First Published 2013 by
Faculty of Science,
Universiti Brunei Darussalam
Jalan Tungku Link
Bandar Seri Begawan BE1410
Brunei Darussalam

This book consists of papers prepared by staff of Universiti Brunei Darussalam or the Brunei Ministry of Primary Resources, and refereed locally.

Cataloguing in Publication Data

Scientia Bruneiana / Chief Editor Prof. Hj Mohamed Abdul Majid

23 p.; 30 cm

ISSN : 1819-9550

1. Research – Brunei Darussalam. 2. Science – Brunei Darussalam

Q180.B7 B788 2013

Cover photo: Fractal motifs created using the software Mutiara. (Courtesy: Nor Jaidi Tuah).

Printed in Brunei Darussalam by
Educational Technology Centre,
Universiti Brunei Darussalam

mk

SCIENTIA BRUNEIANA

A journal of science and science-related matters published each year by the Faculty of Science, University Brunei Darussalam. Contributions are welcomed in any area of science, mathematics, medicine or technology. Authors are invited to submit manuscripts to the editor or any other member of the Editorial Board. Further information including instructions for authors can be found on pages 22 and 23.

EDITORIAL BOARD

SCIENTIA BRUNEIANA is published by the Faculty of Science, Universiti Brunei Darussalam, Brunei Darussalam BE 1410

ISSN : 1819-9550

1. Research – Brunei Darussalam. 2. Science – Brunei Darussalam

Q180.B7 B788 2013

SCIENTIA BRUNEIANA

Publication Ethics Policy

The Editorial Board of *Scientia Bruneiana* is committed to implementing and maintaining the publication standards of a high-quality peer-reviewed scientific journal.

Each manuscript submitted to *Scientia Bruneiana* is examined by a referee with recognised expertise in the manuscript's subject area, and all communications between the referee and the author(s) pass must first through the Editorial Board, so that the identity of the referee remains confidential.

No one will be appointed as the referee of a manuscript if he or she is known to have a potentially compromising relationship with one or more of the authors of the manuscript, as for example in being related through blood or marriage to an author, or in being the research supervisor or research student of an author.

The Editorial Board of *Scientia Bruneiana* makes every effort to ensure that each paper published in the journal is free of plagiarism, redundant or recycled text, and fabricated or misrepresented data. Where possible, plagiarism detection software will be used to check for plagiarised or recycled text.

Provided that a manuscript is free of the ethical lapses described in the previous paragraph, the decision to publish it in *Scientia Bruneiana* is based entirely on its scientific or academic merit, as judged by the referee. The referee's assessment of the merit of the manuscript is final. While a full statement of the reasons behind the referee's decision will be passed on to the author(s), no appeals from the author(s) will be entertained.

Under no circumstances will the referee of a paper published in *Scientia Bruneiana* be credited as one of the authors of the paper, and other papers that have been authored or co-authored by the referee will be admitted to the paper's list of references only after an independent third party with expertise in the area has been consulted to ensure that the citation is of central relevance to the paper.

If a member of the Editorial Board of *Scientia Bruneiana* is listed as an author of a manuscript submitted to *Scientia Bruneiana*, that Board member will play no part whatsoever in the processing of the manuscript.

Where necessary, any corrections or retractions of papers previously published in *Scientia Bruneiana* will be printed in the earliest possible edition of the journal, once the need for a correction or retraction has been drawn to the attention of the Editorial Board.

SCIENTIA BRUNEIANA

2013

Research Articles

MUTIARA: AN INTERACTIVE DESIGNER FOR INTRICATE MOTIFS

Nor Jaidi Tuah[1], Seyed Mohamed Buhari[2], Abd Ghani Naim[3], Kim Onn Chong[4] and Sei Guan Lim[5]

Computer Science, Faculty of Science, Universiti Brunei Darussalam,
Tungku Link, Gadong BE 1410, Brunei Darussalam

[1]Email: norjaidi.tuah@ubd.edu.bn
[2]Email: mibuhari@gmail.com
[3]Email: ghani.naim@ubd.edu.bn
[4]Email: kimonn.chong@ubd.edu.bn
[5]Email: seiguan.lim@ubd.edu.bn

Abstract: This paper describes the features of a specialized drawing application, called Mutiara, that allows the user to design and apply motifs, the kind that repeat along a line so as to form a decorated border around an invitation card. Motifs are generated using iterated function systems derived from seeds that are interactively edited using intuitive click and drag actions. The seeds have user-configurable properties that affect their participation in the motif generation process. In particular, the seeds define the mapping functions used, and determine when they are used and how their outputs are rendered. Mutiara uses a class of non-linear mapping functions that we believe is unique, making it particularly appealing to fractal enthusiasts.

Keywords: motif design, iterated function system, fractal, interactive

1. Introduction

Border motifs are patterns, typically flowery or abstract, that repeat to form a line, for example to decorate an invitation card or a ceremonial plaque. In Brunei, contrary to the modern trend of keeping things plain and simple, the local folks cling tenaciously to the tradition of using flowery border motifs. Such motifs adorn doors, roof skirting, fences, ceilings, institutional logos, picture frames, cover pages of journals, and even school sports uniforms, sometimes in stark contrast to the plain surroundings. We are not claiming that the passion for intricate motifs is unique to Brunei, as a cursory glance at the artwork from various other countries indicates.

Our work was initially motivated by the traditional wavy motifs of Brunei called *air muleh*. We thought it would be an interesting cultural enrichment to use fractals. Our tinkering with fractal *air muleh* has resulted in Mutiara, an application that allows the user to design motifs (not necessarily of the *air muleh* variety) based on iterated function systems (IFS). Mutiara is designed to be easy to use so that even a casual user can be productive in no time. Mutiara also offers a new avenue for fractal enthusiasts in the pursuit of their hobby.

In Section 2, we review the few previously published works that have studied fractals, particularly those generated by IFS, for their beauty rather than other aspects such as realism. Mutiara already has a built-in gallery of fractal motifs that can be used straight away to decorate lines in the same way that we use the usual dotted and dashed styles in typical drawing applications. This is briefly explained in Section 3. Section 4 describes the motif design process, particularly from the user's perspective. The various options available to the user are used to define the IFS. The IFS uses non-linear mapping functions, as elaborated in

Section 5. Section 6 showcases several motifs selected from Mutiara's built-in gallery to illustrate the rich variety of patterns that it is capable of producing.

2. IFS Patterns as Art

Patterns produced using IFS are rich in visual variety and provide an inexhaustible source of aesthetically appealing examples. Sprott (Sprott, 1994) wrote a computer program to randomly generate such patterns. Despite working with only linear mapping functions, he obtained many pleasing patterns.

The beauty of IFS fractals can be considerably improved by assigning a color to each mapping (Draves and Reckase, 2008; Wijk and Saupe, 2004). Each time a mapping is applied, its color is mixed in to the accumulated result. Mutiara also uses colors. In addition, it allows the user to alter the mixing strength of each color.

The Fractal Flame algorithm (Draves and Reckase, 2008) greatly expands the variety of IFS fractals by using non-linear mappings, and enhances their appearance by employing several rendering techniques such as gamma adjustment. Non-linear mappings can be defined by using functions that operate on the complex plane, such as taking the square root of a complex number (Loocke, 2009). Mutiara also uses non-linear mappings, but in a completely different way.

3. Fancy Line Drawings

Our motif designer, Mutiara, can be regarded as a drawing application, albeit a very specialized one. In a typical drawing application, the user creates and edits lines using intuitive click and drag actions, and furnishes these lines with properties such as color, thickness and style (e.g. solid, dotted or dashed). Mutiara also uses the familiar click and drag actions to create and edit lines. However, unlike a typical drawing application, the lines are furnished with motifs. Motifs may either be chosen from a built-in gallery or designed by the user. Figure 1 shows some examples of lines embellished with motifs chosen from the gallery.

For each line, the following additional properties are also configurable:

- its thickness or width,
- how many copies of the motif to render along its length,
- how much detail to render,
- how alternate copies are flipped.

Figure 2 shows several lines embellished with the same motif but with different values for the properties mentioned above.

Note that even though Mutiara allows the user to configure how much detail to render, it will automatically stop when it deems that further detail is not discernible.

4. Line drawings as motif seeds

Mutiara features a gallery of motifs. The user can edit these motifs or create totally new ones. Each motif is based on an IFS, possibly non-linear, using line drawings as its seed. The user employs the usual click and drag actions to create and edit these lines.

Figure 1 Examples of lines styled with motifs

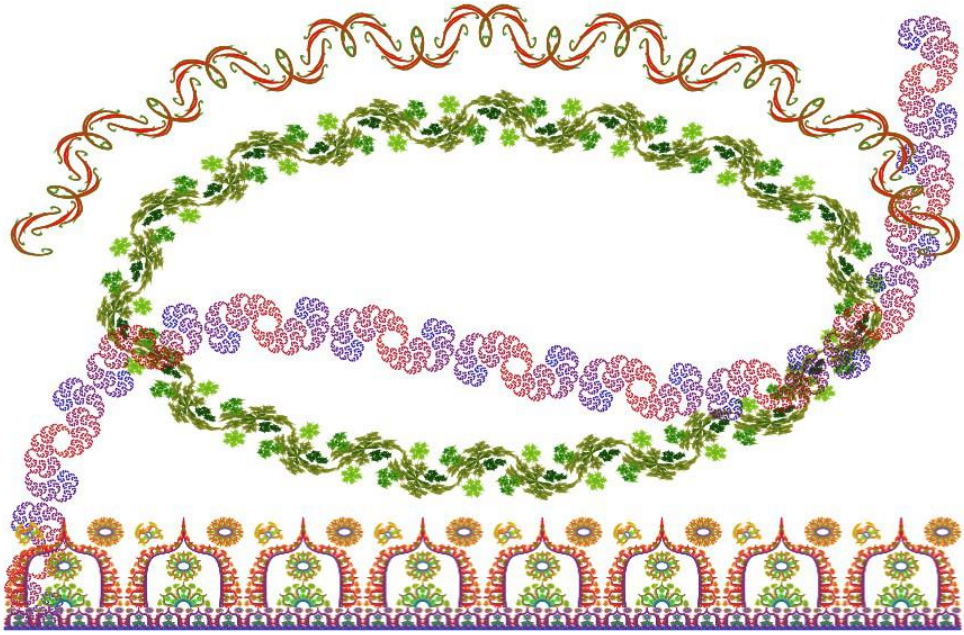
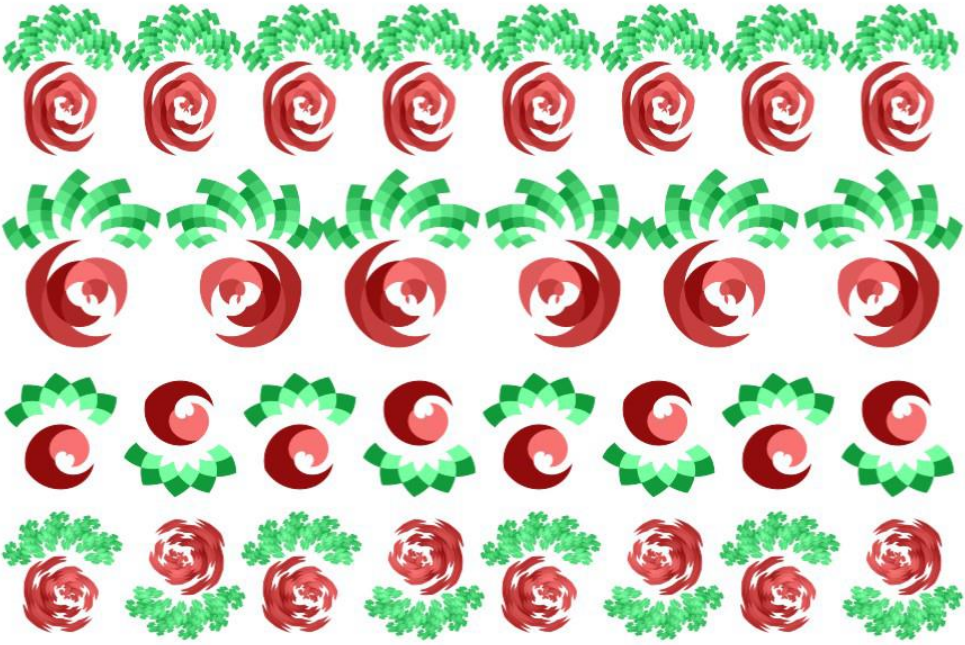


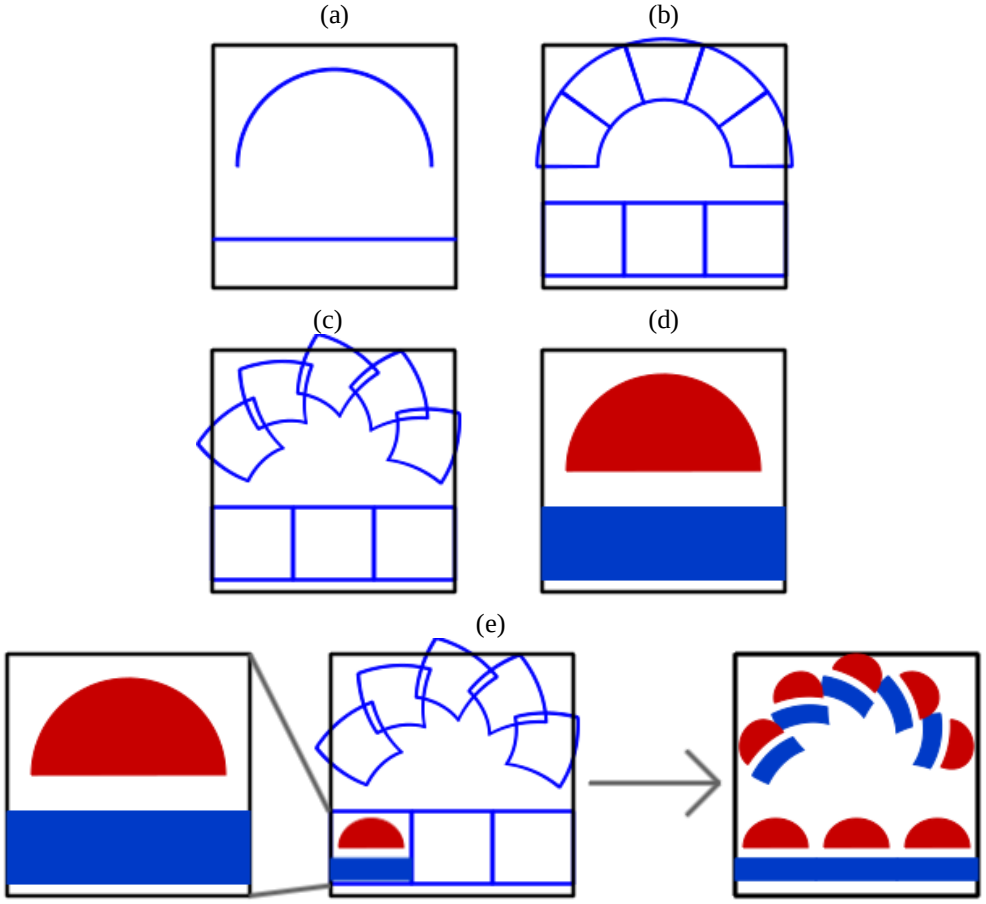
Figure 2 The same motif can appear in different sizes, different levels of detail, and flipped. In its default setting, the motif in this figure is actually a fractal. However, as can be seen here, its pre-fractal forms are also attractive.

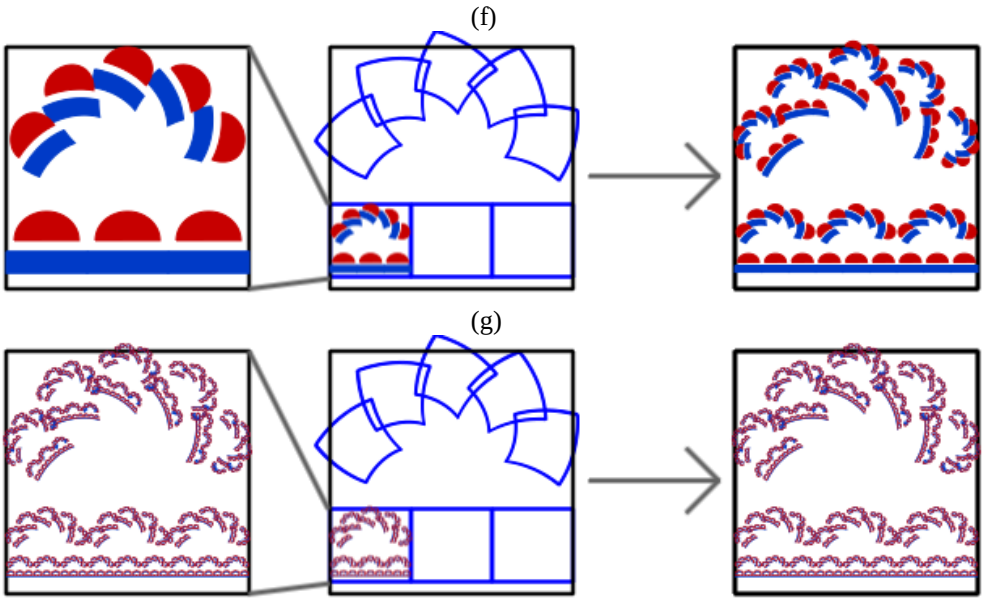


Intricate motifs are obtained by fractalizing the seed lines. To illustrate this, we will refer to Figure 3. Seed lines (Figure 3a) are thickened and segmented (Figure 3b). The segments can be geometrically tweaked (Figure 3c). Together, the tweaked segments define an IFS. At the same time, the seeds are given rendering styles. The rendering styles define the appearance of the motif at level 0 (Figure 3d). When a level-0 motif is replicated onto the segments, we obtain a level-1 motif with finer detail (Figure 3e). Similarly, a level-2 motif is obtained by replicating a level-1 motif (Figure 3f). Each additional iteration through the system produces a motif of finer detail than before. However, we will eventually hit the physical limit of our output device where further detail cannot be resolved (Figure 3g).

The segmentation, the replication and the rendering style of each seed are user-configurable through the properties described in the following subsections.

Figure 3 Intricate motifs are grown from relatively simple seeds. Seed lines are thickened (a) and segmented (b), tweaked (c) and styled (d). They are replicated (e), and replicated (f), ..., until further details cannot be resolved (g).





4.1 Properties for segmentation

The following properties are used to control the segmentation of a seed and, together with the formula for the seed itself, define the mapping functions for the IFS.

Baseline: This indicates the "vertical" position of a seed line relative to its segments prior to tweaking. The top row of Figure 4 shows the effect of changing this property.

Split: A seed is split into a number of segments of equal length. This property is used to set this number. The middle row of Figure 4 shows the effect of changing the split property.

Width: The width or thickness of a seed may vary along its length, as illustrated in the last row of Figure 4.

4.2 Properties for replication

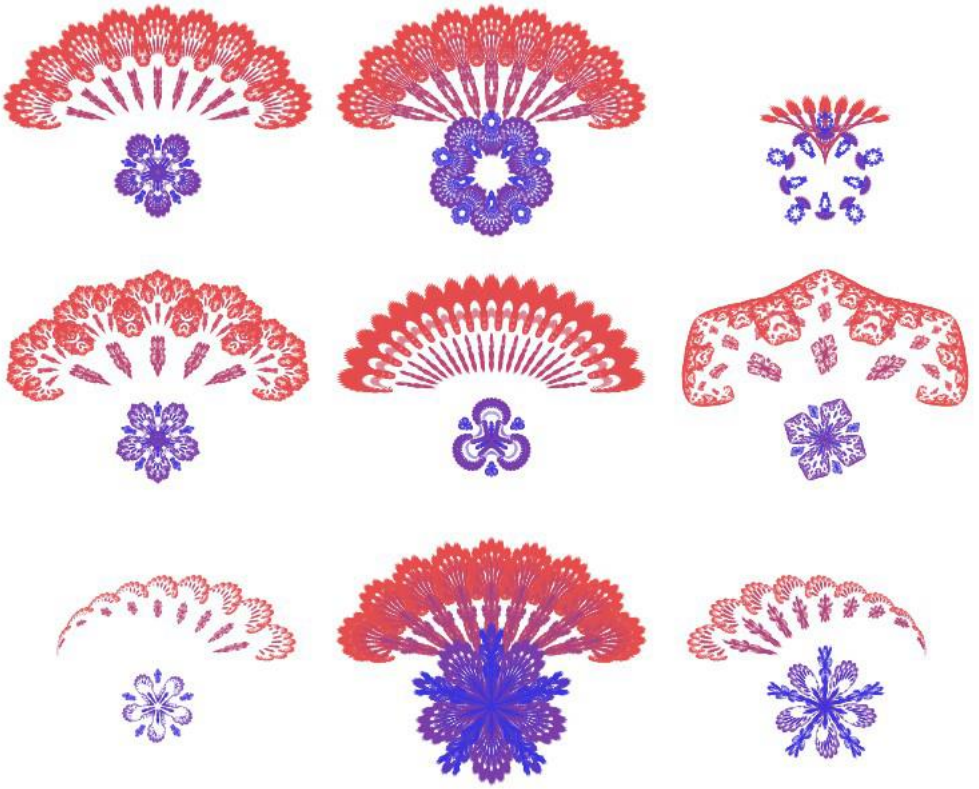
The following properties control the replication process. They are the meta-controls for the system. They determine which mapping functions to apply and when.

Role: A seed may be designated as *base*, *motif* or both. Seeds designated as base are operative only in the first iteration of the system. In effect, their segments are used as receptacles for replications. Seeds designated as motif are operative from the second iteration onward including the final rendering. In effect, they get replicated. The top row of Figure 5 illustrates this property.

Tweak: The segments of a seed can be rotated, scaled, flipped and nudged. Alternate segments can be flipped differently. The tweak property is applied prior to the baseline, the split and the width properties. So it is more convenient to regard it as tweaking the replicas rather than the segments. The middle row of Figure 5 shows motifs with different tweaks.

Detail: This property indicates the number of iterations that a seed is operative from the moment it is triggered. The effect of this property is shown in the last row of Figure 5.

Figure 4 The effect of changing the segmentation properties can be seen here. Every motif in this figure is generated using the same two seeds: a small circle and a semi-circular arc. The motifs in each row are configured the same way except for their values in one particular property. In the top row, the motifs differ in the baselines of their seeds. In the middle row, the differences are in their splits. In the last row, the differences are in their widths.



4.3 Properties for rendering

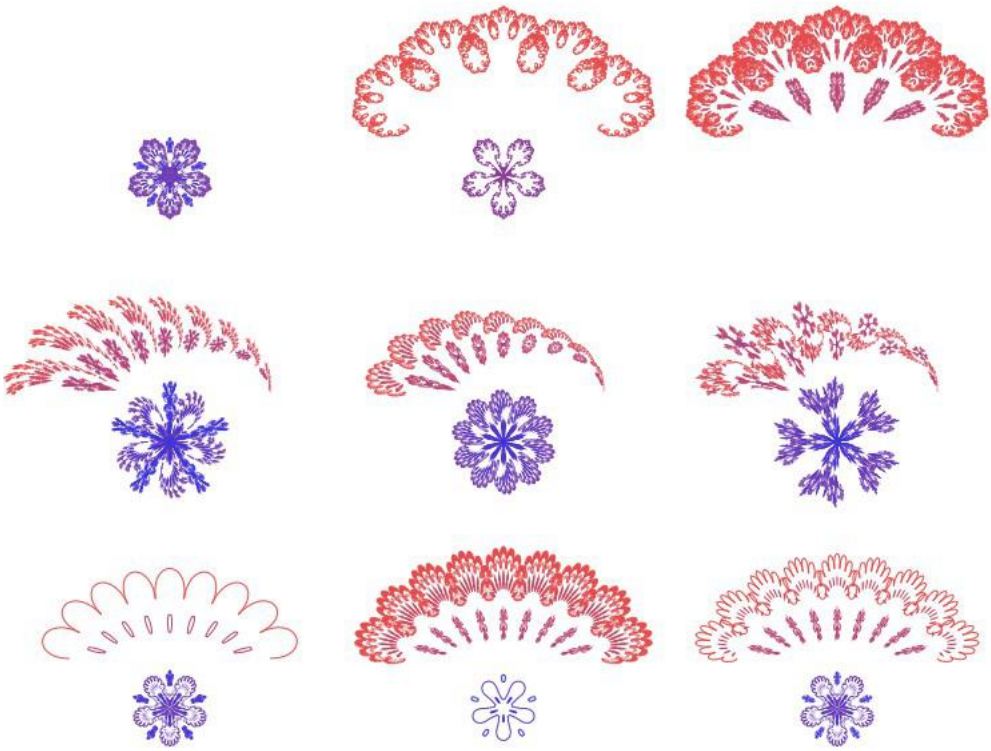
The following properties control how the seeds are rendered.

Color: Each seed contributes a color. This color is mixed with colors contributed by other seeds in earlier iterations. The top row of Figure 6 shows the same motifs but with different colors.

Mix: This property determines the "strength" of the color contributed by the seed. A value of 0 means its color will not be added at all, except in the very first iteration as an initial color. A value of 1 means it will totally override the currently effective color set by earlier iterations. A value in between means its color will be mixed with the current color in due proportion. The middle row of Figure 6 shows the effect of adjusting this property.

Style: Each seed is given a style such as hollow or thick or even none. This determines the set of input points to the system. The effect of this property is apparent when the level of detail is low. When the level of detail is high the effect is normally lost, as any set of starting points converges to the same set of fixed points. The last row of Figure 6 shows the same motif with different styles.

Figure 5 The effect of changing the replication properties is illustrated here. Every motif in this figure uses the same seeds as those in Figure 4. Each row depicts motifs having the same configuration except for one particular property. In the top row, the distinguishing property is the role; in the middle row, the tweak; in the last row, the detail.



4.4 Property for tweaking the entire motif

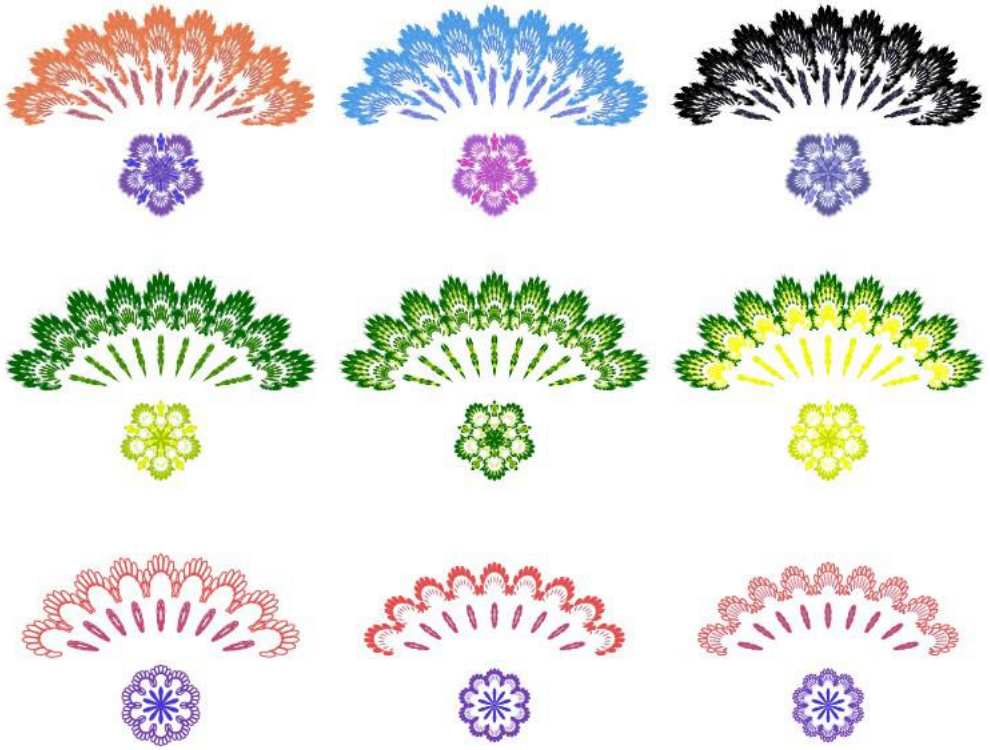
Finally, there is a property to tweak the entire segment. It is applied outside the IFS iterations and is similar to the final transform in Fractal Flame (Draves and Reckase, 2008). In Fractal Flame, the final transform acts like a camera to artistically distort the image. In Mutiara, the final tweaking also acts like a camera, but its primary purpose is to ensure that the final image is suitably oriented and sized for its role as a motif tile.

5. Mapping functions represented by seed segments

As explained in the previous section, seeds are segmented. Each segment defines a function that maps a point onto another. The mapping can be non-linear because the segment follows the curvature of the seed, which may indeed be curved as illustrated in Figure 3. In this section, we explain the mapping function.

A seed is defined by a parametric function s such that $s(0)$ and $s(1)$ are the two end-points of the seed and $s(t)$ is a point whose distance along the seed from $s(0)$ is proportional to t. For example, $s(0.5)$ is the midpoint of the seed.

Figure 6 The effect of changing the rendering properties is shown. Every motif in this figure uses the same seeds as those in Figure 4. Each row depicts motifs having the same configuration except for one particular property. In the top row, the distinguishing property is the color; in the middle row, the mix; in the last row, the style.



Suppose seed s is split into n segments of equal length. Let s_i, $1 \leq i \leq n$, be the i-th segment of s. We define s_i parametrically so that $s_i(0)$ and $s_i(1)$ are its end-points. Segment s_i starts at $s((i-1)/n)$ and ends at $s(i/n)$. Hence, $s_i(t) = s((i-1+t)/n)$.

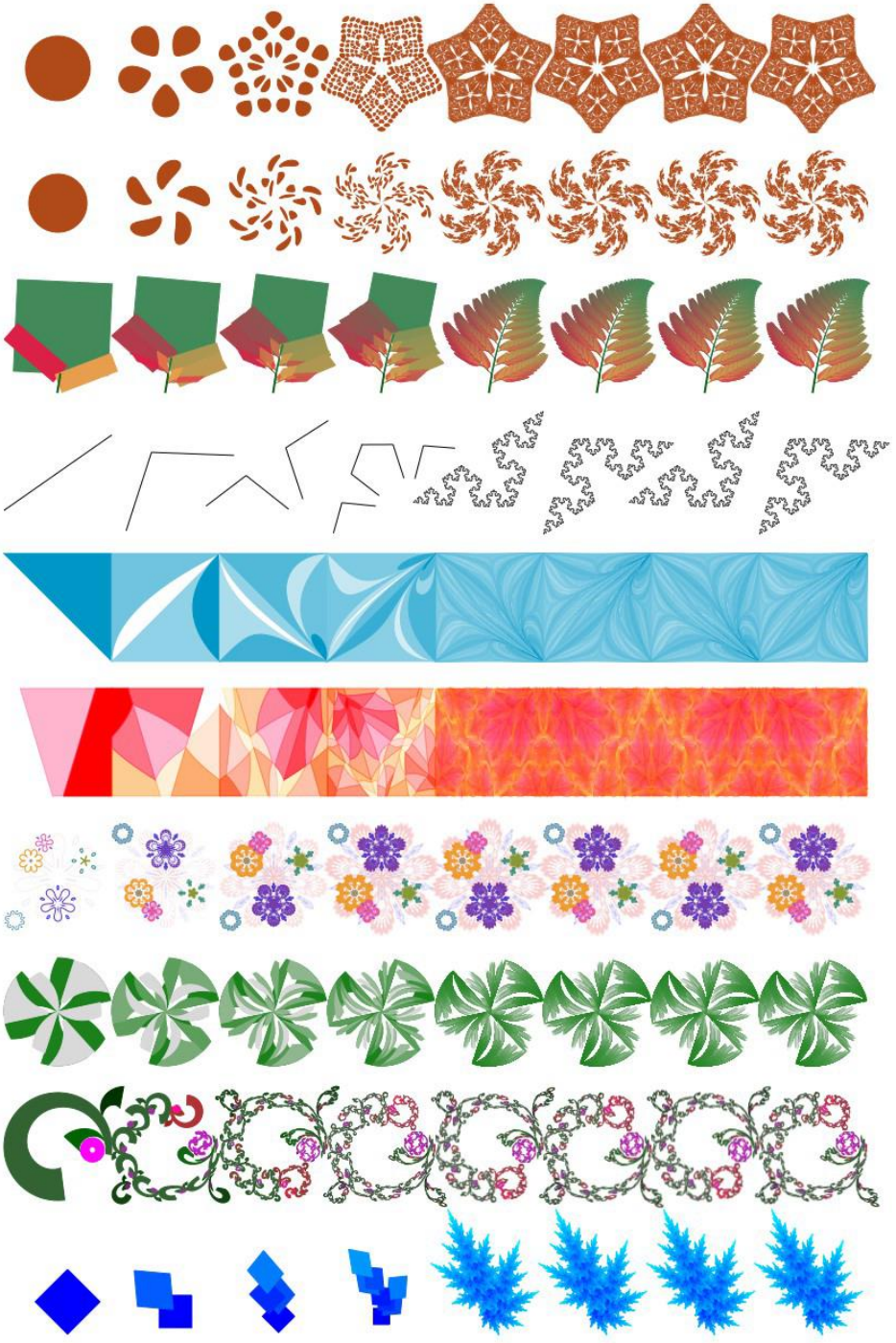
Let f^*_i be the mapping function of segment s_i assuming that it is not tweaked, i.e. as in Figure 3b rather than Figure 3c. The baseline property was explained above, in Section 4.1. The line (x, y_b), where y_b is a constant determined by the baseline property, is mapped linearly along s_i, i.e. $f^*_i(x, y_b) = s_i(x)$.

Now we deal with points that do not lie on the baseline. An arbitrary point (x, y) is mapped onto the normal at $s_i(x)$ in such a way that the distance from $f^*_i(x, y)$ to $s_i(x)$ is in proportion to $y - y_b$. The width property mentioned in Section 4.1 is used to calculate the span of the normal across the segment.

So far we have assumed we are working with a non-tweaked segment. Tweaking does not really present much of a problem as the tweak property is restricted to textbook linear transformations, namely translation, scaling, rotation and reflection; so we will omit the detail and simply use g to refer to the aggregated tweak function. The mapping function f_i represented by segment s_i is simply $f_i = f^*_i\, g$.

To summarize, the i-th segment of seed s represents the following function:

Figure 7 Mutiara patterns are rich in variety. Each row in this figure uses a motif selected from Mutiara's built-in gallery. The first four tiles in each row are progressively detailed pre-fractal versions.



$$f_i(x, y) = s(t) + (d \cos \theta, d \sin \theta)$$

where

$t = (i - 1 + x_g)/n$,
n = the number of segments in s,
$d = (y_g - y_b)w$,
y_b = a constant determined by the baseline property,
w = the width of the seed at $s(t)$,
θ = the slope of the normal at $s(t)$,
$(x_g, y_g) = g(x, y)$ where g is the tweak operation.

To the best of our knowledge, this mapping function is unique to Mutiara. What gives it a far greater variety than simple linear mappings is the *s* part. Currently, Mutiara allows straight lines, elliptical arcs and Bezier curves for *s*.

6. Sample motifs

This section is more graphical than textual. Figure 7 complements the previous figures to show the variety of patterns that can be produced with Mutiara.

7. Concluding remarks

Mutiara is an accidental application. We initially experimented to find out how a certain type of motifs traditionally used in Brunei, called *air muleh*, would look like if they were designed recursively. Although the Brunei folks use all sorts of motifs, they are particularly fond of *air muleh* and we thought that it would be an interesting cultural contribution to enrich this genre with fractals. The first *ad hoc* programming in this direction produced encouraging results and motivated us to develop an *air muleh* designer to help us produce more examples. We eventually decided to make it a general motif designer, not just for *air muleh*. Despite having an originally narrow focus, we believe that Mutiara has a much wider appeal and nicely fills a niche.

Beside its intended purpose as a motif designer, Mutiara is also a welcome addition for fractal enthusiasts. Its gallery features a few common IFS fractals such as the iconic fern and the dragon curve. Fans can revisit these classic fractals and experiment with them in a new way. They can also discover new kinds of fractals using Mutiara's unique class of mapping functions.

References

Draves, S. and Reckase, E. 2008. The fractal flame algorithm. http://flam3.com/flame_draves.pdf [Retrieved 14 Dec 2009]

Loocke, P.V. 2009. Non-linear iterated function systems and the creation of fractal patterns over regular polygons. *Computer & Graphics* 33, 698–704.

Sprott, J. C. 1994. Automatic generation of iterated function systems. *Computer & Graphics* 18(3), 417–425.

Wijk, J. and Saupe, D. 2004. Image based rendering of iterated function systems. *Computers & Graphics* 28, 937–943.

MULTIPLE LINEAR APPROXIMATIONS IN DIFFERENTIAL-LINEAR CRYPTANALYSIS OF 8-ROUND DES

Abd Ghani Naim

Computer Science, Faculty of Science, Universiti Brunei Darussalam,
Tungku Link, Gadong BE 1410, Brunei Darussalam

Email: ghani.naim@ubd.edu.bn

Abstract: The Data Encryption Standard (DES) has been extensively studied since its inception in the 1970s, even though it was replaced by the newer Advanced Encryption Standard (AES) in the 1990s. DES continued to be studied well after its replacement, as it was one of the first encryption algorithms to be used as a standard for securing communications in business transactions. It was replaced because its key size was deemed too short for secure communications given current advances in computer technology. In this article, we concentrate on cryptanalysis of 8-round DES (i.e. a reduced variant of the full 16-round DES) which has also been extensively studied, since it is presumed to be much weaker than the full 16-round DES. For 8-round DES, one of the best cryptanalysis techniques used is Differential-Linear Cryptanalysis, which is a Known Plaintext Attack that uses a single linear approximation and specific differential characteristic to recover its subkeys. In this article, we implement multiple linear approximations and specific differential characteristics to recover 16-bit subkeys. Our implementation makes some improvements over Differential-Linear Cryptanalysis of 8-round DES in terms of the number of known plaintexts required to recover these subkeys.

1. Introduction

Multiple Linear Approximations in Linear Cryptanalysis was introduced by Burton S. Kaliski Jr. and M.J.B. Robshaw (Kaliski and Robshaw, 1994) as an enhancement of Linear Cryptanalysis, which was developed by Matsui (Matsui, 1993). In Linear Cryptanalysis we can use a single linear approximation to recover certain key bits of ciphers such as DES, for instance. In addition to this, linear cryptanalysis also recovers parity bits of the key. Normally, the best linear approximation is used in linear cryptanalysis. Kaliski and Robshaw (Kaliski and Robshaw, 1995) first introduced the idea of using more than one linear approximation in linear cryptanalysis in 1994. The advantage of using multiple linear approximations rather than a single one is the increase in the success rate without having to increase the number of known plaintexts.

In Differential-Linear cryptanalysis of 8-round DES, there are two parts to the cryptanalysis, i.e. the differential cryptanalysis part and the linear cryptanalysis part. In this paper, we are concentrating on improving the linear cryptanalysis part. Details of differential-linear cryptanalysis of 8-round DES can be found in (Langford and Hellman, 1994).

Differential-Linear cryptanalysis of 8-round DES uses Matsui's optimal 3-round approximation in the linear part of the cryptanalysis. In this paper, we investigate the feasibility of using multiple 3-round linear approximations to decrease the number of required plaintexts.

Let us now discuss the different linear approximations that we can use in improving differential-linear cryptanalysis.

2. Multiple Linear Approximations

First, note that the i^{th} round DES subkey K_i is divided into 6-bit blocks so that $K_{i,j}$ refers to the j^{th} block of the subkey K_i (reading from left to right). S_i refers to the i^{th} S-box of DES. Also, K_i[j] refers to the j^{th} bit of the subkey K_i where the rightmost bit of K_i is K_i[0].

The linear cryptanalysis part of the differential-linear cryptanalysis relies on bits that are unchanged after going through the differential part of the cryptanalysis. The linear cryptanalysis part begins from round 5 to round 7. Figure 1 shows the setup used in differential-linear cryptanalysis of 8-round DES. After going through the 3-round differential characteristic, we are left with several bits that are unchanged. These bits are the ones that are used in the linear cryptanalysis part of differential-linear cryptanalysis. Figure 2 shows how a differential characteristic is used to produce unchanged bits to be used in linear cryptanalysis.

Figure 1 Differential-linear attack on 8-round DES

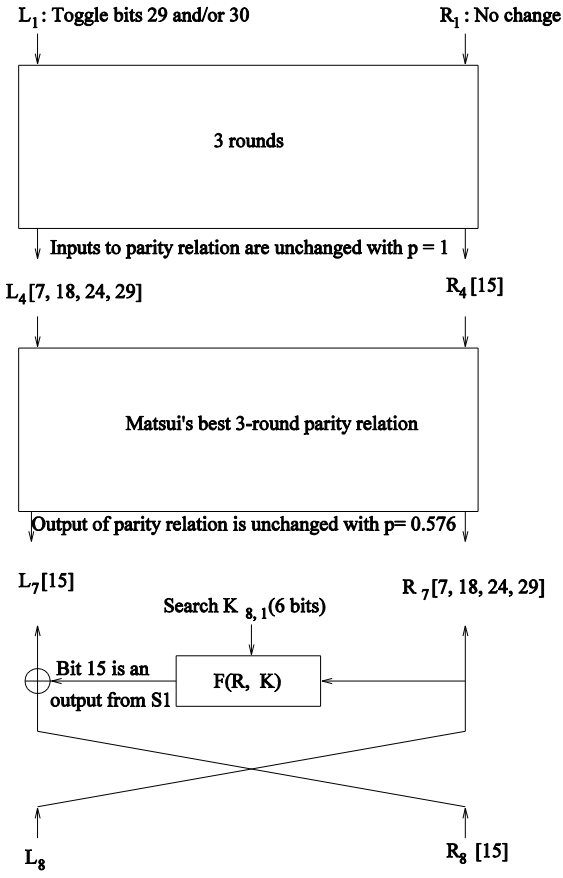
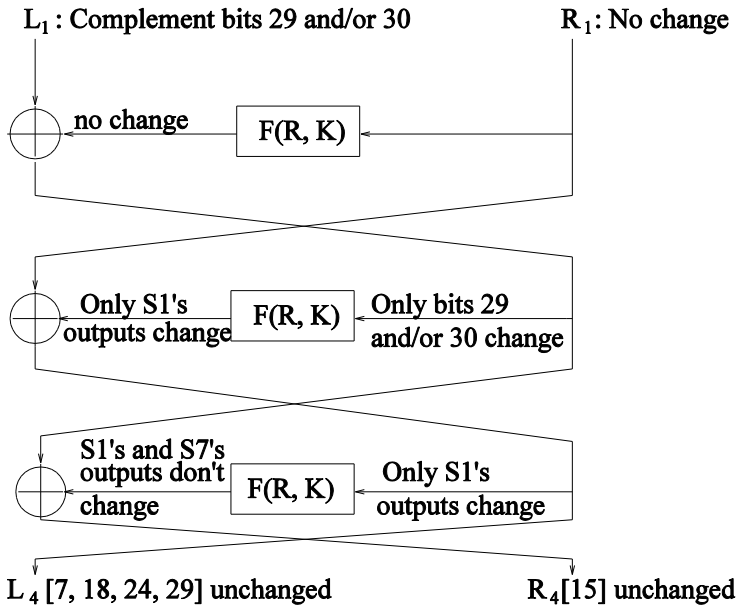


Figure 2 Differential characteristic for differential-linear attack



In Figure 2, we see that bits 7, 18, 24 and 29 of L_4 and bit 15 of R_4 are unchanged. These are not the only bits which are unchanged. Along with these bits, we also have bits 0, 2, 3, 4, 5, 6, 8, 10, 11, 12, 13, 14, 16, 17, 19, 20, 21, 22, 25, 26, 27, 28, 30 and 31 of L_4 and bits 0, 1, 9, 10, 20, 23, and 25 of R_4 unchanged, using the differential characteristic in Figure 2. We need to construct parity relations having different biases in order to improve the existing Matsui optimal parity relation

$$(L_4[7,18,24,29] \oplus R_4[15]) \oplus (L_7[15] \oplus R_7[7,18,24,29]) = 0 \quad (1)$$

which has a bias of $\epsilon = 1.95 \times 10^{-1}$ (by Matsui's Piling Up Lemma). Let us look at parity relations involving these unchanged bits. The parity relations that are used involving these bits are of the form:

$$(L_4[V] \oplus R_4[W]) \oplus (L_7[X] \oplus R_7[Y]) = 0 \quad (2)$$

with bias ϵ (using Matsui's Piling Up Lemma) where *V*, *W*, *X*, *Y* and ϵ and the cumulative sum of squares of the bias ϵ are given in Table 1. We can see that the 3 linear approximations in Table 1 are amongst the best linear approximations that we can use in differential-linear cryptanalysis of 8-round DES.

The main idea is to search for several correct key bits used in 8-round DES encryption. In order to do this, we need to know which of DES's S-boxes are involved. The values of *X* in Table 1 indicate which key bits we can search for. In Table 1, only the first and second linear approximations involve the same S-box, i.e. S_1, because bit 15 and bit 1 (in column *X* in Table 1) are output bits from S_1. This means that we can search for the 8-round DES subkey, $K_{8,1}$ using these two approximations. Although by using these two approximations we can recover only the same key bits, the advantage is that we are able to reduce the number of required chosen plaintext pairs. The third linear approximation in Table 1 involves S_6 so that we can search for the subkey $K_{8,6}$. Altogether we should be able to recover a total of 12 key bits from using these 3 linear approximations.

However, in order to use these approximations, equation (2) needs to be satisfied for each of the linear approximations. Equation (2) depends on the xor (exclusive or) of certain key bit values. Parity relations need to be satisfied for all the linear approximations in Table 1. The parity relation for the first linear approximation is $K_5[22] \oplus K_7[22] = 0$. For the second linear approximation, the parity relation is $K_5[2] \oplus K_7[2] = 0$. In the third linear approximation, we notice that the ϵ value in Table 1 is negative so the parity relation is $K_5[2] \oplus K_7[4] \oplus 1 = 0$. As $K_5[2]$ is involved in both the second and third linear approximations, there are only 5 key bits involved in these parity relations.

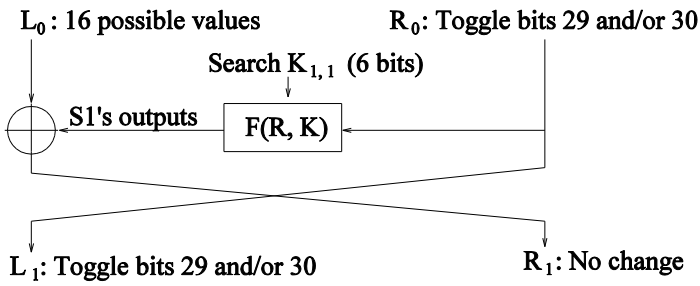
Table 1 Values of *V*, *W*, *X*, *Y* and ϵ for $K_{8,1}$ and $K_{8,6}$ key search

No.	*V*	*W*	*X*	*Y*	ϵ	Cumulative $\sum \epsilon^2$
1	7,18,24,29	15	15	7,18,24,29	1.95×10^{-1}	3.80×10^{-2}
2	5,11,17,27	1	1	5,11,17,27	9.58×10^{-2}	4.72×10^{-2}
3	5,11,17,27	1	3	5,11,17,27	-1.10×10^{-1}	5.93×10^{-2}

3. Implementation

In (Langford and Hellman, 1994; Langford, 1995) a plaintext structure was used to generate the chosen plaintext pairs required for differential-linear cryptanalysis of 8-round DES. We decided not to use any plaintext structure in our experiment. Figure 3 demonstrates that we generate the required chosen plaintext pairs in the first round of 8-round DES. For each plaintext, say *P*, we get another plaintext, say *P′*, with bits 29 and/or 30 in the right half of *P′* toggled and the bits 1, 9, 15 and 23 of the left half of *P′* changed. Note that each half of *P* or *P′* is 32 bits in length, starting from bit 0 to bit 31, with bit 0 being the rightmost bit. The right halves of *P* and *P′* go into the *F* function of the first round of DES together with the first 48 bit subkey of the 8-round DES key, *K*, to produce a differential output, say Δ. We use this Δ to produce the left half of *P′* by xor-ing Δ with the left half of *P*. We finally xor the left halves of *P* and *P′* with the two respective outputs of the function *F* to produce a differential output of zero, which is required for the next stage in differential-linear cryptanalysis of 8-round DES as previously mentioned.

Figure 3 First round of 8-round differential-linear attack



In this first round, we are also able to search for the correct 6-bit subkey $K_{1,1}$. Altogether, we should be able to recover the correct key bits of $K_{1,1}$, $K_{8,1}$ and $K_{8,6}$ using this setup and using multiple linear approximations for differential-linear cryptanalysis of 8-round DES.

However, since we have 2 key bits which are duplicated in $K_{1,1}$ and $K_{8,1}$, we are only able to recover a total of 16 key bits. The outcome of our implementation involving multiple linear approximations is a key ranking of all possible key bits.

4. Prediction

According to Matsui's rule of thumb, if we use only a single optimal linear approximation for the linear part of the differential-linear cryptanalysis of 8-round DES we can estimate the number of plaintext pairs required with a very high percentage of success using the expression $8/(r-\frac{1}{2})^2$ where $r=\frac{1}{2}+2\epsilon^2$ and ϵ is the bias for the optimal linear approximation. So, as the bias $\epsilon = 1.95 \times 10^{-1}$ for the first linear approximation in Table 1, we predict that the number of plaintext pairs required is about 1384 pairs. By multiplying our initial estimate of the required plaintext pairs using a single linear approximation, i.e. 1384 pairs, with the square of the bias using the first linear approximation and then dividing it by the cumulative sum of squares of the bias using the first two linear approximations in Table 1, i.e. $\frac{3.80\times10^{-2}}{4.72\times10^{-2}} \times 1384$, we estimate that we can reduce the plaintext pair requirement to about 1115 plaintext pairs without reducing the percentage of success. Similarly, if we use all three linear approximations in Table 1, we predict that we would require about 887 plaintext pairs when using the same formula given above, except that we divide by the cumulative sum of squares of the bias using three linear approximations instead of dividing by the cumulative sum of squares of the bias using two linear approximations, i.e. $\frac{3.80\times10^{-2}}{5.93\times10^{-2}} \times 1384$.

5. Experimental results

Table 2 gives the results of our implementation of the multiple linear approximations method in differential-linear cryptanalysis of 8-round DES. The results are based on performing 100 trials using 100 random keys (with certain key bits set to specific values as described above). Table 2 indicates the position of the correct key bits out of 65536 possible key bit positions. As we can see from Table 2, the results we obtained conformed to the predictions we had previously made.

Table 2 Recovery of 16 key bits using multiple linear approximations in differential-linear cryptanalysis of 8-round DES

No. of plaintext pairs used	1384	1115	887
Percentage of success	100%	100%	100%
Average position in key ranking list	1.00	1.00	1.00

6. Discussion

In Section 2, we used three linear approximations to perform the cryptanalysis. In fact there are four linear approximations we could use, but we opted not to use the fourth one because it would be unwieldy for a normal PC, as we would be searching a keyspace of 2^{28} for the correct key bits. If we use only three linear approximations, as we have done in this article, we would be searching a keyspace of only 2^{16}.

In Section 3, we developed an implementation which does not require the use of any plaintext structures, because with plaintext structures we are restricted to only multiples of 64

plaintext pairs, whereas without it we do not have any such restriction. On the other hand, the use of plaintext structures has the advantage that all remaining pairs are generated from one particular pair. In our implementation, by contrast, we just generate a plaintext pair and modify it accordingly, then move on to generate the next plaintext pair, and so on, until the required number of plaintext pairs has been reached. Nonetheless, the principal result of this article is that we have developed an alternative implementation which we can use to predict the plaintext pair requirements and to produce the expected results.

References

Kaliski, B.S. and Robshaw, M.J.B. 1994. Linear Cryptanalysis Using Multiple Linear Approximations. *Advances in Cryptology – Proceedings of Crypto '94*. *Lecture Notes in Computer Science* 839, 26-39.

Kaliski, B.S. and Robshaw, M.J.B. 1995. Linear Cryptanalysis Using Multiple Linear Approximations and FEAL. *Fast Software Encryption*. *Lecture Notes in Computer Science* 1008, 249-264.

Langford, S.K. and Hellman, M.E. 1994. Differential-Linear Cryptanalysis. *Advances in Cryptology – Crypto '94*. *Lecture Notes in Computer Science* 839, 17-25.

Langford, S.K 1995. Differential-Linear Cryptanalysis and Threshold Signatures. Ph.D Thesis, Stanford University, USA.

Matsui, M. 1993. Linear Cryptanalysis Method for DES Cipher. *Advances in Cryptology – Eurocrypt '93. Lecture Notes in Computer Science* 765, 386-397.

Matsui, M. 1994. The First Experimental Cryptanalysis of the Data Encryption Standard. *Advances in Cryptology – Crypto '94. Lecture Notes in Computer Science* 839, 1-11.

Matsui, M. 1995. On Correlation between the Order of S-Boxes and the strength of DES. *Advances in Cryptology – Eurocrypt '94*. *Lecture Notes in Computer Science* 950, 366-375.

National Bureau of Standards 1977. Data Encryption Standard. Federal Information Processing Standard 46.

FIRST REPORT OF *PERONOSCLEROSPORA SORGHI* CAUSING DOWNY MILDEW ON MAIZE (*ZEA MAYS*) IN BRUNEI DARUSSALAM

F. Hamdan, N.A. Hj Mohd Noor, J. Jormasie, G. Athikesevan and K.W. Liew [*]

Department of Agriculture and Agrifood,
Ministry of Industry and Primary Resources,
Brunei Darussalam.

[*]Associate, CAB International, SE Asia & East Asia Regional Centre

Corresponding Address: Plant Pathology Laboratory, Crop Protection Unit,
Brunei Agricultural Research Centre, Department of Agriculture and Agrifood,
Brunei Darussalam.
Email: plantpatho@gmail.com

Maize (*Zea mays*) is an economically important crop in Brunei Darussalam, as sweet corn accounts for about 41.3% of the overall production and 43.4% of the overall market value of miscellaneous crops grown in Brunei Darussalam (Department of Agriculture and Agrifood Brunei, 2010). In March 2011, about 5% of maturing sweet corn plants in a farm in Brunei Muara District (Latitude 4°47'40.45"N, Longitude 114°50'36.94"E) were observed to be stunted in growth with extensive longitudinal chlorotic leaf symptoms. The upper and lower leaf surfaces of the younger leaves were covered with a white downy growth which was diagnosed presumptively as Downy Mildew. Downy Mildew had never before been recorded in maize crops in Brunei Darussalam (Peregrine and Ahmad, 1982).

Further investigations were carried out to identify the pathogen. The symptoms on the diseased plants included leaves of older infected plants that showed mottling, chlorotic streaking, lesions and white stripes which eventually shredded the lamina. A white downy growth was often observed on both leaf surfaces but was more common on the lower leaf surface. Infected plants were often stunted, and had malformed reproductive organs (tassels and ears) and abnormal seed set. Infected seedlings had chlorotic leaves, with extreme stunted growth followed by premature death.

Fresh leaf samples were dried in a plant press, and specimens were deposited in the Herbarium at the Plant Pathology Laboratory, Department of Agriculture and Agrifood (National Collection of Fungi No. 1434, 1436 and 1439). Mycelial growth on infected leaves were gently scraped off and also lifted using adhesive cellotape, mounted onto glass slides and then stained before microscopic examination. Based on the morphology of the conidia and conidiophores, the pathogen was identified as a member of the family Peronosporaceae, presumptively either *Peronosclerospora sorghi* or *P. sacchari*. Subsequent microscopic measurements of the oval to almost spherical conidia gave a range of 15.0-30.0 µm x 7.5-20.0 µm. This conidia size is within the described conidia size for *P. sorghi* (14.4-27.3 µm x 15-28.9 µm), but slightly smaller than the conidia of *P. sacchari* (25-41 µm x 15-23 µm) (Shurtleff, 1980). *P. sorghi* has the following morphology: Sporangiophores are 180-300 µm long, erect, hyaline, bloated, dichotomously branched 2 to 3 times, and septate near the base (Shurtleff, 1980).

Downy mildew caused by *P. sorghi* is an economically damaging disease of maize in the tropics and subtropics of Asia, Central and South America (Shurtleff, 1980; CABI, 2013). Its prevalence is particularly acute in South East Asia, with an estimated 20%-80% of all maize

harvests affected by it in Indonesia (Mikoshiba, 1983) and about 8% of the maize crop in the Philippines lost to downy mildew each year (Sharma et al., 1993).

A national survey in 2012 of this particular disease showed that the incidence in Brunei Darussalam is still fairly limited, indicating that eradication of the pathogen in this country is a possibility. In the interim, reduction of the inoculum is emphasised and farmers are advised to destroy young infected plants. To manage yield losses, susceptible sweet corn varieties are being phased out and, currently, varietal screening for tolerant and resistant maize varieties is being conducted.

To the best of our knowledge, this is the first time that *Peronosclerospora sorghi* has been reported in Brunei Darussalam.

References

CABI, 2013. *Peronosclerospora sorghi* datasheet. Wallingford, UK: CAB International.

Department of Agriculture and Agrifood, 2013. Brunei Darussalam Agriculture statistics in brief - 2012. Brunei Darussalam, Department of Agriculture and Agrifood.

Mikoshiba, H. 1983. Studies on the Control of Downy Mildew Disease of Maize in Tropical Countries of Asia. *Technical Bulletin of the Tropical Agricultural Research Center* No. 16.

Peregrine, W.T.H. and Ahmad, K.B. 1982. Brunei: A first annotated list of plant diseases and associated organisms. *CMI Phytopathological Papers* 27, 1-87.

Sharma, R.C., De Leon, C, and Payak, M.M. 1993. Diseases of maize in South and South-East Asia: problems and progress. *Crop Protection* 12(6), 414.

Shurtleff, M.C. 1980. Compendium of Corn Diseases, second edition. APS Press.

SCIENTIA BRUNEIANA

NOTES TO CONTRIBUTORS

Manuscript Submission and Specifications

Scientia Bruneiana is published once a year. The deadline for submission of manuscripts is the **end of December**.

Manuscripts should be submitted to the Associate Editor, Dr Malcolm Anderson, Faculty of Science, UBD, either in the form of hard or as an electronic copy (manderso@fos.ubd.edu.bn). An electronic version should be in MS Word or a similar format.

Papers will be refereed prior to acceptance. Authors are welcome to suggest potential international referees.

Articles outlining original research findings as well as mini-review articles are welcomed. There are two special categories: "Brief Communications" and "Research Notes". Contributions to either category should be 300 to 1000 words long (no more than 3 pages in length). The "Research Notes" section is earmarked for summaries of the results and outcomes of projects receiving UBD Science Faculty research grants.

Manuscripts should be written in English (British or American). All manuscripts should be in 12pt Times New Roman, DOUBLE-SPACED and A4 formatted. The first page should include the TITLE of the article, author's names and addresses only. No other text should be given on the first page of the manuscript.

Example format:

CRYPTOGRAPHY: CAESAR TO RSA

Vasudevan Mangalam

Department of Mathematics, Faculty of Science, Universiti Brunei Darussalam

The second page of the manuscript should include the ABSTRACT only.

Original research articles should be formatted to include the following sections: INTRODUCTION, MATERIALS AND METHODS, RESULTS, DISCUSSION, ACKNOWLEDGMENTS and REFERENCES. Review articles will obviously not conform to this format. In the case of other submissions where the above format may be unsuitable, you are advised to contact the editor prior to submitting the article.

The reference list should only include references cited in the text and should be arranged alphabetically by the author's name or surname followed by initials of surname or name, respectively.

Journal Article:

Norjaidi, P.T. 2002. Pasang Emas: A software revival of the traditional game of pasang. Bruneiana: Anthology of Science Articles 3, 6-14.

Hanfling, B. and Brandl, R. 2000. Phylogenetics of European cyprinids: Insights from allozyme electrophoresis. Journal of Fish Biology 57, 265-276.

Book:

Roberts, T.R. 1989. The freshwater fishes of Western Borneo (Kalimantan Barat, Indonesia). California Academy of Sciences, San Francisco. 209 pp.

Campbell, N.A. and Reece, J.B. 2002. Biology, sixth edition. Benjamin Cummings, San Francisco. 1247 pp.

Article or Chapter in a Book:

Mayden, R.L. 1997. A hierarchy of species concepts: the denouement in the saga of the species problem. In: Species: The Units of Biodiversity, M.F. Claridge, H.A. Dawah, and M.R. Wilson (eds), pp. 381-424. Chapman and Hall, London.

Dissertation or Thesis:

Hymel, T.M. 1985. Water quality dynamics in commercial crawfish ponds and toxicity of selected water quality variables to *Procambrus clarkii*. Master's thesis. Louisiana State University, Baton Rouge, Louisiana, USA.

All TABLES and FIGURES and FIGURE LEGENDS should be given at the end of the manuscript after the reference list, in the order that they appear in the paper. These must be appropriately numbered and Tables must include a title. It is important that you do not include tables and figures in the text body.

Manuscripts that do not conform to the above instructions will be returned without review.