Cryptosystems based on an analog of heat flow

G. R. Blakley
William Rundell

Department of Mathematics
Texas A&M University
College Station, Texas 77843-3368

CONTENTS

C. Pomerance (Ed.): Advances in Cryptology - CRYPTO '87, LNCS 293, pp. 306-329, 1988.

1. Introduction.

It is commonplace to base computer security and information security on hard problems. Recent cryptosystems have been based on the knapsack problem [DE83, pp. 118-126; BR85] and the problem of factoring an integer [DE83, pp. 104-109]. The former problem is NP complete [GA79, p. 247]. The place of the latter problem in complexity theory is not well understood, but it has been around in number theory for a long time.

We will base a family of discrete cryptosystems and continuous scramblers on ideas related to ill-posed problems in differential equations. Once again, complexity theory has little to say about the difficulty of such problems. But they have been around in analysis for a long time.

Our approach will be to proceed from a toy system through a series of more realistic systems, taking up various considerations of decodability, bandwidth expansion, intersymbol dependence, computational expense, and security as they arise. One feature of this approach which will be evident from the outset is its neutrality between the discrete and the continuous realms in communication security. In this respect it is much in the spirit of [BL85, BL86, BL87], which treat various information-theoretic objects in a way designed to minimize dependency on any finiteness properties they may have. More specifically, it is an approach to cryptosystem design closely akin to that recently taken by Davida, Gilbertson and Walker [DA86].

The paper begins by examining partial differential equations (PDEs) suggested by the time evolution of the distribution of temperature in a rod. The plaintext is the initial conditions (i.e. the temperature profile u at $t = 0$), and the cryptext is the $t = 1$ temperature profile. The key is, in a sense to be made clearer below, largely the PDE itself. The role of the boundary conditions is important too. In the end we will not confine ourselves to Dirichlet (fixing a solution u at the end points of the rod) or Neumann (fixing u_x at the end points of the rod) problems. But for a while we will restrict our consideration to the Dirichlet conditions:

$$u(0,t) = u(1,t) = 0 \qquad \text{for all } t \geq 0\,.$$

It is important to note at the outset that security dictates the introduction of nonlinearities at many points. In our particular case this entails the use of nonlinear partial differential operators. Clarity of exposition, on the other hand, calls for casting the discussion herein in linear terms whenever possible. This engages the reader's intuition more readily, and enables us to bring a larger variety of theoretical

results to bear on the subject matter. The inevitable compromise this entails should not mislead anyone. Any good implementation of these ideas will be nonlinear through and through, despite any linear biases in the exposition below.

2. Plaintext and encoded text as related time slices through a solution of a PDE.

Throughout this section we largely ignore secrecy considerations, and simply speak of encoding and decoding a message f. By a message we mean a member f of C^D, i.e. a function

$$f : D \to C$$

whose domain D and codomain C have some useful structure, usually group-theoretic. In cryptography and error control, D is often a finite set of symbols. Thus the message

HELLO

is the function

$$f : \{1, 2, 3, 4, 5\} \to \{A, B, \ldots, Z\}$$

defined by setting

$$f(1) = \mathrm{H}$$
$$f(2) = \mathrm{E}$$
$$f(3) = \mathrm{L}$$
$$f(4) = \mathrm{L}$$
$$f(5) = \mathrm{O}.$$

An encode/decode pair (c, d) is a pair of functions

$$c : C^D \to E$$
$$d : E \to C^D$$

such that $d(c(f)) = f$, or at least such that $d(c(f))$ is usually fairly near f. E is some appropriate set of encoded messages. Often encoded messages look just like plaintext messages. In this important case, of course, we have

$$E = C^D.$$

Consider an encode map

$$C : R^{[0,1]} \to R^{[0,1]}$$

defined as follows. Take a function $u = u(x,t)$ whose domain is the semi-infinite strip $[0,1] \times [0,\infty)$ and let the plaintext f be the initial condition

$$u(x,0) = f(x) \qquad x \in [0,1] \tag{2.1}$$

for the classical heat equation

$$u_t - u_{xx} = 0 \tag{2.2}$$

subject to the boundary conditions

$$u(0,t) = u(1,t) = 0 \qquad t \geq 0\,. \tag{2.3}$$

Now let u be the unique solution of (2.1), (2.2), (2.3). Then the encoded text g corresponding to the plaintext f is just the restriction of the solution u to the set $\{(x,t) : t = 1\}$, i.e.

$$g(x) = u(x,1) \qquad x \in [0,1]\,.$$

Somebody who would rather encode a discrete message than a continuous one can use the same technique after source coding in some appropriate manner. One simple system is to use ASCII code, or some similar code, or to assign integer values to symbols. It is necessary to decide on a block length N, and to partition $[0,1]$ into N subintervals. After that, the plaintext is a function whose value on the j^{th} subinterval is the integer corresponding to the j^{th} symbol in the block. For example, one way to source code is by the correspondence

$$\left.\begin{array}{r} \text{blank} \leftrightarrow 0 \\ \mathrm{A} \leftrightarrow 1 \\ \mathrm{B} \leftrightarrow 2 \\ \vdots \\ \mathrm{Z} \leftrightarrow 26. \end{array}\right\} \tag{2.4}$$

Then it is possible to choose block size $N = 8$. The plaintext message HELLO can be represented by the piecewise continuous function $f : [0,1] \leftrightarrow \{0,1,2,\ldots,26\}$ such that

$$\begin{aligned} f(x) &= 0, \quad x \in [0,1/8) \\ &= 8, \quad x \in [1/8,2/8) \\ &= 5, \quad x \in [2/8,3/8) \\ &= 12, \quad x \in [3/8,5/8) \\ &= 15, \quad x \in [5/8,6/8) \\ &= 0, \quad x \in [6/8,1] \end{aligned}$$

Here the eight subintervals have been chosen to be of equal length, 1/8. This is not necessary. Also, $f(x)$ has been modified to satisfy the homogeneous boundary conditions $f(0) = f(1) = 0$.

So, at this point there is a way to represent a (continuous or discrete) plaintext message as a function

$$u(x, 0) = f(x)$$

defined on the closed unit interval [0,1] and to produce a corresponding (continuous) encoded message

$$u(x, 1) = g(x) .$$

3. Intersymbol dependence, smearing, and diffusion in a one-way function based on the heat equation

The encoding of the previous section has the feature that small changes in the plaintext message f cause changes in the encoded message g which are spread throughout g. There is, in the terminology of classical cryptography, "strong intersymbol dependence". A mathematical term for this is "smearing", and it is referred to physically as "diffusion". We will use whatever one of these expressions seems appropriate to the context below. There are provable lower bounds on the extent of this intersymbol dependence, a point to which we shall return in Section 10 below.

Parabolic PDEs such as (2.2) or (2.4) obey the strong maximum principle [RU76]. Thus they enable a cryptosystem designer to produce provable, and sometimes very precise, estimates of diffusion (intersymbol dependence). The physical idea behind this is straightforward, though unrealistic, since energy would have to be transmitted faster than light. In elementary texts [BO77, pp. 511-515] students read about the ordinary heat equation

$$u_t - u_{xx} = 0. \tag{3.1}$$

They discover that it has the feature that, when a torch is applied to a point x_0 on a cold rod at time $t = 0$, every point of the rod is warm [BO77, p. 489] (some warmer than others) at every subsequent time ($t > 0$). The same infinitely fast heat propagation property characterizes generalizations such as

$$u_t - u_{xx} + a(x)u = 0 \tag{3.2}$$

Mathematically, this says that if $f(x) \geq 0$ for $x \in [0,1]$, and is nontrivial then $g(x) > 0$ for $x \in (0,1)$. Suppose that f is zero, except in some small neighborhood $(x_0 - \epsilon,\ x_0 + \epsilon)$ of the point x_0 (Cryptographically this f could correspond to the difference between two plaintext messages which agreed everywhere but at a single symbol position, say the i^{th} symbol in a block). For the heat equation (3.1) the maximum value of $u(x,t)$ for $x \in [0,1]$ will continue to occur at x_0 for all positive t. However the introduction of the function $a = a(x)$ can change the location of this maximum.

Strong intersymbol dependence is not, by itself, sufficient to make an encoding process cryptographically strong. Error control codes also have this feature, but possess little cryptographic strength. And, in fact, the encoding above is not a cryptosystem. It is a fixed process relying on a known equation, the heat equation. There is no secret key material which a sender and a receiver can share before they begin using this encoding as the basis of a secure communications system.

But is the encoding process of this section a one-way function? It would appear to be. There is no known feasible way to turn the process around, so as to proceed from knowledge of an encoded message g to produce the plaintext message f which gave rise to g. The heat equation does not "reverse in time". We will expand on this point toward the end of the next section.

At this point we have the rudiments of a possible alternative to Purdy's [PU 74] high security login scheme.

4. Introducing key dependence into the encoding process. Ill-posed problems.

We will now assume that the sender and receiver get together in secret to modify the heat equation as the basis of an encoding process. Suppose they replace it by the equation

$$u_t - u_{xx} + au = 0, \tag{4.1}$$

where $a = a(x)$ is a function of position only and, as before, $u = u(x,t)$ is a function of position and time. The function $a = a(x)$ is a secret, shared only by the designated encoder (i.e. sender) and the designated decoder (i.e. receiver) who plan to use this encoding process as the basis of a true cryptosystem. Physically, $a\,u = a(x)u(x,t)$ is a reaction term proportional to the value of u. Here we have a position-dependent (but time-independent) proportionality constant a. The function a plays

the role of a secret cryptographic key.

Somebody who wants the key material more thoroughly mixed into the encryption process can go beyond (4.1) to

$$\sigma u_t = (cu_x)_x - au \tag{4.2}$$

Here $\sigma = \sigma(x)$ is a function of position only, which corresponds to specific heat. Also $c = c(x)$ is a function of position only, which corresponds to conductivity. Such functions c and a will clearly influence the temperature distributions $u(x,t)$ subsequent to the initial condition

$$u(x,0) = f(x),$$

and cause them to differ from what would be observed in a homogeneous rod with the same initial condition. An obvious change of variable leads to recasting (4.2) in the form

$$u_t = Lu \tag{4.3}$$

where L is the linear operator defined by setting

$$Lu = (pu_x)_x - qu \tag{4.4}$$

Here $p = p(x)$ and $q = q(x)$ are functions of position only, and $p(x)$ is strictly positive for every $x \in [0,1]$.

The introduction of key material moves the original heat-equation-encoding process over toward the cryptographic realm. But it is still deficient in one respect. All the PDEs (2.2), (3.2), (4.2), and (4.4) are parabolic. Consequently the problem of decoding remains ill-posed. In fact we are faced, at this point, with the classical backwards heat flow problem. The backwards parabolic equation is perhaps the canonical example of an ill-posed problem in PDEs [PA74]. J. Hadamard [HA23] initiated the custom of calling a problem *well-posed* if it satisfied the three criteria:

(1) There exists a solution to the problem;

(2) The solution is unique;

(3) The solution depends continuously on the boundary data.

A problem is *ill-posed*, in Hadamard's terminology, if it is not well-posed.

Let us look at the Fourier expansion

$$u(x,t) = \Sigma a_n e^{-n^2\pi^2 t} \sin(n\pi x)$$

of the solution of the problem (2.1) -(2.3), where

$$f(x) \sim \Sigma a_n \sin(n\pi x).$$

Evidently $u(x,t)$ is analytic in x for any $t > 0$. If there is to be any hope of recovering even an $L^2[0,1]$ function from a cryptext

$$g(x) = u(x,1),$$

it must be true a priori that g is analytic (and its Fourier coefficients must decay exponentially fast to zero). Worse still, the sequence

$$u_n(x,t) = (1/n^3)\sin(n\pi x)e^{(1-t)n^2\pi^2}$$

$(n = 1,2,3,...)$ satisfies (2.1) and (2.2), as well as (2.3) with

$$f_n(x) = (1/n^3)e^{n^2\pi^2}\sin(n\pi x).$$

The functions $g_n(x)$ are in this case

$$g_n(x) = (1/n^3)\sin(n\pi x).$$

They tend uniformly to zero in any norm one might conceivably impose on the problem. On the other hand the sequence $\{f_n\}$ cannot converge, in any reasonable norm. In consequence no error, no matter how small, in the codetext can be guaranteed to produce a bounded error in the decoded approximation to the plaintext, even if it were possible to produce such a decoded approximation.

Is there an alternative heat flow model which possesses the main features (certainly including the maximum principle) of equation (4.3) and yet allows time to progress in both directions? The answer is yes, and leads to our next change of the driving equation. This time there is a model which makes decoding, as well as encoding, possible.

5. Decoding an encoded message g to recover an approximation to the plaintext message f.

Consider the pseudoparabolic [SH70a] PDE

$$(L - I)u_t + Lu = 0 \tag{5.1}$$

where I is the identity map and L is an appropriate differential operator. One example might be the linear operator L defined in (4.4). A designated decoder knowing L, the boundary conditions (2.3), and the encoded message

$$g(x) = u(x,1) \tag{5.2}$$

can hope to recover a good approximation to the plaintext message

$$f(x) = u(x,0) \tag{5.3}$$

which represents the initial condition (2.1).

Pseudoparabolic PDEs have received much study in the last 15 years. They arise in a variety of problems where it is appropriate to add a certain type of correction term to a parabolic PDE. Examples are problems dealing with second order fluids, with the seepage of fluid through fissured rock, or with certain two-temperature theories of thermodynamics. Perhaps the best reference to the general features of these equations is the paper [SH70a] of Showalter and Ting. See also [SH70b]. On any of the usual function spaces in which one would usually pose (5.1), the operator $L - I$ turns out to be invertible if $q(x) \geq -1$ on $[0,1]$. Examples of such spaces are the Sobolev space ${H_0}^1 \cap H^2$ or the Schauder space $C_0^{2+\alpha}$. On such spaces the operator $(L-I)^{-1}L$ has an extension to a bounded operator on all of L^2 or of $C^{0+\alpha}$. These considerations lead to the abstract differential equation

$$u_t + (L-I)^{-1}Lu = 0. \tag{5.4}$$

Coupled with the boundary condition (2.3) and the initial condition (2.1), it has a solution

$$u(x,t) = \exp(-t[L-I]^{-1}L)f(x) \tag{5.5}$$

where

$$\{\exp(-t[L-I]^{-1}L) : t \text{ is a real number }\}$$

is a group of operators generated by $[L-I]^{-1}L$ (see [SH70a; SH70b]). If

$$g(x) = u(x,1)$$

then

$$g = \exp(-(L-I)^{-1}L)f. \tag{5.6}$$

Hence f can be recovered from g by means of the relation

$$f = \exp((L - I)^{-1}L)g$$

The positivity of these two exponential transformations can be shown using the maximum principle for pseudoparabolic partial differential equations [RU76].

6. Computational procedures and costs in the linear case.

Linear and affine maps are abhorrent to cryptosystem designers. Nevertheless it is instructive to examine encryption and decryption calculations in the linear case to show how pseudoparabolic PDEs differ from parabolic, and to get a jumping-off place from which to examine the nonlinear case later. Let L be a linear ordinary differential operator involving only differentiation with respect to the position variable x, for example as in (4.4). Consider the linear pseudoparabolic partial differential equation

$$u_t + (L - I)^{-1}Lu = 0 \tag{6.1}$$

coupled with the initial condition (2.1).

The plaintext message

$$f(x) = u(x, 0) = \exp(0)f(x)$$

gives rise to the cryptext message

$$g(x) = u(x, 1) = \exp(-[L - I]^{-1}L)f(x)$$

and vice versa. Thus

$$\begin{aligned} f(x) &= If(x) \\ &= \exp([L - I]^{-1}L)\exp(-[L - I]^{-1}L)f(x) \\ &= \exp([L - I]^{-1}L)g(x). \end{aligned}$$

The most simple-minded way to structure the computation of g, given f, and the computation of f, given g, is the following. Under the transformation

$$v(x, t) = e^t u(x, t) \tag{6.2}$$

the equation (6.1) becomes

$$v_t + (L - I)^{-1}v = 0 \tag{6.3}$$

Suppose that $y(x)$ satisfies the ordinary differential equation

$$(L - I)y(x) = -h(x) \quad 0 < x < 1 \tag{6.4}$$

$$y(0) = y(1) = 0$$

then $y(x)$ can be written as

$$y = -(L - I)^{-1}h = -\int_0^1 G(x, s)h(s)\, ds \equiv \mathcal{G}h \tag{6.5}$$

where $G(x, s)$ is the Green's function for the equation (6.4). Noting that $v(x, 0) = f(x)$ and $g(x) = e^{-t}v(x, 1)$ we see that

$$g = e^{-1} \exp(-(L - I)^{-1})f \tag{6.6}$$

and since $\mathcal{G} = -(L - I)^{-1}$ is a bounded operator

$$g = e^{-1}(I + \mathcal{G} + \mathcal{G}^2/2! + \ldots + \mathcal{G}^n/n! + \ldots)f \tag{6.7}$$

Thus g can be computed by truncating the exponential series. The k^{th} term of this series consists of k consecutive integrations of the function $f(s)$ with the Green's function $G(x, s)$. (This last function of course depends on $p(x), q(x)$). It is this integration process that provides the "smearing" of the function $f(x)$.

In (5.1) the operator L need not be the same in both positions, we could equally well have chosen the equation

$$Mu_t + Lu = 0 \tag{6.8}$$

where M is of the same form as L. However if we wish to retain the maximum principle, and for our cryptosystem this is certainly the case, then M and L can only differ in their non-differentiated terms [RU79].

To further expose the action of our method we shall analyze a particularly simple case of (6.8). This example will be used again in later sections to demonstrate further properties. Choose M to be the differential operator $\frac{d^2}{dx^2}$ and L to be $\frac{d^2}{dx^2} + a(x)$, the domain being those $C^2[0, 1]$ functions that vanish at $x = 0$ and $x = 1$. The function $u(x, t)$ will thus satisfy

$$u_{xxt} + u_{xx} + a(x)u = 0 \tag{6.9}$$

with

$$u(x,0) = f(x), \quad u(0,t) = u(1,t) = 0 \tag{6.10}$$

Again the change of variable $v = e^{-t}u$ will put the equation in a slightly easier form,

$$v_{xxt} + a(x)v = 0 \tag{6.11}$$

or in the abstract formulation

$$\begin{aligned} v_t + B^{-1}v &= 0 \\ v|_{t=0} &= f(x) \end{aligned} \tag{6.12}$$

where B denotes the operator $Bu = M^{-1}(a(x)u(x))$.

In this case (6.5) takes the form

$$\mathcal{G}h = -\int_0^1 G_0(x,s)a(s)h(s)ds \tag{6.13}$$

where G_0 is the Green's function for M, and is given by

$$G_0(x,s) = \begin{cases} (1-s)x & x \leq s \\ (1-x)s & x \geq s \end{cases}$$

In each successive application of $\mathcal{G}$ that is required to approximate $g(x)$ by (6.7), we see that the coefficient $a(x)$ (our chosen key) comes in to modify the input function by integration against $a(x)$.

The above analysis was presented only to show the workings of the encryption operator and its dependence on the coefficient $a(x)$. In practice one would not use a power series method, but rather a finite difference scheme based, for example, on the Crank-Nicholson method [YO73, pp. 1078, 1086-1088].

It must be emphasized that the decryption process is entirely the same as encryption

$$\begin{aligned} \text{codetext} &= \exp(-A)\ (\text{plaintext}) \\ \text{plaintext} &= \exp(A)\ (\text{codetext}) \end{aligned}$$

since both initial and final value problems are solved in the same manner for the equation (5.1).

7. Cryptosystems based on nonlinearities in pseudoparabolic PDEs.

If $p = 1$ in L as in (4.4), and the transformation $u \to ue^{-t}$ is performed on (5.1) then, as in section 6, we arrive at the simplest form of our equation

$$u_{xxt} - u_t + q(x)u = 0 \tag{7.1}$$

Even in this basic case it is not easy to find $q(x)$ from a knowledge of both $f(x)$ and $g(x)$. Problems of this kind are called undetermined coefficient problems and in most cases are notoriously ill-posed. There may be an infinite number of $q(x)$ that would yield the same $g(x)$ from a given $f(x)$. Even if this were not so, there may be two functions $q_1(x)$ and $q_2(x)$ that take a given plaintext onto very similar codetexts, yet for another plaintext, the corresponding codetexts would not be close. If $q = q(x,t)$ were allowed to be time-dependent then it would, at least in theory, be impossible to obtain this function of two variables by giving only one additional function g at the single variable x.

Equations of the form (6.6) are simply matrix equations of size N by N. Such equations could be solved for the eigenvectors of the matrix given a sufficient quantity of plaintext-codetext pairs. Even if it were not possible to recover the key, it might be possible to read the messages.

This leads to the final modification in our cryptosystem, the addition of some nonlinear terms. Equations of the type (4.3) or (5.1) are referred to as "diffusion equations". This terminology is based on one of their features. They spread initially localized heat throughout the body — the very property that gives us our intersymbol dependence. The addition of a reaction term F, if correctly chosen, can tend to counteract this diffusing tendency by further increasing the temperature at places where $u(x,t)$ is already large. These combination equations are referred to as "reaction-diffusion" equations. Our chosen type of equation reads

$$(L - I)u_t + Lu = F, \tag{7.2}$$

where the function F may depend on x, t, u and u_x. In (7.2) one could also assume that the coefficients of L depend on x, t, u, u_x and the maximum principle would still hold (under suitable restrictions on sign) as would the invertibility of (7.2) in time. The boundary conditions $u(0,t) = u(1,t) = 0$ could be generalized to conditions of the form $u_x + h(t)u = \beta(t)$ where h and β could form part of the key. They could also be made nonlinear.

8. Computational considerations regarding encryption and decryption in the nonlinear case.

How would one solve (7.2), how would the key enter, and what is the additional expense in computation?

Let us first consider a possible attack on the problem that uses the discussion in section 6 as a basis. If $v(x,t)$ satisfies

$$\begin{aligned}(L-I)v_t + Lv &= 0 \\ v(x,0) &= f(x)\end{aligned} \tag{8.1}$$

subject to the usual boundary conditions (2.3) at $x = 0$ and $x = 1$, then we can write the solution in abstract form

$$v = \exp(tA)f$$

with $A = (L-I)^{-1}L$. Of course if we know the coefficients of L we can construct the operator $\exp(-tA)$.

If $u(x,t)$ now satisfies (7.2) then

$$\begin{aligned}u_t + Au &= (L-I)^{-1}F[u] := \hat{F}(u) \\ u(x,0) &= f(x)\end{aligned} \tag{8.2}$$

where $\hat{F}$ depends on (x,t,u,u_x) and the solution to this can be written in the form

$$u(t) = \exp(-tA)f + \int_0^t \exp(-(t-\tau)A)\hat{F}(u(\tau))d\tau \tag{8.3}$$

We can consider (8.3) as a nonlinear integral equation for u whose free term is $\exp(-tA)f$ and whose kernel is $K(t,\tau,u) = \exp(-(t-\tau)A)\hat{F}$. The equation can be solved by the method of successive approximations under mild conditions on the function F (smoothness in x,t, Lipschitz continuity in the variables u, u_x). The approximation scheme starts with an initial guess $u_0(t)$ (usually one uses the free term; $\exp(-tA)f$ in this case) and then updates by

$$u_{n+1}(t) = \exp(-tA)f + \int_0^t \exp(-(t-\tau)A)\hat{F}(u_n(\tau))d\tau \tag{8.4}$$

for $n \geq 0$.

In practice one would not invoke the machinery in quite this form. A finite difference scheme would again be used and the nonlinear term would be evaluated

by successive approximations in a subloop. This subloop is usually quite short — typically four or five iterations suffice. For a given mesh size, the cost of adding nonlinearities is roughly a fixed amount (independent of the mesh size) times the cost of the linear case.

9. Resistance of the nonlinear pseudoparabolic PDE cryptosystem to chosen plaintext attack.

Even for the simple model problem (6.11) it is not known whether one can recover the key material $a(x)$ from a knowledge of $f(x)$ and $g(x)$. If, of course, enough message pairs are intercepted then eventually, since the problem is linear, it is possible for a cryptanalyst to find the action of the system on each element of a basis for the set of possible plaintexts. This could then be used to read subsequent messages. In the case of a nonlinear version of the system this method is no longer applicable. The possibility of recovering a function F of the form $F(x, t, u, u_x)$ from measurements of f, g pairs lies outside the scope of present research in the area of undetermined coefficient problems in partial differential equations at present. And the image of a vector space of plaintext messages will be, at best, a complicated manifold of cryptext messages. So even reading subsequent messages in the absence of key information appears to be a difficult problem.

10. Block size, its effect on security, the cost of encryption and error control

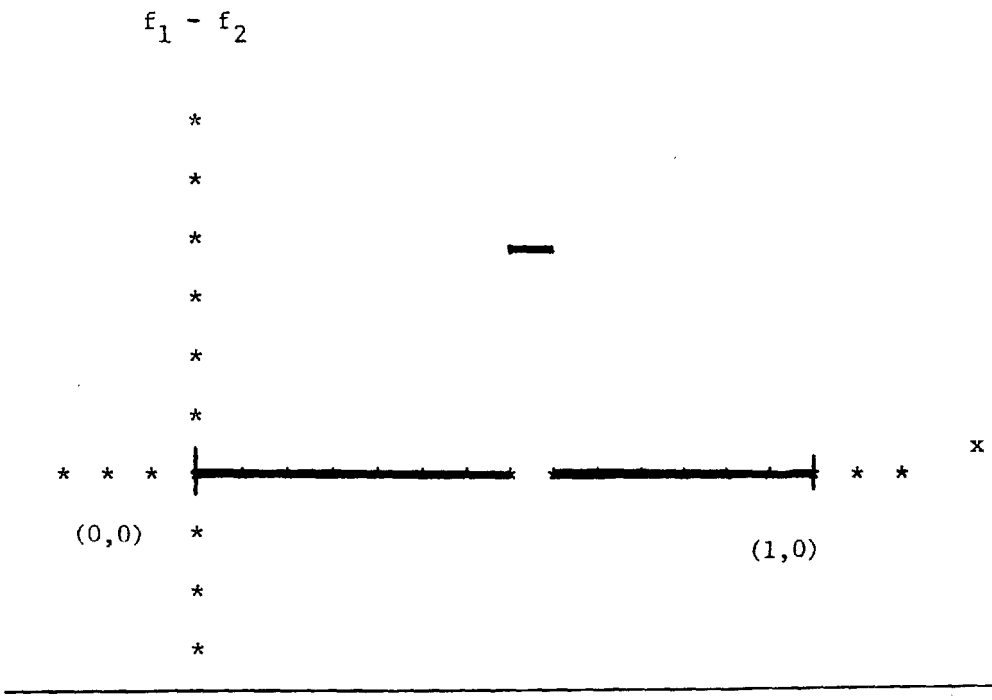
In the discrete case, and for linear models, we can give an indication of what should be the expected dependence of the performance of the system on blocksize.

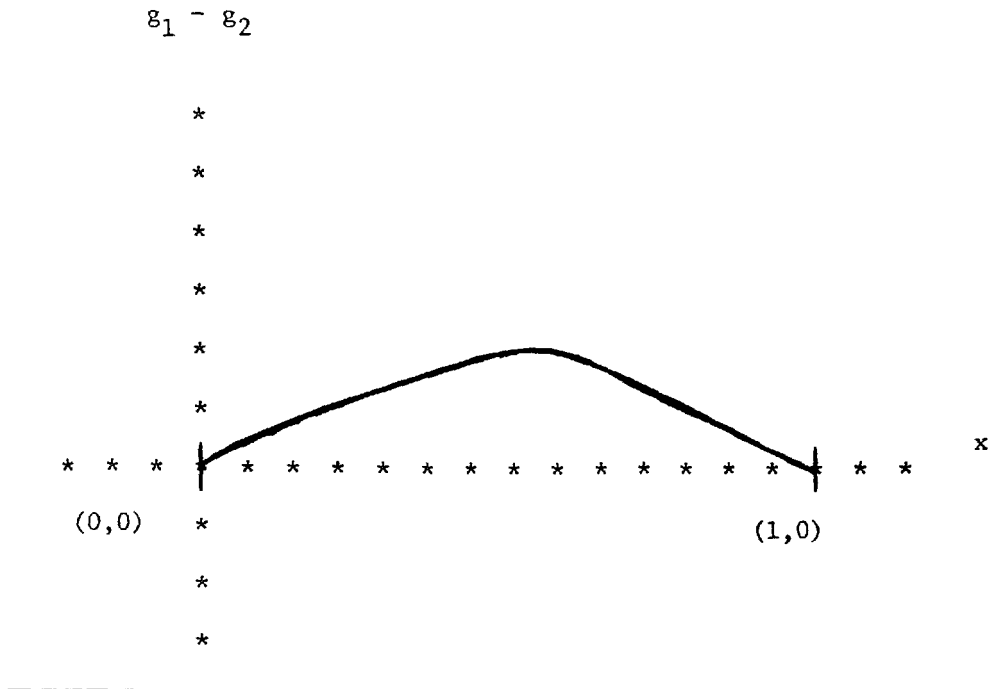
Suppose two messages f_1 and f_2 differ only at one position. The if we let g_1 and g_2 be the associated codetexts, we see that

$$g_1 - g_2 = \exp(-tA)(f_1 - f_2)$$

where $f_1 - f_2$ is zero except for the interval, say I, where the difference occurs. See Figure 10.1.

From the physical interpretation of the system as a model of heat flow, one would expect that $g_1 - g_2 \neq 0$ for all x, $0 < x < 1$, and this is in fact guaranteed by the maximum principle for pseudoparabolic equations [RU76]. Thus somebody who kept the codetext g to arbitrary precision would be assured that every change


$f_1 - f_2$
x
(0,0)
(1,0)


$g_1 - g_2$
x
(0,0)
(1,0)

in the plaintext would give rise to a change in every element of the codetext. In practice it is necessary to truncate $g(x)$. To how many places should $g(x)$ be kept? The answer depends on what you expect from the system. It is necessary to retain $g(x)$ to sufficient accuracy so you can invert to recover $f(x)$. For example if $f(x)$ is formed by the scheme (2.4) then an error in $f(x)$ of at most 0.5 is tolerable, and a further margin of safety would be desirable. Standard estimates for differential operators make it possible to calculate the maximum error in $g(x)$ for a given error in $f(x)$, and conversely. The actual factor will, of course, depend fairly strongly on the choice of the coefficients. For example if $a \leq 0$ in (6.11), the error in $f(x)$ obtained by inverting the equation in time will be less than that in $g(x)$. This is a consequence of the maximum principle. For this example, one could thus recover $f(x)$ despite an error in $g(x)$ of as much as ± 0.5.

An important consideration is to retain sufficient accuracy in $g(x)$ so that all (or at least a high fraction) of the values of g at the grid points are still seen to be positive after truncation, when g is the codetext coming from the function $f_1 - f_2$. Can this be estimated? This is in theory possible even in the full nonlinear case, but a practical bound may be exceedingly difficult to obtain. In the case of our simple model (6.11) we can make a reasonable quantifiable attempt. However, it should be stressed that in all cases the value of a $g(x)$ that is produced from such an $f_1 - f_2$ will decrease (linearly) with blocksize. Suppose the change in f represents one of unit height in a blocksize N. Then, if transformed to blocksize nN for $n > 1$, this would represent n changes, each of unit height.

If $a(x) \leq 0$ then from (6.7), (6.13) we see that

$$g(x) = e^{-1}(I + \mathcal{G} + \mathcal{G}^2/2 + \ldots)f$$

and

$$\mathcal{G}h = -\int_0^1 G_0(x,s)a(s)h(s)ds$$

Note that if $h \geq 0$ then $\mathcal{G}h \geq 0$ (a simple consequence of the maximum principle for ordinary differential equations). Thus $\mathcal{G}$ is a positive operator and it follows that

$$g(x) \geq e^{-1}\{f(x) - \int_0^1 G_0(x,s)a(s)f(s)\,ds\} \tag{10.1}$$

If

$$f(x) = \begin{cases} 1 & x_0 \leq x < x_0 + 1/N \\ 0 & \text{otherwise} \end{cases}$$

then, for example, if $x > x_0 + 1/N$

$$\begin{aligned} g(x) &\geq e^{-1} \int_{x_0}^{x_0+1/N} G_0(x,s)a(s)\,ds \\ &= \frac{1-x}{e} \int_{x_0}^{x_0+1/N} s\,a(s)\,ds \end{aligned} \tag{10.2}$$

The above can be used to obtain a lower bound for $g(x)$ at each of the gridpoints $x_i = 1/N, 2/N, \ldots, (N-1)/N$. To guarantee that such a locally supported $f(x)$ gives, in the truncated form of $g(x)$, a nonzero contribution at each of the gridpoints, it is enough to choose the accuracy to exceed the lower bound.

Note that in (10.2) of $a(x) \leq -1$ for all x in [0,1] then

$$g(x) \geq \frac{1-x}{e}\left(\frac{x_0}{N} + \frac{1}{2N^2}\right)$$

so that in the worst case of $x_0 = 1/N$, $x = (N-1)/N$ (looking at the change in g at the right hand boundary from a change in $f(x)$ concentrated at the left hand boundary)

$$g\left(\frac{N-1}{N}\right) \geq \frac{3}{2e}\frac{1}{N^3}$$

this would predict that with $N = 100$ one would require 7 figure accuracy in $g(x)$.

The choice of blocksize will have an effect on computational speed. In section 6 we showed that in the simplest case the calculation of $g = \exp(-A)f$ was equivalent to multiplying the N vector f by an $N \times N$ matrix, the matrix $\exp(-A)$, where N is the blocksize. In the nonlinear case the computational time using the common finite difference schemes is also proportional to N^2. Of course for a given size text, increasing the blocksize by a factor of N puts the text through the encryption process N times as fast, so that there is a linear increase in total computation time with increase in blocksize.

11. Bandwidth expansion in the linear case.

A limited bandwidth function is a function that can be represented as a finite sum of trigonometric polynomials,

$$f(x) = \sum_{n=1}^{M} a_n \sin(n\pi x) \tag{11.1}$$

We have restricted ourselves to sines above, because we have chosen to speak primarily of Dirichlet boundary conditions (2.3). Obviously, more general trigonometric polynomials are possible, as we have noted elsewhere. The reader can easily fill in the details. We shall say a function $f(x)$ has *L-limited bandwidth* if

$$f(x) = \sum_{n=1}^{M} a_n \phi_n(x). \tag{11.2}$$

Here $\{\phi_n(x)\}_{n=1}^{\infty}$ are the normalized eigenfunctions for the operator L defined in (4.4). It acts on $C^2[0,1]$ functions that vanish at $x = 0$, $x = 1$. Note that $\{\sin n\pi x\}_{n=1}^{\infty}$ are the eigenfunctions for the operator $Lu = u''$, with $u(0) = u(1) = 0$. Sturm-Liouville theory guarantees that the eigenvalues $\{\lambda_n\}_{n=1}^{\infty}$ of L obey the asymptotic formula

$$\lambda_n \approx n^2\pi^2 \tag{11.3}$$

and that the eigenfunctions $\phi_k(x)$ have exactly $k-1$ zeroes in $0 < x < 1$.

We claim that the encryption process using (5.1) - (5.3) does not increase the *L*-limited bandwidth of a function. Let

$$f(x) = \sum_{1}^{M} a_n \phi_n(x)$$

where $\phi_n(x)$ satisfies

$$\left.\begin{aligned} -L\phi_n &= -(p\phi_n')' + q\phi_n = \lambda_n \phi_n \qquad 0 < x < 1 \\ \phi_n(0) &= \phi_n(1) = 0. \end{aligned}\right\} \tag{11.4}$$

Then a simple separation of variables argument shows that if $u(x,t)$ satisfies

$$\begin{aligned} (L - I)u_t + Lu &= 0 \\ u(x,0) &= f(x) \end{aligned}$$

then

$$u(x,t) = \sum_{n=1}^{M} a_n e^{-\lambda_n t/(1+\lambda_n)} \phi_n(x)$$

and hence the codetext $u(x,t) = g(x)$ must equal

$$g(x) = \sum_{n=1}^{M} b_n \phi_n(x), \qquad b_n = a_n e^{-\lambda_n/(1+\lambda_n)}. \tag{11.5}$$

Not only does $g(x)$ have the same L-limited bandwidth as $f(x)$ but, for large n, the energy in each band is of the same order of magnitude

$$\frac{b_n}{a_n} = e^{-\lambda_n/(1+\lambda_n)} \approx e^{-1}e^{1/(1+n^2\pi^2)} \tag{11.6}$$

A transmitter can be thought of as the vibration of a *homogeneous* material, the governing equation of motion being the wave equation, which in one space dimension is

$$u_{tt} - u_{xx} = 0.$$

If we could build an *inhomogeneous* transmitter whose motion was governed by the hyperbolic equation

$$u_{tt} - Lu = 0 \tag{11.7}$$

defined on the domain $0 \leq x \leq 1$, $0 \leq t \leq 1$, and with the operator L as before, then the output, instead of being the sum of the eigenfunctions of the operator $\frac{d^2}{dx^2}$ with the corresponding frequencies corresponding to the eigenvalues. $\{n^2\pi^2\}_{n=1}^{\infty}$ will be the eigenfunctions of the operator Lu with the frequencies determined by the eigenvalues $\{\lambda_n\}_{n=1}^{\infty}$.

If a transmitter/receiver pair were to be built using (11.7) as the governing equation then this process would share many of the ideas of our cryptosystem for the linear case. Even if an eavesdropper could determine all the frequencies of vibration of this linear system, (tantamount to knowing all the eigenvalues of L), then this is insufficient to recover the operator L [HO73]. This is a statement of the classical inverse Sturm-Liouville problem. If additional information is given, for example the *energy* in each eigenmode, then recovery methods are possible in one space dimension. In higher space dimensions the determination of L from such spectral data remains an enigma. With current technology it might be possible to reconfigure a transmitter/receiver pair electronically without actually modifying the hardware.

Finally, it should be noted that the bandwidth expansion problem in the nonlinear case is difficult to treat theoretically due to the loss of the superposition principle. The maximum principle guarantees that there will be bandwidth expansion, and numerical simulations could be performed in order to obtain quantitative estimates.

12. Numerical examples.

We ran a numerical simulation of the encryption system in a simple case; taking the linear model and restricting our attention to the equation (5.1) with L the operator $\frac{d^2}{dx^2} = q(x)$. We converted alphanumeric plaintext into a piecewise constant function $f(x)$ by means of (2.4), and adjusted the norm of the key $q(x)$ so that the codetext $g(x)$, when evaluated at the gridpoints $x_i = i/N$, lay in the range 0–999 after rounding off to the nearest integer. This retention in accuracy in $g(x)$ was sufficient to recover the plaintext $f(x)$ in all cases that we ran. The maximum blocksize attempted was $N = 512$. With N in the range 50–100 we found that a single change in a character in the plaintext changed on average all of the values of the codetext, although some by only one or two numbers. The greatest change was usually near the gridpoint where the change in the plaintext occured, but this was somewhat key-dependent.

When we attempted to decode a message with a key that differed at only one gridpoint from the one used to encrypt, we found that the resulting "plaintext" had changed by one or two numbers in about 1/3 to 1/2 of the positions, again concentrated near the position of key change. Blocksize was again in the range 50–100.

There is no reason why such an approach need be restricted to a single space variable x. In fact fax, photos and other multidimensional messages might more naturally be considered by means of pseudoparabolic PDEs in $\Omega \times [0,1]$ where Ω is some appropriate region in $\mathbb{R}^n$.

13. The need for error control in the discrete case.

It is clear that finite computational resources produce a decryption which is merely close to, not exactly equal to, the plaintext which was originally encrypted. The difference can be enough to change a symbol here and there. This will put a slightly perturbed version of the original plaintext into the receiver's hands. If large block size (of the order of hundreds of bits) is used there is not much overhead expense in applying an agreed-upon error-control coding process to the plaintext before encrypting. If the language in which the plaintext message is written has a fair amount of redundancy this may not be necessary. If that language has almost no redundancy, then very cheap simple cryptosystems are probably adequate to conceal message traffic in it.

14. Discussion

The purpose of this paper has been to demonstrate the feasibility of basing a family of conventional (as opposed to public key) cryptosystems on a circle of hard problems arising in the theory of partial differential equations. We have shown why it is hard to avoid the use of nonlinear pseudoparabolic PDEs and, possibly, of nonlinear boundary conditions in formulating such a cryptosystem. Our methodology is neutral as regards continuous or discrete messages. It seems quite amenable to analog calculations now that there are natural purely analog methods [PE 86] for time reversal of an optical signal.

As often happens in cryptographic discussions, we have actually said only that somebody who knows how to solve interesting and long-standing hard problems (in analysis, in the case of this family of cryptosystems) can break cryptosystems expeditiously. But like other cryptosystem designers we allow ourselves to think that, so far, it looks as if the converse is also true.

This research was supported, in part, by NSA Grant MDA 904-83-H-0002.

15. References

BL85 G. R. Blakley, Information theory without the finiteness assumption, I: Cryptosystems as group-theoretic objects. in *G. R. Blakley and D. Chaum (editors), Advances in Cryptology, Proceedings of Crypto '84*, Springer-Verlag, Berlin (1985), 314-338.

BL86 G. R. Blakley, Information theory without the finiteness assumption, II: Unfolding the DES. in *H. Williams (editor), Advances in Cryptology, Proceedings of Crypto '85*, Springer-Verlag, Berlin (1986), to appear.

BL87 G. R. Blakley and C Meadows, Information theory without the finiteness assumption, III: Data compression and codes whose rates exceed unity. *Proceedings of the 1986 Cirencester Conference on Cryptography and Coding, IMA*, (1987) to appear.

BR85 E. Brickell, Breaking iterated knapsacks, in *G. R. Blakley and D. Chaum (editors), Advances in Cryptology, Proceedings of Crypto '84*, Springer-Verlag, Berlin (1985), 342-358.

DA86 G. I. Davida, C. Gilbertson and G. Walter, Analog cryptosystems, in *Proceedings of Eurocrypt '85*, Springer-Verlag, Berlin (1986), to appear,
also
Technical Report TRCS-84-1. *Department of Electrical Engineering and Computer Science, University of Wisconsin, Milwaukee*, (1984).

DE83 D. E. R. Denning, Cryptography and Data Security, Addison-Wesley, Reading, Massachusetts (1983).

FR64 Partial Differential Equations of Parabolic type, Prentice Hall, Englewood Cliffs, New Jersey, (1964).

GA79 M. R. Garey and D. S. Johnson, Computers and Intractibility: A Guide to the Theory of NP-Completeness, Freeman, San Francisco (1979).

HA23 J. Hadamard, Lectures on the Cauchy Problem in Linear Partial Differential Equations, Yale Univ. Press. New Haven, 1923.

HO73 H. Hochstadt, The inverse Sturm-Liouville problem, *Comm. Pure Applied Math.*, **16,** 715–729, (1973).

PA74 L. E. Payne, Improperly Posed Problems in Partial Differential Equations, Springer Lecture Notes, (1974).

PE 86 D. M. Pepper, Applications of optical phase conjugation, *Scientific American,* **254**, No 1, 74–83, (1986).

PR67 M. H. Protter and H. Weinberger, Maximum Principles in Differential Equations, Prentice Hall, Englewood Cliffs, New Jersey (1967).

PU 74 G. B. Purdy, A high security log-in procedure, *Comm. ACM,* **17** 442–445 (1974).

RU76 W. Rundell and M. S. Stecher, Maximum principles for pseudoparabolic partial differential equations, *SIAM J. Math. Analysis,* **7** (1976), 898–912.

RU79 W. Rundell and M. S. Stecher, The nonpositivity of solutions to pseudoparabolic equations, *Proc. Amer. Math. Soc.*, **75,** No. 2 (1979), 251–254.

SH70 R. E. Showalter and T. W. Ting, Pseudoparabolic partial differential equations, *SIAM J. Math. Analysis,* **1** (1970), 1–25.

SH70b R. E. Showalter, Well-posed problems for a partial differential equation of order $2m + 1$. *SIAM J. Math. Anal.* **1** (1970) 214–231.

WE65 H. Weinberger, A First Course in Partial Differential Equations, Xerox College Publishing, Lexington, Massachusetts, (1965).

YO73 D. M. Young and R. T. Gregory, A Survey of Numerical Mathematics, Volume 2, Addison-Wesley, Reading, Massachusetts (1973).