

Encryption Schemes using Finite Frames and Hadamard Arrays

Ryan Harkins, Eric Weber, and Andrew Westmeyer

CONTENTS

- 1. Introduction
- 2. Introduction to Frames
- 3. Five Encryption Schemes
- 4. Experimental Results and Conclusions
- 5. Pseudocode
- 6. Graphs
- Acknowledgments
- References

We propose a cipher similar to the one-time pad and McEliece cipher based on a subband coding scheme. The encoding process is an approximation to the one-time pad encryption scheme. We present results of numerical experiments that suggest that a brute force attack on the proposed scheme does not result in all possible plaintexts, as the one-time pad does, but the brute force attack does not compromise the system. However, we demonstrate that the cipher is vulnerable to a chosen-plaintext attack.

1. INTRODUCTION

In this paper, we propose a private key cipher, the idea for which comes from frame theory and multiple access communications. The cipher has similarities to the Hill cipher, the one-time pad, and the McEliece cipher [Menezes et al. 97, Chabaud 95]. Indeed, one of the design goals for our cipher is to approximate the one-time pad.

Other design goals include the following:

1. include randomness in the encryption process;
2. require that the key be shared only once;
3. use a relatively small key size;
4. make it computationally fast;
5. be robust against brute force attacks.

Our proposed cipher satisfies items (1)–(4) above; the purpose of the present paper is to study the cipher's robustness against brute force attacks. We remark here that (5) is not sufficient for the cipher to be a good one, but certainly is necessary. We will demonstrate that this cipher is vulnerable to a chosen-plaintext attack. It is unknown if this cipher is robust against a known-plaintext attack.

2000 AMS Subject Classification: Primary 94A60, 68P25; Secondary 42C99

Keywords: Cryptography, finite frames, random matrix

Our cipher can be described as follows: consider a communications channel; we divide the channel into two subbands, one which will carry the message and the other which will carry noise, or as we call it in this paper, garbage. The message, along with the garbage, is transmitted over the channel; the recipient then filters out the garbage, leaving only the message. This procedure is carried out using orthogonal frames and requires their construction; the easiest way to do this is using Fourier frames (also called harmonic frames). However, as will be described, these frames are not good for our purposes here, and so we present several alternative methods for constructing orthogonal frames.

In Section 2, we give a short introduction to frames and, in particular, orthogonal frames. In Section 3, we give an account of several methods for constructing orthogonal frames, with remarks regarding our design goals. In Section 4, we present the results and conclusions of our numerical experiments and the chosen-plaintext attack. In Section 5, we provide pseudocode to describe the experiments. Finally, in Section 6, we present the graphs displaying the results of our experiments.

2. INTRODUCTION TO FRAMES

Frames for Hilbert spaces are used in many signal processing applications, such as sampling theory, multiple access communications, etc. Frames provide redundancy via overcompleteness, where bases do not, and it is this redundancy that makes them advantageous to use in these settings. In this paper, we will utilize this redundancy of frames for the purpose of encryption.

Let H be a Hilbert space over the field \mathbb{F} with scalar product $\langle \cdot, \cdot \rangle$ and norm $\| \cdot \|$, where \mathbb{F} denotes either \mathbb{R} or \mathbb{C} . A frame for H is a sequence $\mathbb{X} := \{x_n\}_{n \in \mathbb{Z}}$ such that there exist constants $0 < A \leq B < \infty$ such that for all $v \in H$,

$$A\|v\|^2 \leq \sum_{n \in \mathbb{Z}} |\langle v, x_n \rangle|^2 \leq B\|v\|^2. \tag{2-1}$$

Clearly, a frame spans the Hilbert space. Moreover, $\{x_n\}$ defines the following *frame operator*:

$$S_{\mathbb{X}} : H \rightarrow H : v \mapsto \sum_{n \in \mathbb{Z}} \langle v, x_n \rangle x_n,$$

which is positive and invertible. Define $\{\tilde{x}_n\} \subset H$, the *standard dual* of $\{x_n\}$, by $\tilde{x}_n := S_{\mathbb{X}}^{-1} x_n$; then, for all $v \in H$,

$$v = \sum_{n \in \mathbb{Z}} \langle v, x_n \rangle \tilde{x}_n = \sum_{n \in \mathbb{Z}} \langle v, \tilde{x}_n \rangle x_n.$$

If $A = B = 1$, the frame is said to be *Parseval*, and then for all $v \in H$,

$$v = \sum_{n \in \mathbb{Z}} \langle v, x_n \rangle x_n.$$

For elementary frame theory, see [Han and Larson 00, Casazza 00].

If H is finite-dimensional (H will always be assumed to be so from here on, unless specifically stated), then a frame sequence (possibly finite) is any spanning set $\{x_n\}$ such that $\sum_{n \in \mathbb{Z}} \|x_n\|^2 < \infty$. If only a finite number of the x_n are nonzero, then $\{x_n\}$ is a finite frame, and we will discard those that are zero. See [Casazza and Kovacević 01, Dykema et al. 04, Benedetto et al. 03] for more on finite frames.

For convenience of notation, we make the following definition:

Definition 2.1. An $n \times n$ real matrix, M , is an *orthogonal matrix* if $M^T M = kI_n$ for some constant k .

The (finite) Parseval frames in H are characterized in Proposition 2.2.

Proposition 2.2. Let $\{x_n\}_{n=1}^M \subset H$, where H has dimension N . The following are equivalent:

1. $\{x_n\}$ is a Parseval frame for H ;
2. the $M \times N$ matrix whose i th row is x_i (as a row vector) has columns which are orthonormal;
3. there exists a Hilbert space K of dimension $M - N$ and vectors $\{y_n\}_{n=1}^M \subset K$ such that the $M \times M$ matrix formed by

$$\left(\begin{array}{c|c} x_1 & y_1 \\ \vdots & \vdots \\ x_M & y_M \end{array} \right)$$

is a unitary matrix.

Here, we write the vectors x_i and y_i as row vectors with respect to any orthonormal bases for H and K , respectively.

Proof: The proof of the equivalence of (1) and (2) can be found in [Benedetto et al. 03]. The proof of the equivalence of (1) and (2) for infinite frames can be found in [Han and Larson 00, Corollary 1.3, Theorem 1.7]. The case for finite frames is analogous. \square

Remark 2.3. Another way to view Proposition 2.2 is that $\{x_n\}$ is a Parseval frame for H if and only if $\{x_n\}$ is the inner direct summand of an orthonormal basis $\{x_n \oplus y_n\}$ for some superspace $H \oplus K$ of H .

Definition 2.4. Two frames $\{x_n\}_{n=1}^M \subset H$ and $\{y_n\}_{n=1}^M \subset K$ are *orthogonal* if for all $v \in H$, $\sum_{n=1}^M \langle v, x_n \rangle y_n = 0$.

Proposition 2.5. Suppose $\{x_n\}_{n=1}^M \subset H$ and $\{y_n\}_{n=1}^M \subset K$ are Parseval frames; they are orthogonal if and only if

$$\left(\begin{array}{c|c} x_1 & y_1 \\ \vdots & \vdots \\ x_M & y_M \end{array} \right) := (P|Q)$$

has columns which form an orthonormal set.

Proof: (\Leftarrow) Consider the two matrices P and Q whose rows are $\{x_n\}$ and $\{y_n\}$, respectively. A straightforward computation demonstrates that for $v \in H$,

$$\sum_{n=1}^M \langle v, x_n \rangle y_n = Q^* P v, \quad (2-2)$$

where Q^* is the conjugate transpose of Q . It follows that if the above matrix has orthonormal columns, then $Q^* P = 0$, and thus the frames $\{x_n\}$ and $\{y_n\}$ are orthogonal.

(\Rightarrow) Conversely, suppose the Parseval frames are orthogonal. Note that by Proposition 2.2, the left part P of the above matrix has orthonormal columns; likewise, the right part of the matrix Q also has orthonormal columns. By Equation (2-2), we must have that the columns of the left part of the matrix are orthogonal to the columns of the right part of the matrix. Hence, the columns of the matrix form an orthonormal set. \square

Note that if $\{x_n\}$ is orthogonal to $\{y_n\}$, then $\{y_n\}$ is orthogonal to $\{x_n\}$.

Let $\mathbb{X} := \{x_n\}_{n=1}^M \subset H$; the analysis operator $\Theta_{\mathbb{X}}$ of $\{x_n\}$ is given by

$$\Theta_{\mathbb{X}} : H \rightarrow \mathbb{F}^M : v \mapsto (\langle v, x_1 \rangle, \langle v, x_2 \rangle, \dots, \langle v, x_M \rangle).$$

The matrix representation of $\Theta_{\mathbb{X}}$ is given as the matrix P in Proposition 2.5. The proof of Proposition 2.5 shows that two frames $\{x_n\}$ and $\{y_n\}$ are orthogonal if and only if their analysis operators $\Theta_{\mathbb{X}}$ and $\Theta_{\mathbb{Y}}$ have orthogonal ranges in \mathbb{F}^M .

2.1 Encryption Using Orthogonal Frames

We present here an overview of our proposed private key encryption scheme using orthogonal frames. For motivation, consider that the one-time pad is an unconditionally secure cipher, which is optimal among all unconditionally secure ciphers in terms of key length [Menezes et al. 97]. Our encryption scheme, which is similar to a subband coding scheme, is an effort to approximate the one-time pad. The (private) key for this encryption scheme is two orthogonal Parseval frames $\{x_n\}_{n=1}^M \subset H$ and $\{y_n\}_{n=1}^M \subset K$. Let $\Theta_{\mathbb{X}}$ and $\Theta_{\mathbb{Y}}$, respectively, denote their analysis operators. Suppose $m \in H$ is a message; let $g \in K$ be a nonzero vector chosen at random. The ciphertext $c \in \mathbb{F}^M$ is given as follows:

$$c := \Theta_{\mathbb{X}} m + \Theta_{\mathbb{Y}} g.$$

To recover the message, we apply $\Theta_{\mathbb{X}}^*$:

$$\begin{aligned} \Theta_{\mathbb{X}}^* c &= \Theta_{\mathbb{X}}^* \Theta_{\mathbb{X}} m + \Theta_{\mathbb{X}}^* \Theta_{\mathbb{Y}} g \\ &= \sum_{n=1}^M \langle m, x_n \rangle x_n + \sum_{n=1}^M \langle m, y_n \rangle x_n \\ &= m + 0 = m. \end{aligned}$$

There are several things to note about our scheme:

1. The frame $\{x_n\}$ need not be Parseval, but Parseval frames are, in general, easier to work with. Since the Parseval frames form only a small subset of all possible frames, using general frames would allow a much greater choice of specific encryption keys.
2. The frame $\{y_n\}$ need not be Parseval; it need not even be a frame, though again Parseval frames simplify matters. If $\{y_n\}$ is not a frame, then $\Theta_{\mathbb{Y}}$ has a nontrivial kernel, and $\Theta_{\mathbb{Y}} g$ could be 0 if g is chosen to be in the kernel. (Below, we will actually use scalar multiples of Parseval frames for both $\{x_n\}$ and $\{y_n\}$.)
3. Just as with the one-time pad, when done properly, encoding a message twice results in two different ciphertexts.
4. Unlike the one-time pad, in which a brute force attack results in all possible plaintexts, it appears unlikely that a brute force attack on our system would result in the same. Our simulations indicate that an attack produces either a text which is very close to the original plaintext or is gibberish. (See the graphs in Section 6 for more.) However, at this time, we cannot prove why this is so.

Proposition 2.6. *If $\{x_n\}_{n=1}^M \subset H$ and $\{y_n\}_{n=1}^M \subset K$ are orthogonal frames, then $M \geq \dim(H) + \dim(K)$.*

Proof: Let Θ_X and Θ_Y be the respective analysis operators. Note that by the (lower) frame inequality in Equation (2-1), both Θ_X and Θ_Y are one-to-one. Moreover, the orthogonality of the frames is equivalent to the orthogonality of the ranges of Θ_X and Θ_Y . Combining these two observations establishes the proposition. \square

For convenience, we will assume that $M = \dim(H) + \dim(K)$. The ciphertext is

$$c = \Theta_X m + \Theta_Y g,$$

where $\{x_n\}$ and $\{y_n\}$ are orthogonal Parseval frames. Since they are orthogonal, we write

$$c = (\Theta_X | \Theta_Y) m \oplus g,$$

where the matrix $(\Theta_X | \Theta_Y)$ is an isometry. Therefore, our encryption procedure involves generating a large orthogonal matrix.

The next section discusses several ways to construct such matrices. Since the encryption scheme is a private key system, we wish to have a relatively small key size; the entire matrix is too much information to be used as the key. We discuss below some of the strengths and weaknesses of the various construction techniques.

3. FIVE ENCRYPTION SCHEMES

The cipher algorithm depends upon generating a pair of random orthogonal frames, each of which is the size of the message. This is equivalent to producing a random orthogonal matrix of twice the size of the message. We investigate here several methods for doing so. The first method takes the view of producing orthogonal frames using Fourier frames. The remaining methods take the view of producing orthogonal matrices.

Once the orthogonal frames or orthogonal matrices, are determined, the encryption and decryption process is the same. If the frames are given by X and Y , then we write the matrix $(\Theta_X | \Theta_Y)$; if, on the other hand, the matrix is A , we think of $A = (\Theta_X | \Theta_Y)$. Given a message m , choose at random a vector g , called the “garbage” or “noise,” and compute $(\Theta_X | \Theta_Y) m \oplus g = c$ to yield the cipher text c . The recipient computes

$$\Theta_X^T c = \Theta_X^T (\Theta_X | \Theta_Y) m \oplus g = Km \oplus 0 = Km,$$

where K is the square of the norm of any column of the matrix Θ_X . Dividing by K then reproduces the message.

3.1 Scheme 1

The first algorithm utilizes the discrete cosine transform. The original idea came from using the discrete Fourier transform, which involves complex exponentials. The discrete cosine transform, in matrix form, is given by

$$C = [c_{kn}] = \left[\lambda_k \sqrt{\frac{2}{M}} \cos \left\{ \frac{k\pi}{M} (n + 1/2) \right\} \right],$$

where $n = 1, \dots, M$, $\lambda_1 = 1/\sqrt{2}$, and $\lambda_k = 1$ for all $k = 2, \dots, M$. Note that this is normalized to be a unitary matrix. Assuming that $M = 2N$, one can permute the columns of C to yield C' and divide the resulting matrix in half vertically:

$$C' = (\Theta_X | \Theta_Y).$$

The resulting divided matrix can then be viewed as the analysis operators for two orthogonal frames, each for \mathbb{R}^N , consisting of cosine bases projected onto smaller subspaces (Proposition 2.5, see also [Aldroubi et al. 04]). Moreover, the frame vectors can be weighted, which is accomplished by a diagonal invertible matrix D . Let P denote a permutation matrix.

The (private) key for the cipher then consists of the matrix D (or simply its diagonal entries) and the permutation corresponding to P . The encryption algorithm of a message m of length N then consists of randomly generating a garbage vector $g \in \mathbb{R}^N$ and computing the ciphertext

$$c = CDP(m \oplus g).$$

To decrypt the message, we apply the matrix $QP^T D^{-1} C^T$ to the ciphertext, where Q is the projection of \mathbb{R}^M onto the first N coordinates:

$$QP^T D^{-1} C^T CDP(m \oplus g) = Q(m \oplus g) = m.$$

Remark 3.1. We note that the only knowledge unknown to an adversary is D and P ; the adversary will know C . Hence, C is irrelevant to the cipher algorithm. Because of this, the algorithm reduces to rearrangement followed by weighting of the entries of the message and the garbage. We conclude that our first algorithm is a poor one.

3.2 Scheme 2

The second scheme involves using Hadamard arrays to generate orthogonal matrices. We first start with the definition of Hadamard arrays. We remark here that this scheme is related to linear codes [Delsarte and Goethals 69].

Definition 3.2. [Wallis 72] A Hadamard array $H[h, k, \lambda]$ based on the indeterminates x_1, x_2, \dots, x_k , with $k \leq h$, is an $h \times h$ matrix with entries chosen from $\{\pm x_1, \pm x_2, \dots, \pm x_k\}$ in such a way that:

1. in any row there are λ entries $\pm x_1$, λ entries $\pm x_2$, \dots , λ entries $\pm x_k$, and similarly for the columns;
2. the rows and columns are (formally) pairwise orthogonal, respectively.

The matrices we use for our encryption scheme have $h = k$ and $\lambda = 1$. The only possible Hadamard arrays of this type are for $h = 1, 2, 4, 8$ [Agaian 85]. For indeterminants A through H , we have the Hadamard array

$$H[8, 8, 1] = \left[\begin{array}{cccc|cccc} A & B & C & D & E & F & G & H \\ -B & A & D & -C & F & -E & -H & G \\ -C & -D & A & B & G & H & -E & -F \\ -D & C & -B & A & H & -G & F & -E \\ -E & -F & -G & -H & A & B & C & D \\ -F & E & -H & G & -B & A & -D & C \\ -G & H & E & -F & -C & D & A & -B \\ -H & -G & F & E & -D & -C & B & A \end{array} \right].$$

For $\Theta = H[8, 8, 1]$, $\Theta^T \Theta = KI_8$ where $K = A^2 + B^2 + \dots + H^2$.

The Hadamard arrays allow easy construction of matrices (and hence tight frames) needed in our encryption schemes. For the encryption process, we now have only Θ to construct, instead of computing the matrices C, D , and P .

The encryption process starts with a message m of arbitrary length and then divides m into blocks m_1, \dots, m_q of length 4 (padding the last block with 0s if necessary). Then, random vectors g_1, \dots, g_q of length 4 are chosen, and the matrix Θ is applied successively to $m_i \oplus g_i$. The ciphertext is then

$$c = \Theta(m_1 \oplus g_1) \oplus \dots \oplus \Theta(m_q \oplus g_q).$$

The message is then decrypted by dividing c into blocks c_1, \dots, c_q of size 8, computing $K\Theta^T c_i$ for $i = 1, \dots, q$, and reconstructing the message using the first four entries of these resulting blocks.

Remark 3.3. Because of the ease of construction of the Hadamard arrays, the system is quite easy to implement. Unlike the first scheme, the key for the recipient has now been reduced to knowing the chosen entries for Θ ; hence,

in this case, the key is the entries A, B, \dots, H of the matrix Θ . Since Hadamard arrays are small, however, we wish to find an algorithm to generate larger orthogonal matrices.

3.3 Scheme 3

This scheme is an attempt to produce larger orthogonal matrices. Starting with Hadamard arrays A and M with $A^T A = kI_8$ and $M^T M = pI_8$ for constants k and p , we construct a new 16×16 orthogonal matrix

$$S = \begin{bmatrix} A & MA \\ -M^T A & A \end{bmatrix}.$$

Repeat this procedure with Hadamard arrays B and N to get

$$T = \begin{bmatrix} B & NB \\ -N^T B & B \end{bmatrix}.$$

The matrices S and T are then used to construct a 32×32 orthogonal matrix

$$U = \begin{bmatrix} S & TS \\ -T^T S & S \end{bmatrix}.$$

This ‘‘blow up’’ construction is iterated to get the appropriate size matrix for our plain text.

Remark 3.4. In this encryption scheme, the key is the entries of the matrices A, B, M, N , etc., and their positions in the construction. This method, however, is computationally inefficient.

3.4 Scheme 4

We first define the tensor product, \otimes , of two matrices, A and B . The sizes of the matrices are irrelevant.

Definition 3.5. [van Lint 92] Let

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & & \ddots & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}.$$

Then

$$A \otimes B := \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ \vdots & & \ddots & \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{bmatrix}.$$

If A is an $m \times n$ matrix and B is a $p \times q$ matrix, then $A \otimes B$ is an $mp \times nq$ matrix. The tensor product will be the critical element of construction in this and the next

$$C = \begin{bmatrix} H \otimes A & (H \otimes B)(H \otimes A) \\ -(H \otimes B)^T(H \otimes A) & H \otimes A \end{bmatrix} = \left[\begin{array}{cccc|cccc} A & A & A & A & 4BA & 0 & 0 & 0 \\ A & -A & A & -A & 0 & 4BA & 0 & 0 \\ A & A & -A & -A & 0 & 0 & 4BA & 0 \\ A & -A & -A & A & 0 & 0 & 0 & 4BA \\ \hline -4B^T A & 0 & 0 & 0 & A & A & A & A \\ 0 & -4B^T A & 0 & 0 & A & -A & A & -A \\ 0 & 0 & -4B^T A & 0 & A & A & -A & -A \\ 0 & 0 & 0 & -4B^T A & A & -A & -A & A \end{array} \right]$$

FIGURE 1.

scheme. Note that if A and B are orthogonal matrices, then $A \otimes B$ is also an orthogonal matrix.

Definition 3.6. A *Hadamard matrix* is a square orthogonal matrix with entries consisting of ± 1 s.

We start with an Hadamard matrix (not an array), H , of a chosen size 2^p , and then two Hadamard arrays, A and B of chosen sizes 2, 4, or 8. We then construct the new matrix via the tensor products,

$$C = \begin{bmatrix} H \otimes A & (H \otimes B)(H \otimes A) \\ -(H \otimes B)^T(H \otimes A) & H \otimes A \end{bmatrix}.$$

C is now an orthogonal matrix. This matrix is size adaptive with respect to powers of 2, since each matrix is of some order of 2, and the size of H can be chosen.

However, the Hadamard matrix property that $H^T H = I_n$ is actually a disadvantage. Let

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

The resulting matrix (see Figure 1) is relatively sparse, which is undesirable for maintaining secrecy.

3.5 Scheme 5

We choose p Hadamard arrays H_1, H_2, \dots, H_p . Each array can have its own size, say $e_i \times e_i$ for $1 \leq i \leq p$, where each e_i is either 2, 4, or 8. We then construct our $e_1 e_2 \dots e_p$ -sized matrix M by the tensor product of these p matrices:

$$M = \bigotimes_{i=1}^p H_i := H_1 \otimes H_2 \otimes \dots \otimes H_p.$$

The ciphertext is $c = M(m \oplus g)$. With this construction, we eliminate the sparsity that was shown in Scheme 4.

Note that the key in this case consists of the entries of the first rows of H_1 to H_p ; hence, it is an array of numbers of size $e_1 + e_2 + \dots + e_p$ and is relatively small.

We ran some numerical experiments, using Scheme 5 to obtain information regarding several things:

1. We wanted to see if a brute force attack would be a feasible way of defeating the cipher. The results of the experiments and also the computations below suggest that the answer is no.
2. One advantage of the one-time pad is that a brute force attack results in all possible plaintext messages, forcing an adversary to choose which was the original message. We wanted to determine if this was also true of our proposed cipher. The results of our experiments indicate that the answer to this is also no.
3. Finally, we wanted to determine if the size of the entries of the garbage vector g mattered. The experiments and the computations below suggest that the answer is yes.

The results of our experiments, in the form of graphs, are given in Section 6.

4. EXPERIMENTAL RESULTS AND CONCLUSIONS

We want to know how accurate a guess has to be in order to break the cipher. We suppose that an adversary knows that we are using Scheme 5; that is, the adversary knows the structure of the matrix M , but not the entries. We let M be the original matrix of size n , \tilde{M} be the adversary's guess, and w be the original plaintext m concatenated with the garbage g (i.e., $w = m \oplus g$). Then, we consider $\tilde{w} := (1/\tilde{k})\tilde{M}^T M w$ where $\tilde{k} = \|\tilde{M}\|^2$. Since we assume that the structure of M is known by the adversary, we

consider $\tilde{M} = M + P$, where P is a matrix with the same structure as M .

Our first question is whether an adversary can figure out how small the perturbation P must be in order to get a “good guess.” The adversary knows the size of M and $\|Mw\|$; we assume additionally that the adversary knows the structure of M . For convenience, assume that the encryption matrix $M = A \otimes B \otimes C$ for three Hadamard arrays, A , B , and C . We then let $\tilde{M} = (A+a) \otimes (B+b) \otimes (C+c)$ for (small norm) perturbation matrices a , b , and c . We reformulate our question: how big can $\|a\|$, $\|b\|$, and $\|c\|$ be such that $\|M^T Mw - \tilde{M}^T Mw\| < \varepsilon$, where ε is some acceptable tolerance for error? (Here, for a matrix A , $\|A\|$ denotes the operator norm of A . Below, $\|\cdot\|$ shall denote both Hilbert space norm for vectors and operator norm for matrices.)

We let $\|a\| \approx \|b\| \approx \|c\| \approx \beta$ and $\|A\| \approx \|B\| \approx \|C\| \approx \gamma$. We may assume that $\gamma \gg \beta$. If we write out \tilde{M} in terms of the tensor products, we get

$$\tilde{M} = A \otimes B \otimes C + A \otimes B \otimes c + \dots + a \otimes b \otimes c$$

and

$$\|\tilde{M}\| \leq \gamma^3 + 3\gamma^2\beta + 3\gamma\beta^2 + \beta^3.$$

Given any $\varepsilon > 0$, we choose $\delta = \varepsilon/\|Mw\|$. If $|3\gamma^2\beta| < \delta$, then

$$\begin{aligned} \|M^T Mw - \tilde{M}^T Mw\| &\leq \|M^T - \tilde{M}^T\| \|Mw\| \\ &\leq (3\gamma^2\beta + 3\gamma\beta^2 + \beta^3) \|Mw\| \\ &\approx 3(\gamma^2\beta) \|Mw\| < \delta \|Mw\| = \varepsilon. \end{aligned}$$

These computations suggest that the larger the entries of the garbage vector g are, the closer a guess must be in order to reasonably recover the message. This is corroborated by the experiments we ran (see the graphs in Section 6). Thus, we can control the accuracy an adversary would need in order to break the cipher.

4.1 Chosen-Plaintext Attack

We will demonstrate here a chosen-plaintext attack on the cipher which will break the system. A chosen-plaintext attack is an attack mounted by an adversary that chooses a plaintext and is then given the corresponding ciphertext.

Theorem 4.1. *The encryption algorithm proposed above is vulnerable to a chosen-plaintext attack.*

Proof: We assume that the adversary knows the length of the message band and, subsequently, the length of the

noise band. Let the length of the message band be N_m and the length of the noise band be N_n . The attack proceeds as follows:

1. Determine the range of the noise band K of Θ ; that is, determine $(\Theta_{\mathbb{X}}|\Theta_{\mathbb{Y}})(0 \oplus \mathbb{R}^{N_n})$. Choose any plaintext m of size N_m . Encode the plaintext twice, with output, say e_0 and e_1 . Compute $e_1 - e_0 = \Theta(m \oplus g_1) - \Theta(m \oplus g_0) = \Theta(0 \oplus g_1 - 0 \oplus g_0)$. Notice that this yields a vector $f_1 = \Theta(0 \oplus g_1 - 0 \oplus g_0)$ in the range of the noise band of Θ . Encode the plaintext a third time, with output e_2 , and compute $f_2 = e_2 - e_0$. Compute f_3, \dots, f_m until the collection $\{f_1, \dots, f_m\}$ contains a linearly independent subset of size N_n . This determines the range of the noise band K of Θ .
2. Determine the range of the message band T of Θ ; that is, determine $(\Theta_{\mathbb{X}}|\Theta_{\mathbb{Y}})(\mathbb{R}^{N_m} \oplus 0)$. Choose any plaintext m_1 of size N_m ; encode the plaintext with output e_1 ; then project e_1 onto the orthogonal complement of K . This yields a vector x_1 in T . Choose another plaintext m_2 and repeat, yielding vector $x_2 \in T$. Repeat until the collection $\{x_1, \dots, x_q\}$ contains a linearly independent subset of size N_m . This set determines T .
3. Determine the message part of Θ ; that is, determine $\Theta_{\mathbb{X}}$. Suppose in Step 2, $\{m_1, \dots, m_{N_m}\}$ is such that $\{x_1, \dots, x_{N_m}\}$ is linearly independent. If we write $\Theta = (\Theta_{\mathbb{X}}|\Theta_{\mathbb{Y}})$, we have the following system of equations:

$$(\Theta_{\mathbb{X}}|\Theta_{\mathbb{Y}})m_k \oplus 0 = x_k \text{ for } k = 1, \dots, N_m.$$

Given this system of equations, solve for $\Theta_{\mathbb{X}}$.

4. Decode ciphertexts. Given any ciphertext e , the adversary computes the following:

$$\begin{aligned} \alpha^{-1}(\Theta_{\mathbb{X}}|0)^T e &= \alpha^{-1}(\Theta_{\mathbb{X}}|0)^T (\Theta_{\mathbb{X}}|\Theta_{\mathbb{Y}})m \oplus g \\ &= \alpha^{-1}\Theta_{\mathbb{X}}^T \Theta_{\mathbb{X}} m \\ &= m \end{aligned}$$

where α is the square of the norm of any column of $\Theta_{\mathbb{X}}$. □

4.2 Concluding Remarks

The proposed cipher appears to be robust against brute force attacks, but is not robust against a chosen-plaintext attack. We mention, however, that we do not know if the scheme is robust against a known-plaintext attack.

Moreover, this is a private symmetric key cipher; it would be desirable if this method could be altered to be used as a public key cipher. We reiterate that the McEliece cipher is a public key system and is similar in flavor to the cipher presented here.

The ultimate downfall of the cipher is the linearity. We suggest that perhaps there is a way to introduce non-linearity into the algorithm to defeat a chosen-plaintext attack. However, at this point, we know of no methods to accomplish this.

After submission of this paper, we learned of a similar idea using frames for data hiding in [Miotke and Rebollo-Neira 04].

5. PSEUDOCODE

5.1 Encoder.cpp

1. Calculate Matrix.
 - (a) Input the possible range of entries for A, B, C.
 - (b) Make A, B, C either 4×4 or 8×8 Hadamard arrays with entries chosen randomly from the range (for simplicity, we are using the 4×4 Hadamard array).
 - (c) Compute tensor product $A \oplus B \oplus C$.
2. Encode Message.
 - (a) Compute $m \oplus g$ by converting the message to ASCII and filling g with random numbers.
 - (b) Compute $(A \otimes B \otimes C)(m \oplus g)$.

5.2 Hacker.cpp

This code attempts a brute force method on a cypher text.

1. Input min, max, range of key guesses.
2. Input ciphertext.
3. For all possible values of the twelve variables in use:
 - (a) fill the matrices with the possible values;
 - (b) tensor matrices together;
 - (c) calculate possible text messages;
 - (d) output text to file for later examination.

5.3 Analyzer.cpp

This code takes the output of Hacker.cpp and calculates the frequency of occurrence of every ASCII symbol.

1. For each line of text, count number of appearances of each ASCII value.
2. Output information to text file.

6. GRAPHS

We carried out the following computations to simulate a brute force attack on the cipher:

1. for a sample plaintext, encode the plaintext using Scheme 5, choosing the approximate entry size for the matrices and the approximate size for the garbage entries;
2. decode the ciphertext using every combination of key entry and key entry ± 1 ;
3. convert the decoded ciphertext in the previous step to ASCII values;
4. count the appearance of each value in the resulting combinations.

The graphs represent the number of appearances within all possible key guesses from Step (2) above. The plaintext is given above the graph; the ASCII values are on the x -axis, and the approximate key sizes and garbage sizes are given in the captions.

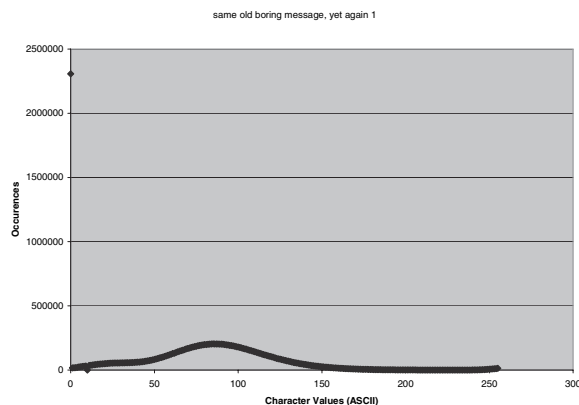


FIGURE 2. Key: 5–7; Garbage: 128.

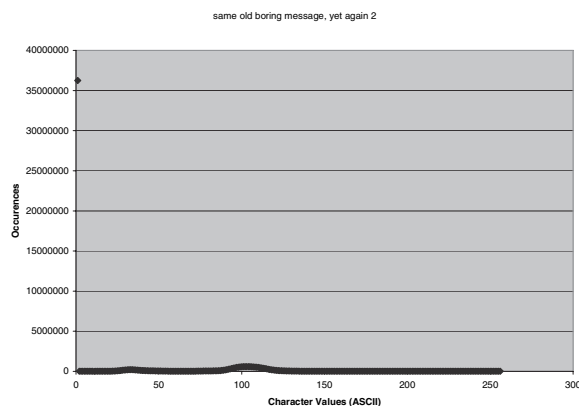


FIGURE 3. Key: 25–27; Garbage: 128.

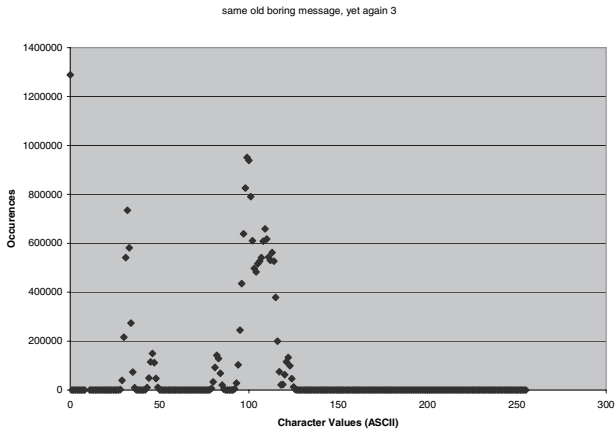


FIGURE 4. Key: 100–102; Garbage: 128.

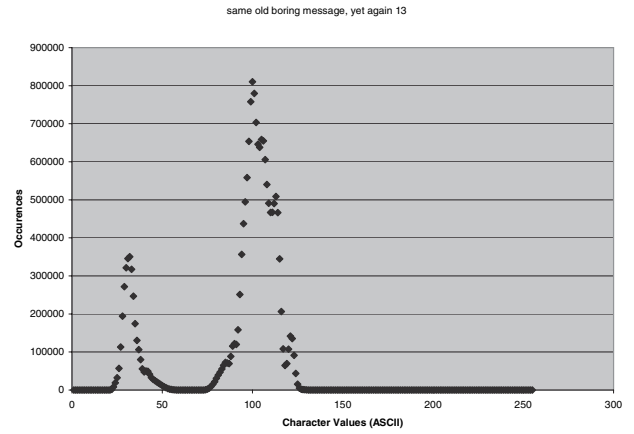


FIGURE 7. Key: 100–102; Garbage: 1,000.

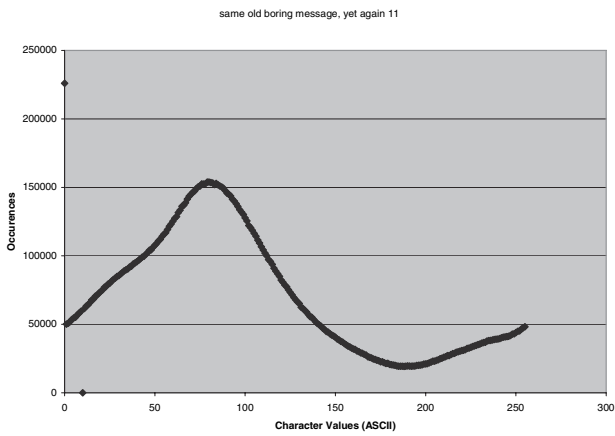


FIGURE 5. Key: 5–7; Garbage: 1,000.

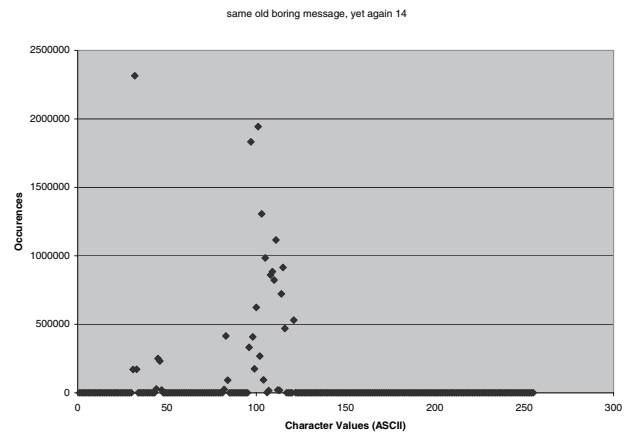


FIGURE 8. Key: 1,000–1,002; Garbage: 1,000.

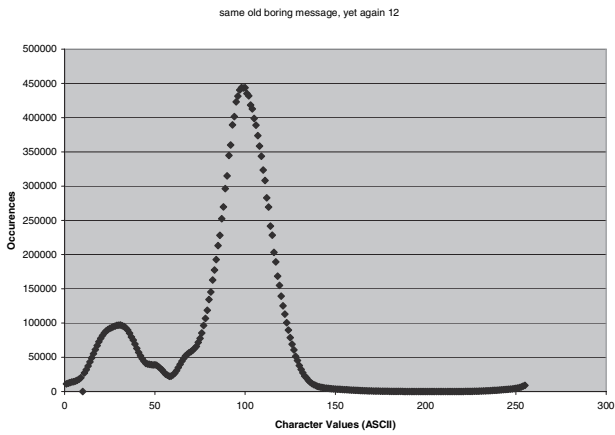


FIGURE 6. Key: 25–27; Garbage: 1,000.

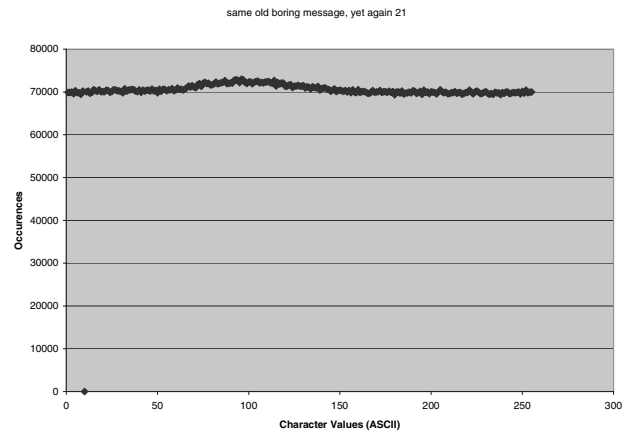


FIGURE 9. Key: 5–7; Garbage: 100,000.

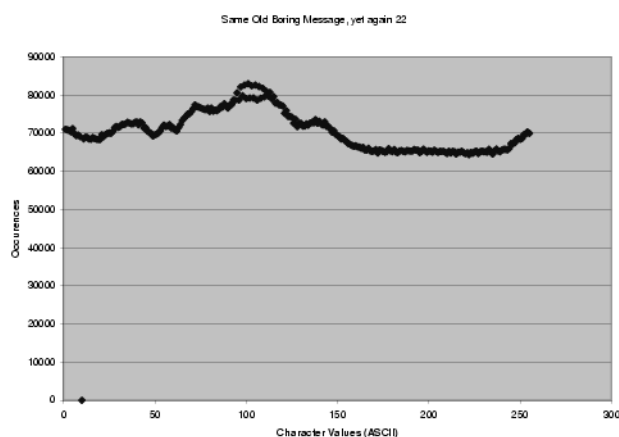


FIGURE 10. Key: 25–27; Garbage: 100,000.

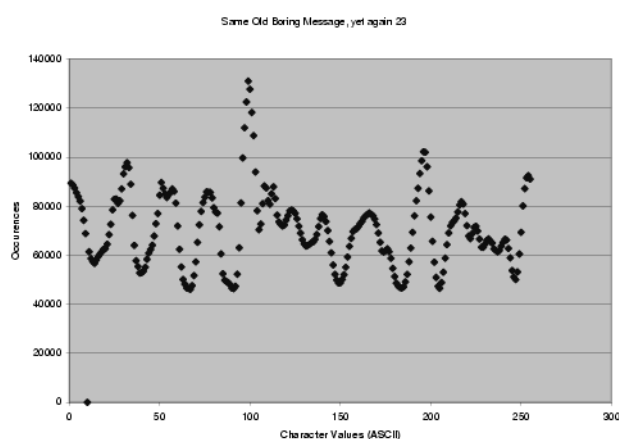


FIGURE 11. Key: 100–102; Garbage: 100,000.

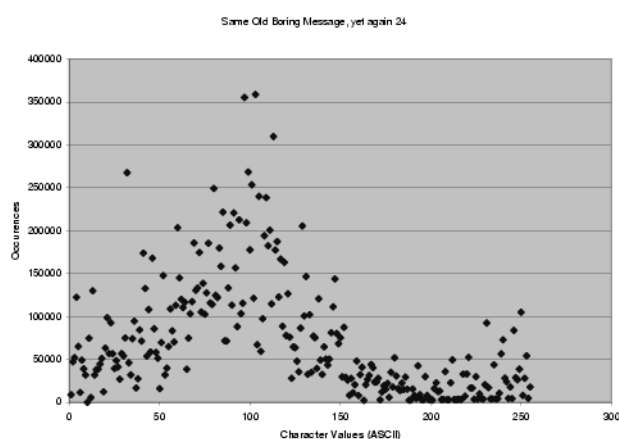


FIGURE 12. Key: 1,000–1,002; Garbage: 100,000.

Note that in Figures 4 and 8, the key size and garbage size are the same. The graphs show that most of the characters that appear in the simulated brute force attack are those that are in the original message.

ACKNOWLEDGMENTS

This work was done while all three authors were at the University of Wyoming, at which time the first author was an undergraduate student, and the third author was a graduate student. The first and second authors were supported by NSF grant DMS-0308634. The third author was supported by a Basic Research Grant from the University of Wyoming.

We thank Bryan Shader, Eric Moorhouse, and Cliff Bergman for helpful discussions.

REFERENCES

- [Agaian 85] S. S. Agaian. *Hadamard Matrices and Their Applications*. New York: Springer-Verlag, 1985.
- [Aldroubi et al. 04] A. Aldroubi, D. Larson, W. S. Tang, and E. Weber. “Geometric Aspects of Frame Representations of Abelian Groups.” *Trans. Amer. Math. Soc.* 356:12 (2004), 4767–4786.
- [Benedetto et al. 03] J. Benedetto and M. Fickus. “Finite Normalized Tight Frames.” *Adv. Comput. Math.* 18:2–4 (2003), 357–385.
- [Casazza 00] P. Casazza. “The Art of Frame Theory.” *Taiwanese Math. J.* 4:2 (2000), 129–201.
- [Casazza and Kovacević 01] P. Casazza and J. Kovacević. “Uniform Tight Frames for Signal Processing and Communications.” *SPIE Proc.* 4478 (2001), 129–135.
- [Chabaud 95] F. Chabaud. “On the Security of Some Cryptosystems Based on Error-Correcting Codes.” In *Advances in Cryptology – EUROCRYPT ’94*, pp. 131–139, Lecture Notes in Computer Science, 950. Berlin: Springer-Verlag, 1995.
- [Delsarte and Goethals 69] P. Delsarte and J. M. Goethals. “Tri-Weight Codes and Generalized Hadamard Matrices.” *Information and Control* 15 (1969), 196–206.
- [Dykema et al. 04] K. Dykema, D. Freeman, K. Kornelson, D. Larson, M. Ordower, and E. Weber. “Ellipsoidal Tight Frames and Projection Decompositions of Operators.” *Illinois J. Math.* 48:2 (2004), 477–489.
- [Han and Larson 00] D. Han and D. Larson. “Frames, Bases, and Group Representations.” *Mem. Amer. Math. Soc.* 147:697 (2000), 94 pp.
- [van Lint 92] J. H. van Lint. *A Course in Combinatorics*. Cambridge, UK: Cambridge University Press, 1992.
- [Menezes et al. 97] A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. Orlando, FL: CRC Press, 1997.
- [Miotke and Rebollo-Neira 04] J. R. Miotke and L. Rebollo-Neira. “Oversampling of Fourier Coefficients for Hiding Messages.” *Appl. Comput. Harmon. Anal.* 16:3 (2004), 203–207.
- [Wallis 72] W. D. Wallis. *Combinatorics: Room Squares, Sum-Free Sets, Hadamard Matrices*. Berlin: Springer-Verlag, 1972.

Ryan Harkins, Department of Mathematics, University of Wyoming, Laramie, WY 82071-3036
(narandus@hotmail.com)

Eric Weber, Department of Mathematics, Iowa State University, 400 Carver Hall, Ames, IA 50011
(esweber@iastate.edu)

Andrew Westmeyer, Department of Mathematics, University of Wyoming, Laramie, WY 82071-3036
(westandy@uwyo.edu)

Received May 17, 2004; accepted June 13, 2005.