach for the secure transmission of encrypted messages using chaos and noise is presented in this paper. The method is based on the synchronization of certain types of chaotic oscillators in response to a common noise input. This allows two distant oscillators to generate identical output which can be used as a key for encryption and decryption of a message signal. The noiselike synchronizing input—which contains no message information—is communicated to identical oscillators in the transmitter and the receiver over a public channel. The encrypted message is also sent over a public channel, while the key is never transmitted at all. The chaotic nature of the oscillators which generate the key and the randomness of the signal driving the process combine to make the recovery of the key by an eavesdropper extremely difficult. We evaluate system performance with respect to security and robustness and show that a robust and secure system can be obtained. © *1998 American Institute of Physics.* [S1054-1500(98)00403-0]

Encryption of messages is the most common technique used for electronically communicating sensitive information. However, this requires that the receiver know the ''key'' used by the sender to encrypt the message. In classical cryptosystems, this key, which is typically a sequence of numbers, is usually sent over a secure channel while the encrypted message is transmitted over a public channel. It would, of course, be very useful if the key did not need to be transmitted at all. This is usually achieved by generating the key synchronously at the transmitter and receiver using identical seeds in a random-number generator. However, this requires exact pre-synchronization between the sender and the receiver. In this paper, we present a scheme which allows a receiver with the correct parameters to decode the encrypted signal without pre-synchronization, making the system very flexible. In this scheme, the sender transmits a noiselike signal instead of the key, and the key is reproduced at the receiver in response to this signal. The noiselike signal can, of course, be sent over a public channel, since it contains no information for someone not in possession of the appropriate key generator, which, in this case, is a type of chaotic oscillator. The chaotic nature of this oscillator ensures that recovering the key by trial and error is extremely difficult for an eavesdropper. The system illustrates how noise can, paradoxically, serve a useful function in chaotic systems but not in those with more regular, periodic dynamics.

I. INTRODUCTION

Several approaches have recently been proposed for secure communications using synchronized chaotic systems. Most of these methods are based on the principle of chaotic masking, where the message signal is added to a much more powerful chaotic carrier generated by the transmitter system. It is recovered at the receiver by regenerating the carrier through synchronization and subtracting it out of the received signal.[1–3] A more general approach is to use the message signal as a driver for the transmitting system, so that the message becomes a ''dynamical perturbation'' rather than an additive one.[4,5] Another approach is to use the message signal to modulate the parameters of the transmitting system.[2,6,7] Several variants and refinements of these methods, as well as other techniques, have been proposed in the literature.[8–12] The impetus for much of this research has come from the Pecora and Carroll seminal work on synchronizing chaotic systems.[13,14]

Recently, we have reported that a class of discrete-time chaotic maps can be synchronized by a common noiselike input signal,[15,16] and can, therefore, be used for secure communication via chaotic masking in the standard way. In this paper, we report on a very different scheme for secure communication using these oscillators. This approach differs from all others presented so far in that it relies purely on noise to achieve synchronization between the transmitter and receiver. Our method is closely related to that proposed by Yang *et al.*,[17] with one major difference: The transmitted signal in their system is a scalar derived from the encrypted message and the transmitter state, whereas we transmit the encrypted message and a pure noise signal over two channels. Our system, therefore, includes a dependency on an external random driver distinct from the message. Indeed, if the sender and receiver agree beforehand on a driving signal produced by, say, a jointly observable process (e.g., a public ticker, or the instantaneous temperature at a certain location), the noise signal need not even be transmitted.

The system we propose belongs essentially to the class of self-synchronizing stream ciphers[18,19] with a nonlinear key generator. Most such systems produce their keys through

[a]Electronic mail: ali.minai@uc.edu

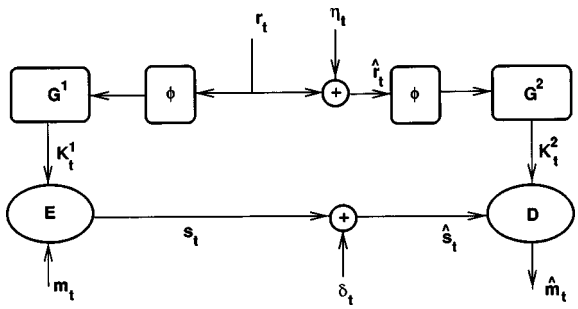

FIG. 1. The system architecture. The transmitter subsystem is on the left and the receiver on the right (see the text for a description).

pseudorandom number generators using a common seed, i.e., the transmitter and receiver key generators must start out pre-synchronized. This is problematic in real-time applications—especially when the message must be broadcast to several receivers, who would all need to synchronize precisely. The system we describe does not require knowledge of a common seed, since synchronization happens dynamically and emergently. The receiver and sender *do not* need to coordinate the timing of their key generators. Any receiver with the correct decoding system can ''join'' the stream at any time, become synchronized with the sender in a short time, and start reading off the message. This creates an ''environment'' in which all legitimate receivers can read a broadcast message while intruders cannot.

To be practical, a system such as we describe above must meet two criteria

(1) **Security:** The system must be such that an intruder cannot decode the message without access to the *precise* system parameter values used by the receiver and the sender. Even a slight mismatch should make recovery impossible. Further, it should be extremely difficult for the intruder to obtain the precise parameter values *even if a correct message or correct key fragment is intercepted*.
(2) **Robustness:** The system should be tolerant of additive noise in the communication channel.

These two requirements are actually rather stringent. We are essentially asking for a system with extreme sensitivity to parameter mismatch *and* significant insensitivity to input perturbations. Typically, chaotic systems will satisfy the first requirement, but not the second. However, we show that, by using a particular type of synchronizing signal, we can meet both requirements.

One point that must be made here is with regard to the term ''secure.'' We, like others in the field of chaos-based communication, use the term for a system that is ''difficult to break.'' We do not imply that our system meets any specific U.S. or international standards for security. However, we hope that the approach taken in this paper will point the way towards a more rigorous evaluation of chaos-based communication systems.

II. SYSTEM DESCRIPTION

Figure 1 shows the architecture of the system described in this paper. The key generators, G^1 and G^2, are identical discrete-time nonlinear oscillators driven by a noiselike external input, r_t, which we term the *synchronizing signal*. The *plaintext message*, m_t, is encrypted by the *enciphering transformation*, E, *using key*, K_t^1, generated by G^1. The resulting *ciphertext message*, s_t, is sent over a public channel to the receiver system, where it is decrypted by the *deciphering transformation*, D, using key K_t^2, generated by G^2, which is driven by the same noiselike input, r_t, as G^1. It is assumed that both E and D use identical keys (one key system[18]). Thus, if K_t^1 and K_t^2 are perfectly synchronized, the deciphered plaintext message, $\hat{m}_t$, is correct. The synchronizing signal, r_t, can either be communicated to the key generators over a separate public channel, or be read from a jointly observable source, such as a public information ticker. Both r_t and s_t are subject to additive channel noise, which is the chief impediment to communication. These noises are denoted by η_t and δ_t, respectively. The block denoted by ϕ is a simple function for transforming r_t before it is used by the key generators.

The oscillators used as key generators are each specified by the map

$$z_{t+1} \equiv F(z_t, u_t) = \tanh[\mu(az_t + u_t)] - \tanh[\mu b z_t], \qquad (1)$$

where z_t denotes the state of the oscillator at time t, u_t is the driving input, and μ, a, and b are fixed parameters. This oscillator is termed a $\mu/a/b$-oscillator. It was studied extensively by Wang[20,21] as a model of competing excitatory and inhibitory neural populations. For $u_t = 0$, $\forall t$ and $a \geqslant 2b$, the map undergoes period-doubling bifurcation to chaos as μ is increased.[20,21] The dynamics of z_t has two domains of attraction separated by $z_t = 0$. If z_0 and u_t have the same sign for all t, the dynamics remains confined to the basin with the corresponding sign. Otherwise, it can switch basins with a small hysteresis.[22] We will assume $z_t > 0$ for all t. If μ is set such that the system is chaotic for $u = 0$, increasing u leads to a series of period-halving bifurcations culminating in a period-2 cycle for large u.[22] A time-varying input, therefore, causes to system to switch between various periodic and chaotic regimes in a continuous series of bifurcations. This can also be seen as the selection of a new map at each time-step.[23,24] Figure 2 shows the map for the 5/5/1-oscillator with fixed input values: $u = 0$ and $u = 0.2$.

When two or more identical $\mu/a/b$-oscillators are driven by the same aperiodic input signal, they synchronize rapidly under a variety of conditions.[15,16] This effect is a consequence of repeated transitions between periodic and chaotic regimes. In the periodic regimes, the oscillators tend to coalesce into a few phase-locked clusters, while in the chaotic regimes, the phase relationship *between* clusters is lost, leading to cluster merging on return to a periodic regime.[15,16] Repeated coalescence and cluster-merging quickly leads to a synchronized state if the average Lyapunov exponent of the system is negative, and certain dynamical conditions are satisfied[15,25] (a negative Lyapunov exponent alone does not guarantee synchronization). We have previously reported in detail on determining the regimes of synchronization when the driving input is a random telegraph signal (RTS),[15] as well as some results for other types of random inputs.[16] An interesting aspect of this synchronization phenomenon is that

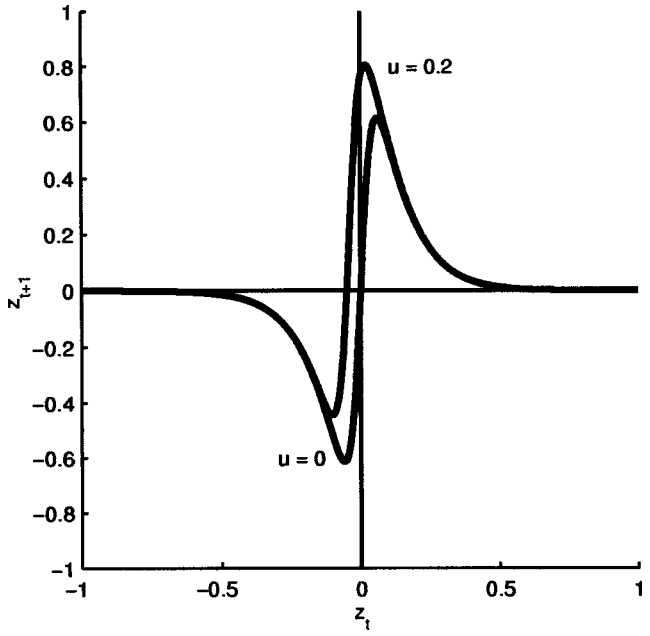

FIG. 2. The 5/5/1 map for fixed inputs $u=0$ and $u=0.2$.

it occurs only in systems which are capable of both chaotic and periodic behavior, and can be switched between these by an external input. The resulting system is, therefore, *intermittently chaotic* but its dynamics is attracted to the phase-space manifold which corresponds to synchronized dynamics. A purely periodic or purely chaotic system would not synchronize in response to noise in this manner.

III. SYSTEM OPERATION AND ANALYSIS

We assume that the message being communicated takes real values in some finite set, $\mathcal{M}\subset\Re$. Since our focus is not on the encryption process, but on key generation, we use a simple n-shift cipher given by

$$\begin{aligned} s_t = E(m_t) &= \Psi(\ldots\Psi(\Psi(m_t,K_t^1),K_t^1)\ldots,K_t^1) \\ &= \Psi^n(m_t,K_t^1), \end{aligned} \tag{2}$$

where $\Psi(x,y)$ is defined by

$$\Psi(x,y)=\begin{cases} (x+y)+2w: & -3w\leqslant(x+y)\leqslant -w, \\ (x+y): & -w<(x+y)<w, \\ (x+y)-2w: & w\leqslant(x+y)\leqslant 3w. \end{cases} \tag{3}$$

Parameter w, which is the width of the transformation, is chosen such that $-w<x,y<w$. Decryption is done by the same algorithm, using $-K_t^2=-K_t^1$ as the key:

$$\begin{aligned} \hat{m}_t = D(\hat{s}_t) &= \Psi(\ldots\Psi(\Psi(\hat{s}_t,-K_t^2),-K_t^2)\ldots,-K_t^2) \\ &= \Psi^n(\hat{s}_t,-K_t^2). \end{aligned} \tag{4}$$

This is by no means the only cipher that can be used with our system. Its choice is motivated by its simplicity and its use in the literature.[17]

The sender's key, K_t^1 is generated from a $\mu/a/b$-oscillator using an input, $\phi(r_t)$, where $\phi(\ )$ is some predefined function,

$$K_{t+1}^1=\tanh[\mu(aK_t^1+\phi(r_t))]-\tanh[\mu bK_t^1]. \tag{5}$$

The receiver's key, K_t^2, is generated using an identical oscillator with $\phi(r_t^{est})$, where r_t^{est} is the estimate of r_t at the receiver, given the received value, $\hat{r}_t$. Since the keys are generated from intermittently chaotic oscillators, they are aperiodic.

The synchronizing signal, r_t, can take many forms, but we focus on a random telegraph signal (RTS) defined by

$$r_t=\begin{cases} \alpha: & \text{with probability } p, \\ \beta: & \text{with probability } q=1-p, \end{cases} \tag{6}$$

with $0<\alpha<\beta$. This type of signal is preferred over others because of its inherent noise rejection attributes. The transformation, $\phi(\ )$, can also take many forms, and we have previously reported that a variety of noiselike signals can synchronize oscillators.[15,16] However, synchronization is particularly reliable and simple to analyze when the input to the oscillators is also a RTS. Accordingly, we define $\phi(r_t)$ as

$$\phi(r_t)=\begin{cases} A: & \text{when } r_t=\alpha, \\ B: & \text{when } r_t=\beta, \end{cases} \tag{7}$$

where A and B are pre-specified constants. The benefit of going from $\alpha-\beta$ to $A-B$ is that an intruder can have access to r_t without knowing the actual input used to generate the keys.

A. Noise immunity

If the sender and receiver key generators are synchronized, i.e., $K_t^2=K_t^1$, the message is recovered perfectly, so choosing A and B to ensure synchronization gives an ideal performance in the absence of channel noise. When noise is present in the received ciphertext, $\hat{s}_t$, its effect depends on the encryption/decryption algorithm. While this issue is not central to this paper, and our purpose is not to investigate any particular cipher, we plot the numerically determined effect of message channel noise on the mean decoding error, $\langle|\hat{m}_t-m_t|\rangle/\sigma_m$, in Fig. 3. Here, σ_m is the standard deviation of the original (unencrypted) message, and is used to scale the error.

The presence of noise in the r_t signal has a direct bearing on synchronization, and is, therefore, central to the method we propose. We have shown previously that the addition of uniform, zero-mean noise to an RTS-type synchronizing signal causes a synchronization error that is linear in the amplitude of the noise.[15] However, this might not be acceptable, since it can lead to a complete garbling of the message. Fortunately, the random telegraph signal, because of its digital nature, has an inherent degree of noise rejection. Thus, suppose that the sender and receiver have decided on the values of α and β in advance (these can be seen as system parameters like a, b, and μ). Then a very simple threshold filtering operation by the receiver can remove all but the most egregious additive noise from r_t. The thresholding operation is defined by

$$r_t^{est}=\begin{cases} \alpha: & \text{if } \hat{r}_t<\theta, \\ \beta: & \text{if } \hat{r}_t\geqslant\theta. \end{cases} \tag{8}$$

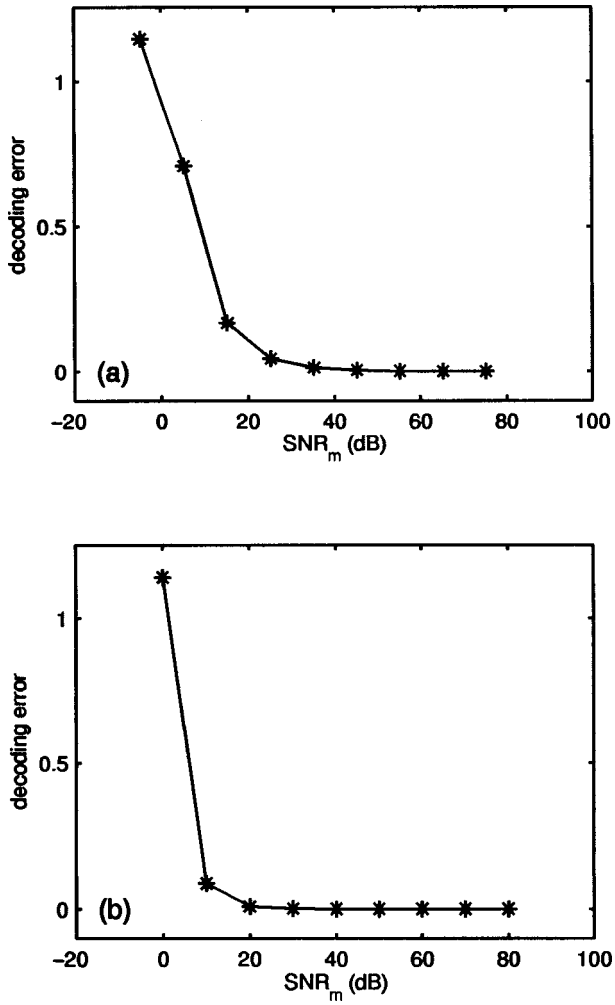

FIG. 3. The graphs show the dependence of the mean relative decoding error, $\langle|\hat{m}_t - m_t|\rangle/\sigma_m$, on δ_t, the additive noise in the message channel. Graph (a) is for 0-mean Gaussian noise with variance σ^2, and graph (b) for uniform noise between $\pm c$. The message, m_t, is a random signal distributed uniformly between 0 and 1. SNR_m is the signal-to-noise ratio of the received signal after addition of channel noise. These results show the noise response of the shift cipher with the uniform random message signal. Other ciphers and messages would show a different response. The parameters for r_t are $\alpha = A = 0.01$, $\beta = B = 0.2$, and $p = 0.3$.

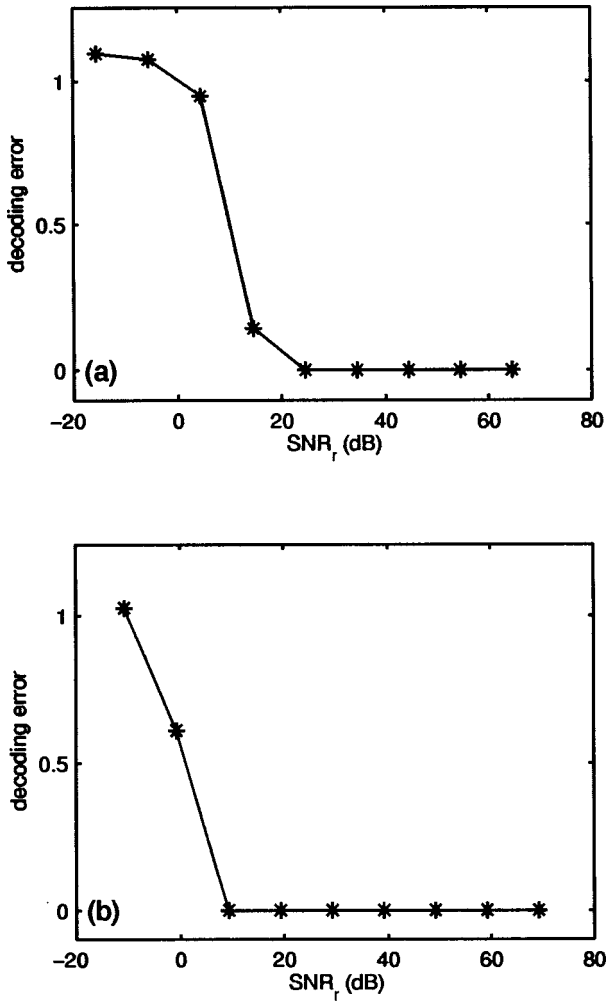

FIG. 4. The graphs show the dependence of the mean relative decoding error, $\langle|\hat{m}_t - m_t|\rangle/\sigma_m$, on η_t, the additive noise in the synchronizing signal, r_t. Graph (a) is for 0-mean Gaussian noise with variance σ^2, and graph (b) for uniform noise between $\pm c$. As in Fig. 3, the message values are random with a uniform distribution between 0 and 1. The parameters for the synchronizing signal are $\alpha = A = 0.01$, $\beta = B = 0.2$, and $p = 0.3$, which give $\langle r_t\rangle = 0.143$ and $\langle r^2\rangle = 2.803 \times 10^{-2}$. SNR_r is the signal-to-noise ratio for the synchronizing signal at the receiver before the thresholding operation. Note that the error reaches 0 at $SNR_r = 25$ dB for Gaussian noise and $SNR_r = 10$ dB for uniform noise.

The choice of threshold, θ, would be dictated by the distribution of the noise, η_t, if this is known. A good default value is

$$\theta = \alpha + (\alpha + \beta)/2, \tag{9}$$

which essentially assumes that η_n is zero-mean and symmetrically distributed. Figure 4(a) shows the error in a recovered plaintext message for a situation where $\eta_t \sim N(0,c)$, and Fig. 4(b) when $\eta_t \sim U[-c,+c]$. For the parameter values used here, the relative decoding error is essentially zero if the SNR for the synchronizing signal is above 25 dB for Gaussian channel noise and above 10 dB for uniform channel noise.

B. Security

The major issue in any secure communication system is whether it is actually secure. In this section, we look at the system we have described from this point of view.

It is clear from the description above that the security of our system depends entirely on the secrecy of system parameters. These are given by the *secure set*: $\mathcal{S} \equiv \{\mu, a, b, A, B, \theta, n, w\}$. Parameter p, while essential to system performance, plays no role in the actual regeneration of the key at the receiver, and can be obtained readily in any case from r_t, which is public. Parameters α and β too are easily obtained from r_t, and so are not part of the secure set. In contrast, θ is used privately by the users, and its secrecy can make a difference in decoding the message. We include the cipher parameters, n and w, in the secure set, but it should be noted that these are not essential to our method.

Other ciphers will have different parameters. We are concerned mainly with the key generation and key security aspects of the process, not the actual cipher used. With this in mind, we designate the set $\mathcal{E} \equiv \{\mu, a, b, A, B, \theta\}$ as the *essential secure set* (ESS), and focus only on the security of the parameters in this set.

The issue of security can now be reduced to two very specific questions

(1) *How accurately must the parameters in $\mathcal{E}$ be known to decode messages?*
(2) *Can an intruder estimate the parameters in $\mathcal{E}$ with accuracy sufficient to decode a message?*

To answer these questions, we numerically evaluate the sensitivity of the decoder with respect to each parameter in $\mathcal{E}$, and analyze the possibility of parameter estimation from intercepted data.

C. System security analysis

In this sub-section, we briefly analyze the ease or difficulty of estimating secure parameters once an intruder has intercepted some specific piece of data. The possible items of data include (1) key fragments; (2) plaintext fragments; (3) ciphertext fragments; and (4) synchronizing signal fragments. We make the following assumptions for our analysis.

(i) The ciphertext and the synchronizing signal are available to the intruder in their entirety since they are public-domain.
(ii) The intruder does not, *a priori*, have access to any parameters in the ESS.
(iii) The intruder *does* know the functional form of the key generator.

To acquire the ability to decode arbitrary messages, the intruder must be able to generate the key from the synchronizing input, r_t. Note that this is only a necessary condition, not a sufficient one, since access to the key does not break the shift cipher—or whatever other cipher is being used. Here, we are concerned primarily with the security of the key generator.

It is clear that plaintext and/or ciphertext fragments are of no use in breaking the key generator, except possibly as a means of obtaining a key fragment. The question, therefore, is the following: *How secure is the key generator if the intruder has access to a key fragment, K_t, and the corresponding synchronizing signal, r_t, $t_1 \leq t \leq t_2$?*

By plotting K_{t+1} against K_t, the intruder will readily obtain two 1-step return maps, F_A corresponding to the case $\phi(r_t) = A$, and F_B corresponding to $\phi(r_t) = B$ (Fig. 5), though she does not, of course, know the values of A and B. Since successive key values are generated recursively, the intruder must obtain the map $F: K_t \rightarrow K_{t+1}$ to obtain the key. This can be done in two ways: (1) Fit the map reconstructed from the intercepted data using an approximation technique, such as neural networks or splines; or (2) estimate the correct ESS parameters for the actual key generator, whose functional form is known. In either case, the map must be determined *extremely* accurately, since errors can grow exponentially with time (the maps are chaotic part of the time with positive Lyapunov exponents).

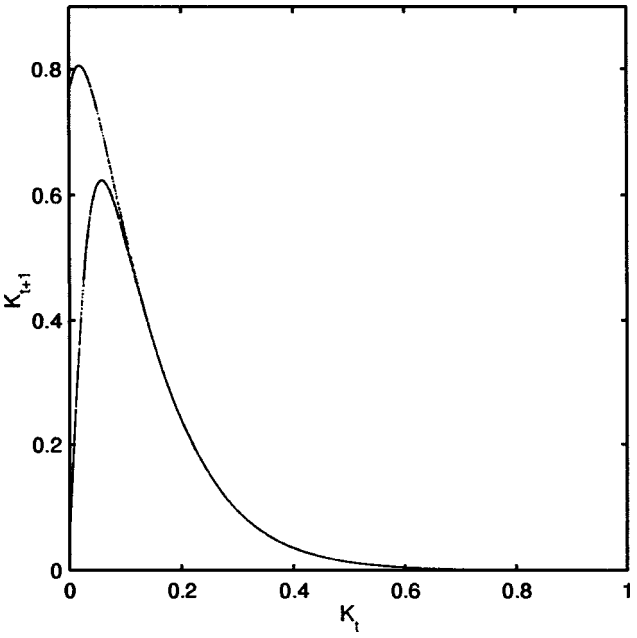

FIG. 5. The graph shows the 1-step return map reconstructed from an intercepted key fragment. The length of the fragment is 9,000 and the system parameters are $\mu = 5.0$, $a = 5.0$, $b = 5.0$, $A = \alpha = 0.01$, $B = \beta = 0.2$, $\theta = 0.1150$, $p = 0.3$, $n = 71$, $w = 1$. The message is a period 4 sinusoid with amplitude 0.8 (see the text). The maps for the A and B inputs are clearly visible.

To evaluate how accurately an intruder must reconstruct the key, we simulate decoding using a key with random errors. The correct key is perturbed by a 0-mean uniform random process of amplitude $\pm\delta$. The signal is the sinusoid $m_t = 0.8 \sin(2\pi t/4)$, so that $\langle |m_t|^2 \rangle = 0.32$, encrypted using a shift cipher with $n = 71$, $w = 1$. The results, shown in Fig. 6(a), indicate that, even for $\delta \sim 10^{-8}$, the relative decoding error is of the order of signal power. Clearly, generating the key from an approximated map is not a viable option in this case, since there will always be some error. In general, however, the sensitivity of decoding to errors in the key will depend on the choice of cipher. If the cipher is adequately sensitive to the key, map approximation will not be feasible.

Figures 6(b)–6(d) show the effect of errors in the estimation of μ, a, and b on the decoding error for the sinusoidal signal. Again, an estimation error of as little as 10^{-8} produces a decoding error of the order of signal power. Clearly, estimating ESS parameters would work *only* if the parameters were estimated essentially without error.

There is some information about the ESS parameters in the reconstructed maps. For example, the y-intercepts of the maps for inputs A and B are given by $K_A = \tanh(\mu A)$ and $K_B = \tanh(\mu B)$, respectively. Also, the tails of both maps are quite well fit by $1 - \tanh(\mu b x)$ if μa is large. However, recall that, to be useful, μ, a, and b must be determined with accuracy far exceeding 10^{-8}, which puts a similarly unrealistic requirement of accuracy on estimating the intercepts and fitting the maps.

The remaining parameter in the secure set, θ, would be very difficult to estimate even by correlating r_t values with

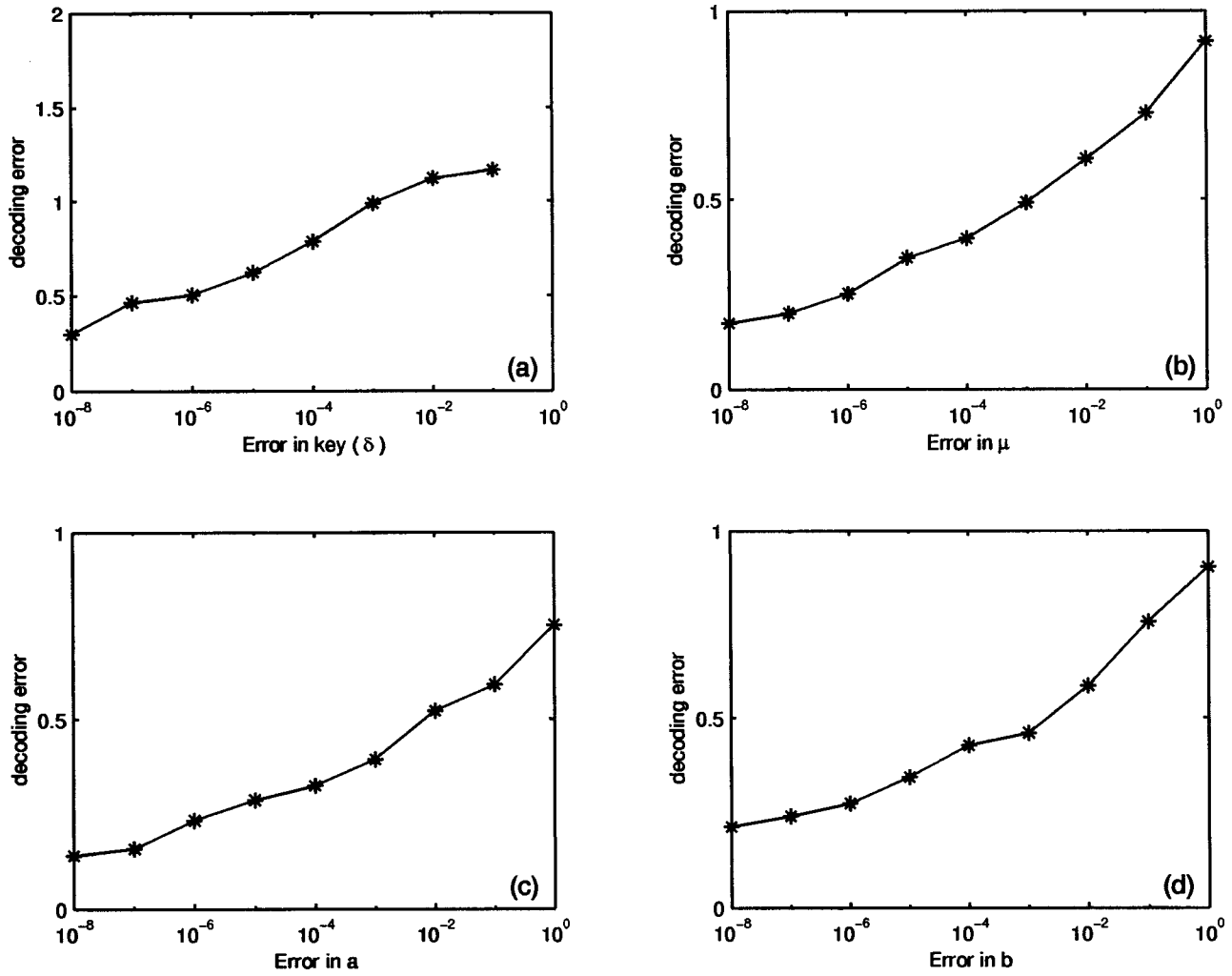

FIG. 6. Graph (a) shows the effect of noise in the key on the mean relative decoding error. The noise is uniform between $\pm\delta$. Graphs (b)–(d) show the dependence of the mean relative decoding error on a fixed error in parameters μ, a, and b, respectively. In all cases, it is clear that even exceedingly small errors lead to very large decoding errors. The message signal is the sinuosoid used in Fig. 5, and the error is in units of the signal RMS power.

the reconstructed return maps. While the intruder could determine when r_t^{est} was being set to α or β, the actual threshold would be impossible to find unless η_t was strong enough to cause many threshold crossings. In that case, the system would not be functional anyway. It should also be noted, however, that estimating θ accurately is of little value to the intruder—an approximate estimate would work almost as well in normal circumstances.

Thus, we conclude that the only way the intruder could obtain the ESS parameters is to somehow steal their exact values, or to steal a long key fragment and fit it extremely accurately. Given that the key is never actually transmitted, this would be quite a difficult thing. The system, therefore, is quite secure—if its users have the ability to set parameters very, very accurately!

D. Enhancing key security

It should be noted that we have based our description on the simplest case of a single map key-generator. While this produces a secure key, the simplicity of the key might make it vulnerable to a sophisticated attack using methods based on delay-coordinate embedding.[26–28] This is particularly true because the keys are only intermittently chaotic, providing at least a remote possibility of correct parameter estimation by an intruder. However, it is very easy to use our approach to produce keys which, even if intercepted, are much less amenable to reconstruction.

One simple method for enhancing key security is to use identical sets of several oscillators in both the transmitter and the receiver. The parameters of the oscillators within the set are different, and are chosen to produce the required chaos-periodicity transition in response to the synchronizing signal. Corresponding oscillators in the transmitter and receiver are, of course, identical. The key is generated by taking the maximum value of all oscillator outputs in the set at each time step:

$$z_{t+1}^k = \tanh[\mu^k(a^k z_t^k + \phi(r_{t+1}))] - \tanh[\mu^k b^k z_t^k], \quad (10)$$

$$K_{t+1} = \mathrm{MAX}_k[z_{t+1}^k], \quad (11)$$

where k indexes different oscillators, each driven by the same input. With as few as four oscillators, the key produced is so complex that reconstruction of the underlying maps from an intercepted key segment appears to be very difficult.

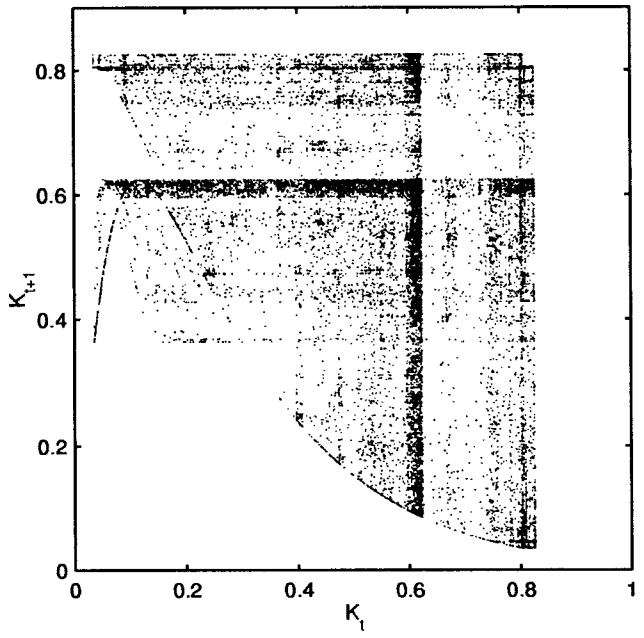

FIG. 7. The graph shows the 1-step return map reconstructed from a key fragment produced by a set of of four oscillators. The oscillators are driven by the same input, $\phi(r_t)$, and the key value at step t is taken to be the maximal output from the oscillator set. The oscillators are 5/5/1, 5/2.5/0.5, 5.5/5/1, and 4.5/5/1. This figure should be compared with Fig. 5. Clearly, no 1-dimensional return map can be recovered from this data. A higher-dimensional reconstruction using delay-coordinate embedding might yield a map, but not the precise key generator parameters, nor the exact key values needed to break the code.

Figure 7 shows the return map from a long key fragment produced using a set of four different oscillators. The enormity of the problem faced by the intruder is quite clear from the figure, which shows that the key has the structure of high-dimensional noise. Careful use of more oscillators could make the key even more complex with essentially no loss of noise-immunity. However, the best combination of oscillators to use remains an open issue. Multiple maps with different parameter values could be combined in many other series, parallel, and feedback architectures to produce even more sophisticated key generators.[29] Such key generators would be high-dimensional nonlinear chaotic dynamical systems. An intruder in possession of a key fragment would have to reconstruct their dynamics using delay-coordinate embedding.[30] Such methods have been used successfully to break simple chaotic masking schemes,[26–28] but are unlikely to work with sufficient accuracy in the case of high-dimensional, noiselike keys with relatively little structure in the phase-space dynamics.[31]

IV. CONCLUSION

In conclusion, we have presented an interesting scheme whereby a common noise source can be used to broadcast secure information over a public channel. The scheme is based on remote key generation via noise-induced synchronization between transmitter and receiver oscillators. Our method illustrates how noise interacts with chaos to produce useful behavior which would not be possible in the absence of chaos. This has broad implications for the potential utility of chaos in natural and artificial systems.

ACKNOWLEDGMENTS

The authors would like to thank Xin Wang, Mingzhou Ding, Chai-Wah Wu, Tao Yang, Kevin Short, and Tom Carroll for providing reprints of their papers, and two anonymous reviewers for their constructive suggestions.

[1] K. M. Cuomo and A. V. Oppenheim, ''Circuit implementation of synchronized chaos with applications to communications,'' Phys. Rev. Lett. **71**, 65–68 (1993).
[2] K. M. Cuomo, A. V. Oppenheim, and S. H. Strogatz, ''Synchronization of Lorenz-based chaotic circuits with applications to communications,'' IEEE Trans. Circuits Syst., II: Analog Digital Signal Process. **40**, 626–633 (1993).
[3] C. W. Wu and L. O. Chua, ''A simple way to synchronize chaotic systems with applications to secure communication systems,'' Int. J. Bifurcation Chaos Appl. Sci. Eng. **3**, 1619–1627 (1993).
[4] L. Kocarev and U. Parlitz, ''General approach for chaotic synchronization with applications to communications,'' Phys. Rev. Lett. **74**, 5028–5031 (1995).
[5] U. Parlitz, L. Kocarev, T. Stojanovski, and H. Preckel, ''Encoding messages using chaotic synchronization,'' Phys. Rev. E **53**, 4351–4361 (1996).
[6] H. Dedieu, M. P. Kennedy, and M. Hasler, ''Chaos shift keying: Modulation and demodulation of a chaotic carrier using self-synchronizing chua's circuits,'' IEEE Trans. Circuits Syst., II: Analog Digital Signal Process. **40**, 635–642 (1993).
[7] T. Yang and L. O. Chua, ''Secure communication via chaotic parameter modulation,'' IEEE Trans. Circuits Syst., I: Fundam. Theory Appl. **43**, 817–819 (1996).
[8] S. Hayes, C. Grebogi, and E. Ott, ''Communicating with chaos,'' Phys. Rev. Lett. **70**, 3031–3034 (1993).
[9] S. Hayes, C. Grebogi, E. Ott, and A. Mark, ''Experimental control of chaos for communication,'' Phys. Rev. Lett. **73**, 1781–1784 (1994).
[10] D. R. Frey, ''Chaotic signal encoding: An approach to secure communication,'' IEEE Trans. Circuits Syst., II: Analog Digital Signal Process. **40**, 660–666 (1993).
[11] H. D. I. Abarbanel and P. S. Linsay, ''Secure communications and unstable periodic orbits of strange attractors,'' IEEE Trans. Circuits Syst., II: Analog Digital Signal Process. **40**, 643–645 (1993).
[12] T. Yang and L. O. Chua, ''Channel-independent chaotic secure communication,'' Int. J. Bifurcation Chaos Appl. Sci. Eng. **6**, 2653–2660 (1996).
[13] L. M. Pecora and T. L. Carroll, ''Synchronization in chaotic systems,'' Phys. Rev. Lett. **64**, 821–824 (1990).
[14] L. M. Pecora and T. L. Carroll, ''Driving systems with chaotic signals,'' Phys. Rev. A **44**, 2374–2383 (1991).
[15] A. A. Minai and T. Anand, ''Chaos-induced synchronization in discrete-time oscillators driven by a random input,'' Phys. Rev. E **57**, 1559–1562 (1998).
[16] A. A. Minai and T. Anand, ''Using noise to synchronize chaotic neural oscillators,'' *Proc IJCNN 98*, 1466–1471 (1998).
[17] T. Yang, C. W. Wu, and L. O. Chua, ''Cryptography based on chaotic systems,'' IEEE Trans. Circuits Syst., I: Fundam. Theory Appl. **44**, 469–472 (1997).
[18] D. E. R. Denning, *Cryptography and Data Security* (Addison-Wesley, Reading, MA, 1983).
[19] R. Beckett, *Introduction to Cryptology* (Blackwell, Oxford, UK, 1988).
[20] X. Wang, ''Period-doublings to chaos in a simple neural network: An anaytic proof,'' Complex Syst. **5**, 425–411 (1991).
[21] X. Wang, ''Discrete-time neural networks as dynamical systems,'' Ph.D. thesis, University of Southern California, 1992.
[22] A. A. Minai and T. Anand, ''Stimulus-induced bifurcations in discrete-time neural oscillators,'' to appear in Biol. Cybern.
[23] L. Yu, E. Ott, and Q. Chen, ''Transition to chaos for random dynamical systems,'' Phys. Rev. Lett. **65**, 2935–2948 (1990).
[24] L. Yu, E. Ott, and Q. Chen, ''Fractal distribution of floaters on a fluid surface and the transition to chaos for random maps,'' Physica D **53**, 102–124 (1991).

[25] A. A. Minai, ''Synchronizing non-homogeneous arrays of chaotic maps with a random scalar coupling'' (submitted for publication).
[26] K. M. Short, ''Steps toward unmasking secure communications,'' Int. J. Bifurcation Chaos Appl. Sci. Eng. **4**, 959–977 (1994).
[27] G. Peréz and H. A. Cerdeira, ''Extracting messages masked by chaos,'' Phys. Rev. Lett. **74**, 1970–1973 (1995).
[28] K. M. Short, ''Unmasking a modulated chaotic communications scheme,'' Int. J. Bifurcation Chaos Appl. Sci. Eng. **6**, 367–375 (1996).
[29] A. A. Minai and T. Anand, ''Synchronizing multiple chaotic maps with a randomized scalar coupling'' (submitted for publication).
[30] D. Kaplan and L. Glass, *Understanding Nonlinear Dynamics* (Springer-Verlag, New York, 1995).
[31] T. L. Carroll and L. M. Pecora, ''Synchronizing hyperchaotic volume preserving maps and circuits,'' to appear in IEEE Trans. Circuits. Syst. I. Fundam. Theory Appl.