# Finding the Position of a Subarray in a Pseudo-random Array

Sheelagh Lloyd, John Burns
Networks and Communications Laboratory
HP Laboratories Bristol
HPL-91-159
October, 1991

position-finding, pseudo-random arrays, discrete logarithms

Pseudo-random arrays have the property that each possible subarray of a certain size except the all zero one occurs exactly once in the array. They are the two-dimensional analogue of pseudo-random sequences. These sequences are used in a number of position-finding applications, and pseudo-random arrays could also be useful in such applications. The problem is, given a subsequence (or subarray), to determine its position in the sequence (or array). Until recently, the only solution to this problem was the clever use of a combination of look-up tables and generation of subsequent subsequences. In this paper, we present a novel approach to this problem, and show how to reduce it to the well-known one of discrete logarithms.

Internal Accession Date Only

# Finding the position of a subarray in a pseudo-random array

**Sheelagh Lloyd and John Burns**

*Hewlett-Packard Laboratories*
*Filton Road*
*Stoke Gifford*
*Bristol*
*BS12 6QZ*

## 1 Introduction

Pseudo-random arrays have the property that each possible $k_1 \times k_2$ subarray except the all zero one occurs exactly once in the array. They are the two-dimensional analogue of pseudo-random sequences. These sequences are used, for example, in measuring the absolute position of automated guided vehicles (Basran et al. 1989), and pseudo-random arrays could also be useful in such applications. Another application for both the one and two dimensional cases is described in (Burns and Mitchell 1991). The problem is, given a subsequence (or subarray), to determine its position in the sequence (or array). Until recently, the only solution to this problem was the clever use of a combination of look-up tables and generation of subsequent subsequences (Basran et al. 1989). The idea was to keep a table of subsequences at known positions spaced out through the sequence and, given a subsequence, to generate the subsequences which follow it until one of the subsequences in the table is found. Recently, however, Paterson (Paterson 1991) has solved this problem in the one-dimensional case by reducing it to the well-known one of computing discrete logarithms in $GF(2^k)$, where the length of the pseudo-random sequence is $2^k - 1$. The discrete logarithm problem is that of, given a primitive element $\alpha$ and an element $\beta$, finding an integer $r$ such that $\beta = \alpha^r$. In this paper, we present the extension to two dimensions of

Paterson's method, again reducing the problem to that of computing discrete logarithms in $GF(2^k)$, where the number of elements in the array is $2^k - 1$.

The remainder of this paper is organised as follows. In Section 2, we shall describe the construction, due to MacWilliams and Sloane (MacWilliams and Sloane 1976), of a pseudo-random array (PRA) from a pseudo-random sequence (PRS). In Section 3, we look at these PRAs in more detail, and in particular at the linear recurrences which generate them. We shall then, in Section 4, describe an isomorphism between two finite fields which will enable us to derive some further properties of these recurrences. The reduction of the problem of finding the position of a subarray to that of discrete logarithms is described in Section 5, and Section 6 is devoted to a small example which illustrates the method.

## 2 Constructing a PRA from a PRS

The construction in this section is due to MacWilliams and Sloane (MacWilliams and Sloane 1976). Let $m = k_1 k_2$, and suppose that $n = 2^m - 1$ is such that $n_1 = 2^{k_1} - 1$ and $n_2 = \frac{n}{n_1}$ are relatively prime and greater than one. Let $\underline{a} = (a_0, a_1, \ldots, a_{n-1})$ be a pseudo-random sequence of length $n$. Then $\underline{a}$ is generated by a primitive polynomial $h(x) = h_0 + h_1 x + \ldots + h_{m-1}x^{m-1} + x^m$, that is, for any $t$,

$$a_{m+t} = h_0 a_t + h_1 a_{t+1} + \ldots + h_{m-1} a_{t+m-1}.$$

Then a pseudo-random array of size $n_1 \times n_2$ is constructed by putting $\underline{a}$ down the main diagonal and continuing from the opposite side whenever an edge is reached. So, for example, if $m = 4 = 2 \times 2$, then $n = 15 = 3 \times 5$, and the sequence (0,0,0,1,0,0,1,1,0,1,0,1,1,1,1) gives rise to the array

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

It is known (Green 1985) that the construction of a PRA is equivalent to the improper decimation of the PRS $\underline{a}$. In other words, there exist integers $R$

and $T$, defined below, such that the array can be written as

$$\begin{pmatrix} a_0 & a_R & a_{2R} & \cdots & a_{T-1} \\ a_T & a_{T+R} & a_{T+2R} & \cdots & a_{2T-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{R-1} & a_{2R-1} & a_{3R-1} & \cdots & a_{n-1} \end{pmatrix}.$$

If $\phi$ denotes Euler's phi function, then $R$ and $T$ are defined by

$$R = n_1 x, \quad T = n + 1 - R, \quad x \equiv n_1^{\phi(n_2)-1} \pmod{n_2}.$$

Note that

$$\begin{aligned} R &\equiv 0 \pmod{n_1} \quad , \quad R \equiv 1 \pmod{n_2} \\ T &\equiv 1 \pmod{n_1} \quad , \quad T \equiv 0 \pmod{n_2} \end{aligned}$$

These arrays have many properties (see, for example (MacWilliams and Sloane 1976)). The property of most interest to us is the windowing property: that, apart from the all zero array, every possible $k_1 \times k_2$ subarray appears exactly once in the array. The other property which will be useful to note is that, apart from one column which is all zeros, the columns of the array are all shifted copies of a pseudo-random sequence of length $2^{k_1} - 1$.

## 3 Linear Recurrences

Since a PRA is formed by folding a PRS along the diagonals, the PRA satisfies the recurrence which generates the original PRS along the diagonals. It has been shown (MacWilliams and Sloane 1976) that this recurrence can be converted into two recurrences, one for moving vertically and one for moving horizontally. Let us write these recurrences as

$$\begin{aligned} a_{i+k_1,j} &= \sum_{r=0}^{k_1-1} f_r a_{i+r,j} \\ a_{i,j+k_2} &= \sum_{r=0}^{k_1-1} \sum_{s=0}^{k_2-1} g_{r,s} a_{i+r,j+s} \end{aligned}$$

The reason that the recurrence for moving vertically is so much simpler than that for moving horizontally is that, as mentioned above, each of the columns of the array (except the all zero column) is a shifted copy of the same pseudo-random sequence of length $2^{k_1} - 1$. A method for deriving the values of the $f_r$ and $g_{r,s}$ from the coefficients of the generating polynomial $h$ of the original pseudo-random sequence is given in (Shi and Chen 1988).

We would like to write these recurrences as matrices, and we do this in the following way. If we write a $k_1 \times k_2$ subarray of $A$ as a vector of length $k_1 k_2$ by writing each of the columns out in turn so that

$$\begin{pmatrix} a_{i,j} & a_{i,j+1} & \cdots & a_{i,j+k_2-1} \\ a_{i+1,j} & a_{i+1,j+1} & \cdots & a_{i+1,j+k_2-1} \\ \vdots & \vdots & \vdots & \vdots \\ a_{i+k_1-1,j} & a_{i+k_1-1,j+1} & \cdots & a_{i+k_1-1,j+k_2-1} \end{pmatrix}$$

is written as

$$\begin{pmatrix} a_{i,j} \\ a_{i+1,j} \\ \vdots \\ a_{i+k_1-1,j} \\ a_{i,j+1} \\ a_{i+1,j+1} \\ \vdots \\ a_{i+k_1-1,j+1} \\ \vdots \\ a_{i,j+k_2-1} \\ a_{i+1,j+k_2-1} \\ \vdots \\ a_{i+k_1-1,j+k_2-1} \end{pmatrix}$$

then each of the two recurrences can be written as a $k_1 k_2 \times k_1 k_2$ matrix which acts on these vectors. Let the matrix for the vertical recurrence be $C$ and the one for the horizontal recurrence be $D$. Now if we denote by $\underline{s}_{r,t}$ the vector corresponding to the subarray with top left hand corner at $(r, t)$, we have

$$\underline{s}_{r,t} = C^r D^t \underline{s}_{0,0}$$

and we have the following result.

**Lemma 3.1** The matrices $C$ and $D$ commute.

Since the columns are copies of a pseudorandom sequence, $C$ in fact consists of $k_2$ copies of a $k_1 \times k_1$ matrix $F$ arranged along the diagonal:

$$C = \begin{pmatrix} F & 0 & \dots & 0 \\ 0 & F & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & F \end{pmatrix}$$

where $F$ is the matrix

$$F = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ f_0 & f_1 & f_2 & \cdots & f_{k_1-1} \end{pmatrix}.$$

The matrix $D$ is somewhat more complicated because, when we apply the recurrence to calculate the value of $a_{i+1,j+k_2}$, for example, we get terms $a_{i+k_1,j+s}$. We need then to apply the first recurrence to obtain $a_{i+1,j+k_2}$ in terms of $a_{i+r,j+s}$ for $0 \le r < k_1$, $0 \le s < k_2$. We get

$$\begin{aligned} a_{i+1,j+k_2} &= \sum_{r=0}^{k_1-1} \sum_{s=0}^{k_2-1} g_{r,s} a_{i+r+1,j+s} \\ &= \sum_{r=0}^{k_1-2} \sum_{s=0}^{k_2-1} g_{r,s} a_{i+r+1,j+s} + \sum_{s=0}^{k_2-1} g_{k_1-1,s} a_{i+k_1,j+s} \\ &= \sum_{r=1}^{k_1-1} \sum_{s=0}^{k_2-1} g_{r-1,s} a_{i+r,j+s} + \sum_{s=0}^{k_2-1} \sum_{r=0}^{k_1-1} g_{k_1-1,s} f_r a_{i+r,j+s} \\ &= \sum_{s=0}^{k_2-1} g_{k_1-1,s} f_0 a_{i,j+s} + \sum_{r=1}^{k_1-1} \sum_{s=0}^{k_2-1} (g_{r-1,s} + g_{k_1-1,s} f_r) a_{i+r,j+s}. \end{aligned}$$

If we write $\underline{g}_i = (g_{0,i}, g_{1,i}, \dots, g_{k_1-1,i})$ for $i = 0, 1, \dots, k_2 - 1$, then

$$\underline{g}_i F = (f_0 g_{k_1-1,i}, g_{0,i} + f_1 g_{k_1-1,i}, \dots, g_{k_1-2,i} + f_{k_1-1} g_{k_1-1,i})$$

and so

$$a_{i+1,j+k_2} = (\underline{g}_0 F, \underline{g}_1 F, \ldots, \underline{g}_{k_2-1} F) \begin{pmatrix} a_{i,j} \\ a_{i+1,j} \\ \vdots \\ a_{i+k_1-1,j} \\ \vdots \\ a_{i,j+k_2-1} \\ a_{i+1,j+k_2-1} \\ \vdots \\ a_{i+k_1-1,j+k_2-1} \end{pmatrix}$$

and so

$$D = \begin{pmatrix} \underline{0} & & & & \\ \underline{0} & & I_{k_1(k_2-1)} & & \\ \vdots & & & & \\ \underline{g}_0 & \underline{g}_1 & \underline{g}_2 & \cdots & \underline{g}_{k_2-1} \\ \underline{g}_0 F & \underline{g}_1 F & \underline{g}_2 F & \cdots & \underline{g}_{k_2-1} F \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \underline{g}_0 F^{k_1-1} & \underline{g}_1 F^{k_1-1} & \underline{g}_2 F^{k_1-1} & \cdots & \underline{g}_{k_2-1} F^{k_1-1} \end{pmatrix}$$

where $I_N$ denotes the identity matrix of size $N \times N$.

## 4 The isomorphism

In this section, we shall describe an isomorphism between a set of pseudo-random sequences and $GF(2^n)$, which will be useful later on. In order to do this, we need the concept of rings of polynomials modulo a particular polynomial.

Suppose that $f$ is a monic polynomial of degree $M$ with binary coefficients, so that

$$f(x) = f_0 + f_1 x + \ldots + f_{M-1} x^{M-1} + x^M$$

for some $f_0, f_1, \ldots, f_{M-1} \in GF(2)$. Let $S$ be the set of polynomials in $x$ of degree at most $M-1$ with binary coefficients. Then we may make $S$

into a ring in the following way. The sum of two elements of $S$ is defined componentwise, so that if $g(x) = g_0 + g_1x + \ldots + g_{M-1}x^{M-1}$, and $h(x) = h_0 + h_1x + \ldots + h_{M-1}x^{M-1}$, then

$$g(x) + h(x) = (g_0 + h_0) + (g_1 + h_1)x + \ldots + (g_{M-1} + h_{M-1})x^{M-1}.$$

Multiplication is performed by multiplying the two polynomials together symbolically, and then substituting for higher powers of $x$ using $f(x) = 0$. This procedure yields an element of $S$, since all powers of $x$ higher than $M-1$ may be replaced. It is straightforward to show that, under these operations, $S$ is a ring (see, for example (Herstein 1964)). In fact, if $f$ is irreducible, then $S$ is a field. The field $GF(2^m)$ may be constructed in this way, where $f$ is a primitive polynomial of degree $m$.

Suppose that we have a pseudo-random sequence $\underline{a}$ of length $n = 2^m - 1$, with associated polynomial $h$. Let $H$ denote the set of PRSs generated by $h$ together with the all zero sequence. We may consider an element of $H$ as a polynomial in a variable $x$, simply by identifying the sequence $(b_0, b_1, \ldots, b_{n-1})$ with the polynomial $b_0 + b_1x + \ldots + b_{n-1}x^{n-1}$. Although the set $H$ does not contain all possible binary polynomials of degree at most $n-1$, we may still define addition and multiplication on elements of $H$ in the same way as above, taking $f(x) = x^n - 1$. It turns out that, under these operations, $H$ is a field, and, since $H$ has $2^n$ elements, we have the following theorem.

**Theorem 4.1 (MacWilliams and Sloane 1976)** $H$ is isomorphic to $GF(2^n)$.

This theorem was stated without proof in (MacWilliams and Sloane 1976). We shall need to use the construction in the proof, so we present the proof in its entirety. In order to prove the theorem, we need first a simple lemma.

**Lemma 4.2** Let $\beta = \alpha^u$ for some integer $u$. Then

$$\sum_{i=0}^{n-1} \beta^i = \begin{cases} 1 & \text{if } u \equiv 0 \pmod{n} \\ 0 & \text{otherwise} \end{cases}$$

**Proof**

If $u \equiv 0 \pmod{n}$, then $\beta = 1$, and so

$$\sum_{i=0}^{n-1} \beta^i = \sum_{i=0}^{n-1} 1 = 1,$$

since $n$ is odd. Otherwise, since $\beta^n = 1$ and $\beta \neq 1$, we have

$$\sum_{i=0}^{n-1} \beta^i = \frac{(1+\beta^n)}{(1+\beta)} = 0.$$

**Proof of Theorem 4.1**

We shall exhibit the isomorphism between these two fields. Note first that they have the same number of elements - the elements of $H$ are just the $n$ circular shifts of any non-zero element of $H$ together with the zero element. There are therefore $n+1 = 2^m$ elements in $H$, the same number as in $GF(2^m)$.

To avoid confusion with elements of $H$, we shall denote elements of $GF(2^m)$ by polynomials in the variable $\alpha$. We shall define the trace of an element of $GF(2^m)$ by

$$Tr(\beta) = \beta + \beta^2 + \ldots + \beta^{2^{m-1}}.$$

We define two functions $\phi : H \rightarrow GF(2^m)$ and $\psi : GF(2^m) \rightarrow H$ as follows:

if $b(x) \in H$, then $\phi(b)$ is $b(\alpha^{-1})$

if $\gamma \in GF(2^m)$, then $\psi(\gamma)$ is $b(x)$, where $b_i = Tr(\gamma\alpha^i)$.

We shall show that $\phi(\psi(\gamma)) = \gamma$ for all $\gamma \in GF(2^m)$. This will establish a one to one correspondence between $H$ and $GF(2^m)$. Now

$$\begin{aligned}
\phi(\psi(\gamma)) &= \sum_{i=0}^{n-1} Tr(\gamma\alpha^i)\alpha^{-i} \\
&= \sum_{i=0}^{n-1}\sum_{j=0}^{m-1} (\gamma\alpha^i)^{2^j}\alpha^{-i} \\
&= \sum_{i=0}^{n-1}\sum_{j=0}^{m-1} \gamma^{2^j}(\alpha^{2^j-1})^i \\
&= \sum_{j=0}^{m-1} \gamma^{2^j} \sum_{i=0}^{n-1} (\alpha^{2^j-1})^i
\end{aligned}$$

$$= \gamma$$

by Lemma 4.2.

We need only show that $\phi$ and $\psi$ are field homomorphisms and we will be done. Now clearly

$$\phi(b+c) = (b+c)(\alpha^{-1}) = b(\alpha^{-1}) + c(\alpha^{-1}) = \phi(b) + \phi(c)$$

and

$$\psi(\gamma+\beta) = \sum_{i=0}^{n-1} Tr((\gamma+\beta)\alpha^i)x^i = \sum_{i=0}^{n-1}(Tr(\gamma\alpha^i) + Tr(\beta\alpha^i))x^i = \psi(\gamma) + \psi(\beta).$$

Since $\alpha^{-1}$ satisfies $(\alpha^{-1})^n = 1$, we see that

$$\phi(b \times c) = (b \times c)(\alpha^{-1}) = b(\alpha^{-1}) \times c(\alpha^{-1}) = \phi(b) \times \phi(c).$$

Now

$$\psi(\gamma) \times \psi(\beta) = (\sum_{i=0}^{n-1} Tr(\gamma\alpha^i)x^i) \times (\sum_{j=0}^{n-1} Tr(\beta\alpha^j)x^j)$$

where all higher powers of $x$ are replaced using $x^n = 1$. So the coefficient of $x^k$ is

$$\begin{aligned}
d_k &= \sum_{i=0}^{n-1} \sum_{\substack{j=0 \\ i+j \equiv k \pmod n}}^{n-1} Tr(\gamma\alpha^i)Tr(\beta\alpha^j) \\
&= \sum_{i=0}^{k} Tr(\gamma\alpha^i)Tr(\beta\alpha^{k-i}) + \sum_{i=k+1}^{n-1} Tr(\gamma\alpha^i)Tr(\beta\alpha^{n+k-i}) \\
&= \sum_{i=0}^{k}\sum_{r=0}^{m-1}\sum_{s=0}^{m-1}(\gamma\alpha^i)^{2^r}(\beta\alpha^{k-i})^{2^s} + \sum_{i=k+1}^{n-1}\sum_{r=0}^{m-1}\sum_{s=0}^{m-1}(\gamma\alpha^i)^{2^r}(\beta\alpha^{n+k-i})^{2^s} \\
&= \sum_{r=0}^{m-1}\sum_{s=0}^{m-1}\gamma^{2^r}\beta^{2^s}(\sum_{i=0}^{k}(\alpha^i)^{2^r}(\alpha^{k-i})^{2^s} + \sum_{i=k+1}^{n-1}(\alpha^i)^{2^r}(\alpha^{n+k-i})^{2^s}) \\
&= \sum_{r=0}^{m-1}\sum_{s=0}^{m-1}\gamma^{2^r}\beta^{2^s}(\alpha^{k2^s}\sum_{i=0}^{k}(\alpha^i)^{2^r-2^s} + \alpha^{(n+k)2^s}\sum_{i=k+1}^{n-1}(\alpha^i)^{2^r-2^s})
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{r=0}^{m-1}\sum_{s=0}^{m-1}\gamma^{2^r}\beta^{2^s}\alpha^{k2^s}\sum_{i=0}^{n-1}(\alpha^i)^{2^r-2^s} \\
&= \sum_{r=0}^{m-1}\sum_{s=0}^{m-1}\gamma^{2^r}\beta^{2^s}\alpha^{k2^s}\sum_{i=0}^{n-1}(\alpha^{2^r-2^s})^i \\
&= \sum_{r=0}^{m-1}\gamma^{2^r}\beta^{2^r}\alpha^{k2^r} \\
&= Tr(\gamma\beta\alpha^k)
\end{aligned}
$$

as required. Hence both $\phi$ and $\psi$ are field homomorphisms, and we have established the isomorphism.

Note that, in $H$, shifting a sequence circularly to the right by $R$ is equivalent to multiplication by $x^{-R}$. This in turn, therefore, corresponds to multiplication by $\alpha^R$ in $GF(2^n)$, under the isomorphism described above. This fact allows us to deduce the characteristic polynomials of the linear recurrence matrices $C$ and $D$ introduced in the previous section.

**Theorem 4.3** Let $f$ be the minimal polynomial of $\alpha^T$; then the characteristic polynomial of $C$ is equal to $f^{k_2}$.

**Proof**
From the structure of $C$, we see that the characteristic polynomial of $C$ is equal to the $k_2$th power of the characteristic polynomial of $F$. Now $F$ is just the matrix which shifts the pseudo-random sequence which forms the array by $T$. As noted above, this means that multiplication by $F$ corresponds to multiplication by $\alpha^T$ in $GF(2^n)$. So $F$ certainly satisfies $f$, the minimal polynomial of $\alpha^T$. But the degree of the characteristic polynomial of $F$ is equal to $k_1$, which in turn is the degree of $f$. So the characteristic polynomial of $F$ is equal to $f$, and we have the desired result.

**Corollary 4.4** The eigenvalues of $C$ are $\alpha^T, \alpha^{2T}, \ldots, \alpha^{2^{k_1-1}T}$, each with multiplicity $k_2$.

**Theorem 4.5** The characteristic polynomial of $D$ is equal to the minimal polynomial of $\alpha^R$.

**Proof**

As in the proof of Theorem 4.1, we see that multiplication by $D$ corresponds to multiplication by $\alpha^R$. The degrees of the characteristic polynomial of $D$ and of the minimal polynomial of $\alpha^R$ are both equal to $k_1k_2$, and so the polynomials are equal.

**Corollary 4.6** The eigenvalues of $D$ are $\alpha^R, \alpha^{2R}, \ldots, \alpha^{2^{k_1k_2-1}R}$, which are all distinct and hence $D$ is diagonalizable.

## 5 Reducing to discrete logarithms

We recall that $\underline{s}_{r,t} = C^r D^t \underline{s}_{0,0}$, where $\underline{s}_{r,t}$ denotes the vector corresponding to the $k_1 \times k_2$ subarray with top left hand corner at position $(r,t)$. We shall use the same method as in (Paterson 1991) to convert this into a matrix equation which we may solve for $C^r D^t$. Now, for any $u$ with $0 \leq u \leq k_1k_2 - 1$, we have $\underline{s}_{r,t+u} = C^r D^t \underline{s}_{0,u}$ so we may construct matrices

$$S_{r,t} = (\underline{s}_{r,t}, \underline{s}_{r,t+1}, \cdots, \underline{s}_{r,t+k_1k_2-1})$$

and deduce that

$$S_{r,t} = C^r D^t S_{0,0}.$$

**Theorem 5.1** $S_{0,0}$ is invertible.

**Proof**

The columns of $S_{0,0}$ are $\underline{s}_{0,0}, \underline{s}_{0,1}, \ldots, \underline{s}_{0,k_1k_2-1}$. Since $\underline{s}_{0,u} = D^u \underline{s}_{0,0}$, we see that they are in fact equal to $\underline{s}_{0,0}, D\underline{s}_{0,0}, \ldots, D^{k_1k_2-1}\underline{s}_{0,0}$. If they were linearly dependent, then there would exist $d_0, d_1, \ldots, d_{k_1k_2-1} \in GF(2)$ not all zero such that

$$(d_0 + d_1 D + \ldots + d_{k_1k_2-1} D^{k_1k_2-1})\underline{s}_{0,0} = \underline{0}.$$

Apart from the all zero vector, any binary vector $\underline{v}$ of length $k_1k_2$ can be written as $C^r D^t \underline{s}_{0,0}$ for some integers $r$ and $t$, since every possible subarray appears in the array. Since $C$ and $D$ commute, this means that

$$(d_0+d_1D+\ldots+d_{k_1k_2-1}D^{k_1k_2-1})\underline{v} = C^r D^t (d_0+d_1D+\ldots+d_{k_1k_2-1}D^{k_1k_2-1})\underline{s}_{0,0} = \underline{0}$$

and so $(d_0 + d_1 D + \ldots + d_{k_1 k_2 - 1} D^{k_1 k_2 - 1})$ must be identically zero. But $D$ satisfies an irreducible polynomial of degree $k_1 k_2$, so this means that all the $d_i$ must be zero, and hence the columns of $S_{0,0}$ must be linearly independent. So $S_{0,0}$ is indeed invertible.

So we deduce that $C^r D^t = S_{r,t} S_{0,0}^{-1}$. Now it would be entirely possible at this stage to consider this as a discrete logarithm problem in the group of matrices generated by $C$ and $D$, but this would be computationally unwieldy, and so we use the same method as Paterson (Paterson 1991). In the one-dimensional case, the analogous equation is $C^r = S_r S_0^{-1}$, and the approach adopted is to diagonalize $C$ and look at the first eigenvalue. Now if $P$ diagonalizes $C$, that is $P^{-1}CP$ is diagonal, then certainly $P^{-1}C^r P$ is also diagonal and, further, if $\alpha$ is the first eigenvalue of $C$, then $\alpha^r$ is the first eigenvalue of $C^r$. So $P$ may be determined beforehand, and then used to diagonalize $C^r$. Looking at the first entry on the diagonal of the resulting matrix gives us $\alpha^r$, and we have thus reduced the problem of determining $r$ to that of discrete logarithms in $GF(2^m)$. In the two dimensional case, we need to diagonalize $C$ and $D$ simultaneously. This is possible because $C$ and $D$ commute, and is particularly easy because $D$ has distinct eigenvalues. We first need a lemma on commutativity of matrices.

**Lemma 5.2** Let $\Delta$ be a diagonal matrix with distinct entries on the diagonal. Then the only matrices which commute with $\Delta$ are the diagonal ones.

**Proof**

Let $A = (a_{i,j})$ be a matrix which commutes with $\Delta$, and let the diagonal elements of $\Delta$ be $\lambda_0, \ldots, \lambda_{m-1}$. Let $\delta_{i,j}$ denote the Kronecker delta, that is

$$\delta_{i,j} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}.$$

So $\Delta = (\lambda_i \delta_{i,j})$. Now the $(i,j)$th element of $A\Delta$ is $\sum_{k=0}^{m-1} a_{i,k} \lambda_k \delta_{k,j}$ and the $(i,j)$th element of $\Delta A$ is $\sum_{k=0}^{m-1} \lambda_i \delta_{i,k} a_{k,j}$, and so we have

$$\sum_{k=0}^{m-1} a_{i,k} \lambda_k \delta_{k,j} = \sum_{k=0}^{m-1} \lambda_i \delta_{i,k} a_{k,j}$$

for all $i, j$ with $0 \leq i, j \leq m-1$. But

$$\sum_{k=0}^{m-1} a_{i,k} \lambda_k \delta_{k,j} = \lambda_j a_{i,j} \quad \text{and} \quad \sum_{k=0}^{m-1} \lambda_i \delta_{i,k} a_{k,j} = \lambda_i a_{i,j}$$

so we have

$$a_{i,j}(\lambda_i - \lambda_j) = 0.$$

Since the $\lambda_i$ are distinct, this forces $a_{i,j} = 0$ if $i \neq j$. In other words, $A$ is diagonal.

We are now able to prove the result on simultaneous diagonalization.

**Theorem 5.3** Suppose that $A$ and $B$ commute, and that $B$ has distinct eigenvalues. Then there exists a matrix $P$ such that $P^{-1}BP$ is diagonal. Furthermore, $P^{-1}AP$ is also diagonal.

**Proof**
Since $B$ has distinct eigenvalues, there exists a matrix $P$ such that $P^{-1}BP$ is diagonal, and has the eigenvalues of $B$ down the diagonal. Now $A$ and $B$ commute, so $P^{-1}AP$ and $P^{-1}BP$ commute. But, by Lemma 5.2, the only matrices which commute with a diagonal matrix with distinct entries on the diagonal are the diagonal matrices. Hence $P^{-1}AP$ is diagonal.

**Corollary 5.4** There exists a matrix $P$ such that $P^{-1}CP$ and $P^{-1}DP$ are both diagonal.

**Proof**
By Lemma 3.1, $C$ and $D$ commute, and by Corollary 4.5, $D$ has distinct eigenvalues. Hence, by the theorem, we can simultaneously diagonalize $C$ and $D$.

**Corollary 5.5** There exists a matrix $P$, independent of $r$ and $t$, such that $P^{-1}C^rD^tP$ is diagonal.

**Proof**
We take any $P$ which diagonalizes $D$; then by the theorem, $P$ also diagonalizes $C$. Hence $P^{-1}C^rD^tP = (P^{-1}CP)^r(P^{-1}DP)^t$ is diagonal.

**Lemma 5.6** If $P$ is such that the first diagonal element of $P^{-1}CP$ is $\alpha^T$ and the first diagonal element of $P^{-1}DP$ is $\alpha^R$, then the first diagonal entry of $P^{-1}C^rD^tP$ is $\alpha^{rT+tR}$.

**Lemma 5.7** If $rT + tR \equiv k \pmod{n_1n_2}$, then $r \equiv k \pmod{n_1}$ and $t \equiv k \pmod{n_2}$.

**Proof**
This follows from the fact that $T \equiv 1 \pmod{n_1}$, $T \equiv 0 \pmod{n_2}$, $R \equiv 0 \pmod{n_1}$, $R \equiv 1 \pmod{n_2}$.

We have thus established the reduction of the original problem of finding the position of a subarray within a pseudo-random array to that of discrete logarithms in $GF(2^m)$.

**Theorem 5.8** Finding the position of a $k_1 \times k_2$ subarray in a $n_1 \times n_2$ pseudo-random array is equivalent to solving the discrete logarithm problem in $GF(2^{k_1 k_2})$.

The procedure for doing this may be summarised as follows. We suppose that we are given the linear recurrence matrices $C$ and $D$, an initial subarray written as a vector $\underline{s}_{0,0}$ and the subarray $\underline{s}_{r,t}$ whose position we wish to determine. Let $\alpha$ denote a root of $h(x) = 0$.

(1) Find a matrix $P$ such that $P^{-1}DP$ is diagonal.

(2) Construct the matrix $S_{0,0} = (\underline{s}_{0,0}, D\underline{s}_{0,0}, \ldots, D^{k_1 k_2 - 1}\underline{s}_{0,0})$ and calculate its inverse.

(3) Construct the matrix $S_{r,t} = (\underline{s}_{r,t}, D\underline{s}_{r,t}, \ldots, D^{k_1 k_2 - 1}\underline{s}_{r,t})$ and calculate $X = S_{r,t}S_{0,0}^{-1}$.

(4) Compute the first diagonal element, $x$ say, of the matrix $P^{-1}XP$.

(5) Find $k$ such that $x = \alpha^k$. Then $r \equiv k \pmod{n_1}$ and $t \equiv k \pmod{n_2}$.

## 6 Example

Let $n = 15$, $n_1 = 3$, $n_2 = 5$, $k_1 = k_2 = 2$. Then $R = 6$ and $T = 10$. We shall use the array from (MacWilliams and Sloane 1976), with $h(x) = x^4 + x + 1$. This array is

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

The matrices for moving in the horizontal and vertical directions respectively are as follows.

$$C = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

and

$$D = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

We first calculate $P$ such that $P^{-1}DP$ is diagonal. The eigenvalues of $D$ are $\alpha^6, \alpha^{12}, \alpha^{24} = \alpha^9, \alpha^{18} = \alpha^3$, and satisfy the equation $x^4 + x^3 + x^2 + x + 1 = 0$. If we denote $\alpha^6$ by $\beta$, then the matrix $P$ is equal to

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ \beta^2 + \beta^3 & \beta + \beta^4 & \beta + \beta^4 & \beta^2 + \beta^3 \\ \beta & \beta^2 & \beta^3 & \beta^4 \\ 1 + \beta + \beta^2 & \beta + \beta^3 & 1 + \beta + \beta^3 & \beta + \beta^2 \end{pmatrix}$$

and from this, we calculate $P^{-1}$ to be

$$\begin{pmatrix} \beta^4 & \beta + \beta^2 & 1 & \beta^2 + \beta^3 \\ \beta^3 & \beta^3 + \beta + 1 & 1 & \beta + \beta^4 \\ \beta^2 & \beta^3 + \beta & 1 & \beta^4 + \beta \\ \beta & \beta^+\beta + 1 & 1 & \beta^3 + \beta^2 \end{pmatrix}$$

We now calculate $P^{-1}CP$, and may check that it is indeed diagonal.

$$P^{-1}CP = \begin{pmatrix} \beta^2 + \beta^3 & 0 & 0 & 0 \\ 0 & \beta + \beta^4 & 0 & 0 \\ 0 & 0 & \beta^2 + \beta^3 & 0 \\ 0 & 0 & 0 & \beta + \beta^4 \end{pmatrix}$$

We may check that $\alpha^T = \alpha^{10}$, the first eigenvalue of $C$, is equal to $\beta^2 + \beta^3$.

Suppose that the initial state of the pseudo-random array is the vector (0010), and that the state whose position we are trying to find is (1001). Recall that, although these correspond to the subarrays $\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$ and $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

respectively, in the original pseudo-random array, in order to apply our method, we write them as vectors as described in Section 3.

We must now construct the matrices $S_{0,0}$ and $S_{r,t}$.

$$S_{0,0} = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

and

$$S_{r,t} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

So we have

$$X = S_{r,t}S_{0,0}^{-1} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

The first column of $XP$ is therefore

$$\begin{pmatrix} \beta+1 \\ \beta^3+\beta+1 \\ \beta^2+\beta \\ \beta^3+1 \end{pmatrix}$$

which means that the first element $x$ of $P^{-1}XP$ is equal to

$$\beta^4(\beta+1)+(\beta+\beta^2)(\beta^3+\beta+1)+1.(\beta^2+\beta)+(\beta^2+\beta^3)(\beta^3+1) = 1+\beta = 1+\alpha^2+\alpha^3.$$

We must therefore find $k$ such that $\alpha^k = 1 + \alpha^2 + \alpha^3$. At this stage, any discrete logarithm algorithm may be used. For purposes of illustration, we shall use the Pohlig-Silver-Hellman method. Of course, in this case the field is so small that simply listing all the powers of $\alpha$ and looking among them for $1 + \alpha^2 + \alpha^3$ would be the way to do it, but obviously, in any real system, the fields would be larger, and this step would form the bulk of the computation. The Silver-Pohlig-Hellman algorithm would be suitable for fields of reasonable size with the property that $2^m - 1$ splits into the product

of small primes. A full description of the algorithm, together with other algorithms for the discrete logarithm problem, may be found in the excellent survey paper by Odlyzko (Odlyzko 1985). The idea is that if $2^m - 1 = \prod p_i^{n_i}$, then $k$ may be determined modulo each $p_i^{n_i}$ in turn, and the results combined by the Chinese Remainder Theorem. In this case, the size of the field is $15 = 3 \times 5$, so we determine $k$ modulo 3 and 5 respectively. The method consists of a precomputation phase, where the $p_i$th roots of unity are calculated and stored. In our example, we must compute the cube roots of unity and the fifth roots of unity. The cube roots of unity are 1, $\alpha^5 = \alpha + \alpha^2$, and $\alpha^{10} = 1 + \alpha + \alpha^2$. The fifth roots of unity are 1, $\alpha^3$, $\alpha^6 = \alpha^2 + \alpha^3$, $\alpha^9 = \alpha + \alpha^3$, and $\alpha^{12} = 1 + \alpha + \alpha^2 + \alpha^3$.

In the next phase of the algorithm, for each prime $p_i$, we raise $x$ to the power $(2^m - 1)/p_i$ to obtain a $p_i$th root of unity, which we may then look up in the precomputed table. In this case, we first compute $y = x^3$ to obtain $k \pmod 5$. Now $x^3 = (1 + \alpha^2 + \alpha^3)^3 = \alpha + \alpha^3$, so we deduce that $k \equiv 3 \pmod 5$. Similarly, we compute $z = x^5 = (1 + \alpha^2 + \alpha^3)^5 = \alpha + \alpha^2$, and so $k \equiv 1 \pmod 3$. We deduce that $r = 3$ and $t = 1$, so we have found the position of the vector (1001) to be $(3, 1)$, which may be checked by reference to the array.

## 7 References


Basran, J. S., Petriu, E. M. and Groen, F. C. A. (1989). Developments in the measuring of the absolute position of automated guided vehicles. *Proceedings of IEEE Instrumentation and Measurement Technology Conference*, IEEE, New York, pp. 132-137

Burns, J. and Mitchell, C. J. (1991). Coding schemes for two-dimensional position sensing. *to appear in Proceedings of this conference*

Green, D. H. (1985). Structural properties of pseudo-random arrays and volumes and their related sequences. *IEE Proceedings, Vol.132, Pt. E, No. 3*, pp. 133-145

Herstein, I. N. (1964). Topics in Algebra. Blaisdell, New York.

MacWilliams, F. J. and Sloane, N. J. A. (1976). Pseudo-Random Sequences and Arrays. *Proceedings of the IEEE 64 No. 12*, pp. 1715-1728

Odlyzko, A. M. (1985). Discrete logarithms in finite fields and their cryptographic significance. *Advances in Cryptology - Proceedings EURO-*

*CRYPT 84*, Springer Verlag, Heidelberg, pp. 224-314.

Paterson, K. G. (1991). Finding your position in a binary M-sequence - a reduction to extracting discrete logarithms in $GF(2^k)$. *Internal note, Hewlett-Packard Labs., Bristol*

Shi, Wen-Hong and Chen, Jin-Guang (1988). Study on the recurrences of pseudo-random array. *Electronic Letters, 24 No. 8*, pp. 499-500