*Applied Mathematics & Information Sciences* **5**(3) (2011), 547-557
– An International Journal


# Encryption schemes using Plotkin arrays

C. Koukouvinos and D. E. Simos
Department of Mathematics, National Technical University of Athens
Zografou 15773, Athens, Greece
*Email Addresses: ckoukouv@math.ntua.gr and dsimos@math.ntua.gr*

Received January 15, 2011; Accepted April 21, 2011

In this paper we propose a cipher similar to the One Time Pad and Hill cipher based on a subband coding scheme using Plotkin arrays. The encoding process is an approximation to the One Time Pad encryption scheme. We present experimental results which suggest that a brute force attack to the proposed scheme does not result in all possible plaintexts, as the One Time Pad does, but still the brute force attack does not compromise the system.

**Keywords:** Cipher, cryptography, encryption, orthogonal designs, Plotkin arrays.

**2000 Mathematics Subject Classification**: 94A60, 68P25, 05B15, 05B20.

## 1 Introduction

In this paper, we propose a private key cipher based on several constructions that have arisen using Plotkin arrays. We were motivated to use Plotkin arrays though they are part of a wider class, called combinatorial designs which are often hard to find and the algorithms for encryption and decryption are of reasonable length. For encryption methods based on combinatorial designs we refer the interested reader to [12]. Applications of combinatorial designs to communications, cryptography and networking can be found in the survey paper, [2]. The cipher has similarities to the Hill cipher and to the One Time Pad [10, 16]. A comprehensive presentation of the aforementioned ciphers can be found in the book ( [16]). A list of typical protocol attacks and reference of the existing protocols can be found in ( [1]). Indeed, one of the design goals for our cipher is to approximate the One Time Pad. Our design goals include the following:

The research of the second author was supported by a scholarship awarded by the Secretariat of the Research Committee of National Technical University of Athens.

1. Include randomness in the encryption process.
2. Require the key be shared only once.
3. Use a relatively small key size
4. Computationally fast
5. Robust to brute force attacks.

Our proposed cipher implements the first four goals, the purpose of the present paper is to illustrate that the cipher is robust to brute force attacks.

Our cipher can be described from the following procedure: consider a communications channel, we divide the channel into two subbands, one which will carry the message, and the other which will carry noise. The message, along with the noise is transmitted over the channel. The recipient then filters out the noise, leaving only the message. This procedure is carried out using orthogonal matrices, specifically Plotkin arrays.

This paper can be regarded as an alternative to the proposed schemes given in [5, 6], and it is organized as follows. In Section 2, we construct several encryption schemes using Plotkin arrays, according to our desing goals. In Section 3 we give pseudocode for an implementation of the encryption algorithm while in Section 4 we present the results and conclusions of our numerical experiments and the brute force attack.

## 2 Encryption schemes

We are interested in constructing encryption schemes using orthogonal matrices. This procedure is accomplished using Plotkin arrays which allow us to generate large orthogonal matrices. We give the necessary brief definitions for encryption schemes and orthogonal designs, a superset of Plotkin arrays.

**Definition 1 (C. Boyd et al. , [1]).** An encryption scheme consists of three sets: a key set $K$, a message set $M$, and a ciphertext set $C$ together with the following three algorithms.

1. A key generation algorithm, which outputs a valid encryption key $k \in K$ and a valid decryption key $k^{-1} \in K$.
2. An encryption algorithm, which takes an element $m \in M$ and an encryption key $k \in K$ and outputs an element $c \in C$ defined as $c = E_k(m)$.
3. A decryption function, which takes an element $c \in C$ and a decryption key $k^{-1} \in K$ and outputs an element $m \in M$ defined as $m = D_k^{-1}(c)$. We require that $D_k^{-1}(E_k(m)) = m$.

**Definition 2 (Brute force attack).** A brute force attack is a method of defeating a cryptographic scheme by trying a large number of possibilities. For most ciphers, a brute force attack typically means a brute-force search of the key space; that is, testing all possible keys in order to recover the plaintext used to produce a particular ciphertext.

We note that any attack on an encryption scheme is only valid if it violates some property that the scheme was intended to achieve. In other words all attacks must be considered relative to the design goals that the encryption scheme is meant to achieve.

**Remark 1.** Since we have included to our design goals the cipher to be robust in brute force attacks, one definition of breaking the cryptographic scheme is either to find a method faster than a brute force attack or the brute force attack is able to recover the plaintext that was used in order to produce a particular ciphertext in a reasonable computational time.

**Definition 3.** An *orthogonal design* of order $n$ and type $(s_1, s_2, \ldots, s_k)$ denoted $OD(n; s_1, s_2, \ldots, s_k)$ in the commuting variables $x_1, x_2, \ldots, x_k$, is a square matrix $D$ of order $n$ with entries from the set $\{0, \pm x_1, \pm x_2, \ldots, \pm x_k\}$ satisfying

$$DD^T = \sum_{i=1}^{k} (s_i x_i^2) I_n,$$

where $I_n$ is the identity matrix of order $n$.

**Definition 4.** A *Hadamard matrix* of order $n$ is an $n \times n$ $\{1, -1\}$-matrix satisfying $HH^T = nI_n$.

It is well known that if $n$ is the order of a Hadamard matrix then $n$ is necessarily $1, 2$ or a multiple of $4$. Orthogonal designs are also used in Combinatorics, Statistics, Coding Theory, Telecommunications and other areas. More details on orthogonal designs and Hadamard matrices can be found in [14, 15]. The last definitions give us the following insights:

1. In any row there are $s_1$ entries $\pm x_1$, $s_2$ entries $\pm x_2$, . . ., $s_k$ entries $\pm x_k$, and similarly for the columns.

2. The rows and columns are pairwise orthogonal, respectively.

The choice of orthogonal designs for constructing orthogonal matrices and afterwards encryption schemes enable us to choose between a large variety of classes of orthogonal designs with different structure. Plotkin [11] showed that, if there is an Hadamard matrix of order $2t$, then there is an $OD(8t; t, t, t, t, t, t, t, t)$. It is conjectured that there is an $OD(8n; n, n, n, n, n, n, n, n)$ for each odd integer $n$. These orthogonal designs are called, *Plotkin arrays*.

**Scheme 1** We initiate the construction of our first scheme based on Plotkin arrays. As an example, we illustrate the construction based on the Plotkin array of order $8$ and type $(1,1,1,1,1,1,1,1)$. The corresponding orthogonal design is the following:

$$OD(8;1,1,1,1,1,1,1,1) = \begin{pmatrix} A & B & C & D & E & F & G & H \\ -B & A & D & -C & F & -E & -H & G \\ -C & -D & A & B & G & H & -E & -F \\ -D & C & -B & A & H & -G & F & -E \\ -E & -F & -G & -H & A & B & C & D \\ -F & E & -H & G & -B & A & -D & C \\ -G & H & E & -F & -C & D & A & -B \\ -H & -G & F & E & -D & -C & B & A \end{pmatrix}, \tag{2.1}$$

If we call the above matrix P, we have that $PP^T = fI_8$ whereas $f = A^2 + B^2 + \ldots + H^2$. The Plotkin arrays allow easy construction of matrices needed in our encryption schemes. For the encryption process we have only to compute the matrix P. The encryption process starts with a message $m$ of arbitrary length, and dividing $m$ into blocks $m_1, \ldots, m_q$ of length $4$ (padding the last block with zeros if necessary). Then random vectors $g_1, \ldots, g_q$ of length $4$ are chosen. For the construcion of noise vectors $g_1, \ldots, g_q$ pseudorandom generators were constructed using techniques from [7]. Finally, the matrix P is applied succesively to $m_i \oplus g_i$. The ciphertext is then $c = P(m_1 \oplus g_1) \oplus \ldots \oplus P(m_q \oplus g_q)$. The notation $m \oplus g$ means that $m$ is concatenated with $g$.
The message is then decrypted by dividing $c$ into blocks $c_1, \ldots, c_q$ of size 8, computing $P^T c_i/f$ for $i = 1, \ldots, q$ and reconstructing the message using the first four entries of these blocks.

**Remark 2.** The key for the recipient is the chosen entries for P, hence in this case is the entries $A, B, \ldots, H$ of the matrix P.

Since the Plotkin array we used so far is relatively small, we continued by modifying appropriate the encryption process using the Plotkin array of orders $16$ and $24$. We note that the use of Plotkin arrays of different orders does not result in an increase to the key search space since the number of variables that appear in the aforementioned orthogonal designs remains the same. Experimental results from the above runs are presented in Section 4. The aforementioned orthogonal designs can be found in the book ( [4]).

**Scheme 2** Although Plotkin arrays exists for larger orders we wanted a more sophisticated procedure for the encryption process which would be able to provide us with a more robust cipher against a brute force attack.

We first define the tensor product $A \otimes B$ between two matrices $A$ and $B$ a crucial definition for the construction of this scheme.

**Definition 5 (van Lint, [8]).** Let $A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & & \ddots & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$

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

If $A$ is an $m \times n$ and $B$ is an $p \times q$ matrix, then $A \otimes B$ is an $mp \times nq$ matrix. We note that if $A$ and $B$ are orthogonal matrices, then $A \otimes B$ is also an orthogonal matrix.

For the encryption process we choose $p$ Plotkin arrays $P_1, P_2, \dots, P_p$. Each array may have different size, let say $e_i \times e_i$ for $1 \leq i \leq p$ where each $e_i$ may be $8, 16$ or $24$. We then construct an $e_1 e_2 \dots e_p$ sized matrix $M$ by the tensor product of these $p$ matrices:

$$M = \bigotimes P_i := P_1 \otimes P_2 \otimes \dots \otimes P_p.$$

The ciphertext then is $c = M(m \oplus g)$. With this construction we eliminate any possible sparsity of zeros in the encryption matrix $M$. We note that the key in this case is the entries of the first rows of $P_1$ to $P_p$, hence is an array of numbers of size $e_1 + e_2 + \dots + e_p$ and therefore it is relatively small. The notation $m \oplus g$ means that $m$ is concatenated with $g$.

We can now discuss, a weakness in the design of the first encryption scheme which in some cases can be eliminated using the previous construction based on the tensor product of orthogonal matrices. It was mentioned earlier that in cases the plaintext $m$ has more than $n$ letters, we repeat the encryption process. This method, is also known as the *electronic codebook* mode, or ECB in the literature ( [3, 9, 10, 17]). A disadvantage of this method is that if two plaintext blocks are the same, then the corresponding ciphertext blocks will be identical, and that is visible to the attacker.

The tensor product construction of the second scheme can reduce the amount of information that can be retrieved from a potential attacker when using ECB mode by restricting the available choices for Plotkin arrays $P_i$, $i = 1, \dots, p$ to be $P_f \neq P_g$ for $i \leq f, g \leq p$ with $f \neq g$. In general, if we choose the $P_i$ encryption matrices to have $\sum_{i=1}^{p} n_i = n$, where $n$ is the size of the plaintext this weakness is eliminated since the encryption process does not have any repetition blocks.

## 3 Encryption algorithm

In this Section we give a brief presentation of the algorithm we used for the encryption, decryption and analysis of the results in terms of pseudocode. The algorithm we developed

for the encoding process is implemented in the encoder function. The hacker function is an implementation to simulate a brute force attack. Finally for the analysis of the results we implemented the analyzer function. The implementation has been performed in C programming language. Programming techniques concerning cryptographic algorithms can be found in ( [13]).

**EncoderSchemeFunction** Encodes a sample plaintext using the second encryption scheme.

**Step 1. Compute the encryption matrix M**

**Step 1a.** Convert the corresponding characters of the plaintext to ASCII values.

**Step 1b.** Input the possible range of entries for the matrices $P_i$.

**Step 1c.** Choose the corresponding Plotkin arrays that will form the matrices $P_i$.

**Step 1d.** Compute the tensor product $M := P_1 \otimes P_2 \otimes \ldots \otimes P_p$.

**Step 2. Encode the input message**

**Step 2a.** Compute $m \oplus g$ by converting the message to ASCII values and filling the noise vector $g$ with random numbers.

**Step 2b.** Compute $M(m \oplus g)$

**HackerSchemeFunction** Simulation of a brute force attack method to a ciphertext.

**Step 1.** Input min, max and range of key guesses.

**Step 2.** Input ciphertext.

**Step 3. Exhaustive key search with respect to Step 1.**
For all possible values of the variables of the orthogonal designs chosen for encryption perform the following steps.

**Step 3a.** Generate the matrices using as entries the possible values from previous step.

**Step 3b.** Compute the tensor product of the matrices created in previous step.

**Step 3c.** Calculate possible text messages.

**Step 3d.** Output text to file for later examination.

**AnalyzerSchemeFunction** Receives the output from the hacker function and calculates the frequency of occurence of every ASCII symbol.

**Step. 1** For each line of text, count number of appearances of each ASCII value.

**Step. 2** Output information to text file.

## 4 Experimental results and analysis for cryptanalytic attacks

We conducted some numerical experiments for the two encryption schemes that were presented in Section 2. Although we wanted to remain focus to the designs goals described in Section 1, we conducted an analysis of known-plaintext attacks for both encryption schemes in the end of this Section. The experimental results presented in this Section concern simulations of brute force attacks to the two encryption schemes presented in Section 2, and the analysis involve an approach to simulate known-plaintext attacks for both schemes.

### 4.1 Simulation of brute force attacks for the first encryption scheme

To carry a brute force attack on the first encryption scheme we carried the following steps for each simulation.

1. We used a sample plaintext of $384$ characters and a random noise vector of the same length.
2. We considered the entries of $A, B, \ldots, H$ as binary variables.
3. We decoded the ciphertext using every key combination of key entry and key entry value equal to $\pm 1$.

From the experimental results we received from the first encryption scheme we obtained the following information:

1. For the Plotkin arrays $OD(8t; t, t, t, t, t, t, t, t)$ for $t = 8, 16, 24$ a brute force attack resulted in a thorough defeat of the cipher. We mention though, that the computational time grows in a non-linear way.
2. Since, this scheme is not robust against brute attacks we have a complete violation to one of the design properties we set in Section $1.$ for this encryption scheme.

The following table presents the computational results for the simulations we conducted. For each orthogonal design we give the size of the key search space and the elapsed CPU time needed for a brute force attack to break the system.

| Design | Key Search Space | Elapsed CPU Time |
|---|---|---|
| $OD(8;1,1,1,1,1,1,1,1)$ | $2^8$ | $4$ hours |
| $OD(16;2,2,2,2,2,2,2,2)$ | $2^8$ | $12$ hours |
| $OD(24;3,3,3,3,3,3,3,3)$ | $2^8$ | $34$ hours |

Table 4.1: Experimental results received from a brute force attack on first scheme.

### 4.2 Simulation of brute force attacks for the second encryption scheme

To carry a brute force attack on the second encryption scheme we carried the following steps for each simulation.

1. We used a sample plaintext of $23$ characters.
2. We encoded the plaintext using the second scheme by approximating the entry size for the Plotkin arrays and approximate size of the noise vector $g$.
3. We used the Plotkin arrays of order 8 to compute the encryption matrix $M$.
4. We decoded the ciphertext using every key combination of key entry and key entry value equal to $\pm 1$.
5. We converted the decoded ciphertext found in the previous step to ASCII values.
6. We counted the frequency of each value that appears in the resulting combinations.

From the experimental results we received from the second encryption scheme we obtained the following information:

1. A brute force attack is not a feasible way of defeating the cipher.
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 for our cipher. The computational results indicate that the answer is no.
3. Finally we wanted to determine if the size of the entries of the noise vector $g$ played a significant role in the decryption process. The computations showed that the answer is yes.
4. All five design goals are fullfiled for this encryption scheme.

The following table presents the computational results for the simulations we conducted. For each simulated brute force attack we give the number of occurences of the ASCII values in their coresponding range and the approximate key and noise vector sizes. The table shows that most of the characters that appear in the simulated brute force attack are those that have been encoded using the sample plaintext.

| key size | noise size | ASCII values occurences$\times 10^5$ | | | | |
|---|---|---|---|---|---|---|
| | | $0-25$ | $26-50$ | $51-75$ | $76-100$ | $101-127$ |
| 10-14 | 128 | 25 | 5 | 5 | 7 | 8 |
| 10-14 | 1024 | 10 | 12 | 8 | 6 | 14 |
| 30-34 | 128 | 120 | 30 | 40 | 30 | 50 |
| 30-34 | 1024 | 65 | 90 | 45 | 50 | 40 |
| 50-54 | 128 | 310 | 50 | 70 | 30 | 40 |
| 50-54 | 1024 | 110 | 100 | 90 | 80 | 120 |

Table 4.2: Experimental results received from a brute force attack on second scheme.

### 4.3 Analysis of known-plaintext attacks for both schemes

**Definition 6 (Known-plaintext attack).** A known-plaintext attack is one where the adversary has a quantity of plaintext and corresponding ciphertext. This type of attack is typically only marginally more difficult to mount.

Supposing a $n \times n$ matrix $P$ is used for encryption, as described in the design of both encryption schemes. In order to recover the matrix $P$ without knowing the private key, we will need $n$ $\overline{m}^i$'s, where with $\overline{m}^i = (m_1^i,\ m_2^i,\ \ldots,\ m_n^i),\ i = 1,\ \ldots,\ n$ we denote the vector consisting of $n$ letters of the message that have been converted to its numerical values, and $n$ $\overline{c}^i$'s, where each $\overline{c}^i = (c_1^i,\ c_2^i,\ \ldots,\ c_n^i)$ is the encryption of $\overline{m}^i$. The i-th column of $P$, $P(i) = (p_{1,i},\ p_{2,i},\ \ldots,\ p_{n,i})$, by solving the following $n$-linear systems, for $i = 1,\ \ldots,\ n$:

$$\begin{aligned} m_1^1 p_{1,i} + m_2^1 p_{2,i} + \cdots + m_n^1 p_{n,i} &= c_i^1 \\ m_1^2 p_{1,i} + m_2^2 p_{2,i} + \cdots + m_n^2 p_{n,i} &= c_i^2 \\ \vdots \qquad\qquad &\vdots \\ m_1^n p_{1,i} + m_2^n p_{2,i} + \cdots + m_n^n p_{n,i} &= c_i^n \end{aligned}$$

or equivalently we denote the previous system

$$MP(i) = C(i)\ ,$$

where $C(i) = (c_i^1,\ c_i^2,\ \ldots,\ c_i^n)$.

**Proposition 1.** All encryption schemes using Plotkin arrays are secure against known-plaintext attacks under the assumption that the adversary has knowledge of less than $n$ messages of length $n$ of the plaintext and the corresponding ciphertext.

**Proof** With the method described previously one can find the encryption matrix $P$, if the matrix $M$ is non-singular. □

## 5 Conclusion

We propose a private symmetric cipher based on the second encryption scheme presented in this paper, that appears to be robust to brute force attacks, and in some cases robust against known-plaintext attacks. Perhaps, an evolutionary decryption algorithm based on the selection and reproduction of the entries of the corresponding orthogonal designs could result in better performance to a brute force attack but this is currently beyond our scopes.

## References

[1] C. Boyd and A. Mathuria, *Protocols for Authentication and Key Establishment*, Information Security and Cryptography Series, Springer-Verlag, 2003.

[2] C.J. Colbourn, J.H. Dinitz and D.R. Stinson, Applications of Combinatorial Designs to Communications, Cryptography, and Networking, in *Surveys in Combinatorics*, J.D. Lamb and D.A. Preece (Eds.), Cambridge, Cambridge University Press, pp. 37–100.

[3] N. Ferguson and B. Schneier, *Practical Cryptography*, Wiley Publishing, Inc., 2003.

[4] A.V. Geramita, and J. Seberry, *Orthogonal Designs: Quadratic Forms and Hadamard Matrices*, Marcel Dekker, New York-Basel, 1979.

[5] R. Harkins, E. Weber and and A. Westmeyer, Encryption Schemes using Finite Frames and Hadamard Arrays, *Experimental Mathematics* **14** (2005), 423-433.

[6] C. Koukouvinos, E. Lappas and D. E. Simos, Encryption schemes using orthogonal arrays, *J. Discrete Math. Sci. Cryptogr.* **12** (2009), 615-628.

[7] M. Luby, *Pseudorandomness and Cryptographic Applications*, Princeton Academic Press, Princeton, 1996.

[8] J.H. van Lint and R.M. Wilson, *A Course in Combinatorics*, Cambridge, Cambridge University Press, 1992.

[9] W. Mao, *Modern Cryptography: Theory and Practice*, Prentice Hall, 2004.

[10] A. Menezes, P. van Oorschot and S. Vanstone, (Eds.), in *CRC Handbook of Applied Cryptography*, CRC Press, 1997.

[11] M. Plotkin, Decomposition of Hadamard matrices, *J. Combin. Theory, Ser. A* **13** (1972), 127–130.

[12] D.G. Sarvate and J. Seberry, Encryption methods based on Combinatorial Designs, *Ars Combinatoria* **21-A** (1986), 237–246.

[13] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd Edition, J. Wiley and Sons Inc., 1996.

[14] J. Seberry and R. Craigen, Orthogonal designs, in *CRC Handbook of Combinatorial Designs*, C.J. Colbourn and J.H. Dinitz (Eds.), CRC Press, Boca Raton, (1996), pp. 400–406.

[15] J. Seberry and M. Yamada, Hadamard matrices, sequences and block designs, in *Contemporary Design Theory: A Collection of Surveys*, J.H. Dinitz and D.R. Stinson (Eds.), J. Wiley and Sons, New York, (1992), pp. 431–560.

[16] W. Stallings, *Cryptography and Network Security: Principles and Practices*, 3rd Edition, Prentice Hall, 2003.

[17] D.R. Stinson, *Cryptography: Theory and Practice*, 3rd Edition, CRC Press, 2005.

---

Christos Koukouvinos is a Professor at the National Technical University of Athens, Department of Mathematics. He holds a Bachelor in Mathematics (1983) and a PhD (1988) in Statistics both obtained from the University of Thessaloniki. He is the author of numerous papers in international peer reviewed journals in the field of Statistics and Combinatorics. He is on the Editorial Board of several international journals. He was awarded the prestigious Hall Medal of the Institute of Combinatorics and its Applications (ICA) in 1996. He is a Fellow of the ICA and was member of Council of the ICA from 2000 to 2003. His research interests include statistical experimental and optimal designs, biostatistics, statistical quality control, combinatorial designs and coding theory.

Dimitris E. Simos received his Bachelor in Mathematics (2006) from the University of Athens, and MSc in Applied Mathematical Sciences (2007) from the National Technical University of Athens. He is currently a PhD candidate at National Technical University of Athens since 2008. He is the author of several papers in international peer reviewed journals in the field of Combinatorics. He is an Associate Fellow of the Institute of Combinatorics and its Applications (ICA). His research interests include combinatorial designs, coding theory, cryptography, symbolic computation and heuristics.