International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:4, No:3, 2010

New DES based on Elliptic Curves

Ghada Abdelmouez M., Fathy S. Helail, and Abdellatif A. Elkouny

Abstract**—**It is known that symmetric encryption algorithms are fast and easy to implement in hardware. Also elliptic curves have proved to be a good choice for building encryption system. Although most of the symmetric systems have been broken, we can create a hybrid system that has the same properties of the symmetric encryption systems and in the same time, it has the strength of elliptic curves in encryption. As DES algorithm is considered the core of all successive symmetric encryption systems, we modified DES using elliptic curves and built a new DES algorithm that is hard to be broken and will be the core for all other symmetric systems.

Keywords**—**DES, Elliptic Curves, hybrid system, symmetric encryption.

I. Introduction

WITHOUT any doubt the first and the most significant modern symmetric encryption algorithm is that contained in the Data Encryption Standard (DES) [1]. DES was published by the United States' National Bureau of Standards in January 1977 as an algorithm to be used for unclassified data. The algorithm had been used in banks for funds transfer security [2]. DES encryption was broken in 1999 by Electronics Frontiers Organization. This resulted in NIST issuing a new directive that year so Triple DES that is three consecutive applications of DES was appeared but as DES became unsecured, researchers proposed a variety of alternative designs, which started to appear in the late 1980s and early 1990s: examples include RC5, Blowfish and IDEA. Most of these designs kept the 64-bit block size of DES; DES itself can be adapted and reused in more secure schemes [3].

In 2001, after an international competition, NIST selected a new cipher, the Advanced Encryption Standard (AES), as a replacement. The algorithm which was selected as the AES was submitted by its designers under the name Rijndael. Other finalists in the NIST AES competition included RC6 and Twofish.

In our study, we will go back to the past and try to bring back DES to life by using elliptic curves which are considered the simplest possible curves after lines and conics. Elliptic curves over finite fields provide an inexhaustible supply of finite abelian groups. Such curves involve elementary arithmetic operations that make it easy to implement (in either hardware or software) [4]. They are generally more secure than others. Elliptic curves could easily be applied to DES to improve its performance and make it hard to be broken.

In the coming sections, you will see how we modify DES to increase its complexity and how to make it secured. The modifications will be accomplished through four stages. The 1st stage will be the base for the last three succeeded stages.

F. A. Ghada Abdelmouez M. is Senior Teaching Assistant in Basic Science dept. in German University in Cairo (GUC), Egypt; (e-mail: ghelhady@yahoo.com or Ghada.abdelhady@guc.edu.eg).

S. B. Fathy S. Helail is Professor in Basic Science dept. in German University in Cairo (GUC), Egypt; e-mail: fathy.helail@guc.edu.eg

T. C. Abdellatif A. Elkouny is Assistant Professor in Air defense Research and Development Center (Egyptian Army); (e-mail: aelkouny@gmail.com).

II. Stage 1: Regular DES

DES is a block cipher, meaning it operates on plaintext blocks of a given size (64-bits) and returns cipher-text blocks of the same size. Each block of 64 bits is divided into two blocks of 32 bits each, a left half block L and a right half R. The operation of the DES can be described in the following three steps:

Step 1: Initial permutation IP:

The 64 bits of the input block to be enciphered are first subjected to IP. This rearranges the bits according to matrix, where its entries show the new arrangement of the bits from their initial order.

Step 2: Iterate the following 16 rounds of operations as in (1).

$$L_i \leftarrow R_{i-1}$$
$$R_i \leftarrow L_{i-1} \oplus f(R_{i-1}, K_i) \quad for\ i = 1,2,\ldots,16 \tag{1}$$

Here k_i is called "round key" which is a 48-bit substring of the 56-bit input key; f is called "S-box Function" ("S" for substitution) and is a substitution cipher. This operation features swapping two half blocks, that is, the left half block input to a round is the right half block output from the previous round [2].

Step 3: The result from round 16, (L_{16}, R_{16}), is input to the inverse of IP to cancel the effect of the initial permutation. The output from this step is the output of the DES algorithm. We can write this final step as in (2).

$$\text{Output Block} \leftarrow IP^{-1}(R_{i16}, L_{16}) \tag{2}$$

Please pay a particular attention to the input to IP–1:
the two half blocks output from round 16 take an additional swap before being input to IP^{-1}[5].

These three steps are shared by the encryption and the decryption algorithms, with the only difference in that, if the round keys used by one algorithm are $k_1, k_2, \ldots, k_{16}$, then those used by the other algorithm should be $k_{16}, k_{15}, \ldots, k_1$. This way of arranging round keys is called "key schedule". All these steps are shown in fig. 1. These steps are

accomplished in MATLAB to be the base of our work.

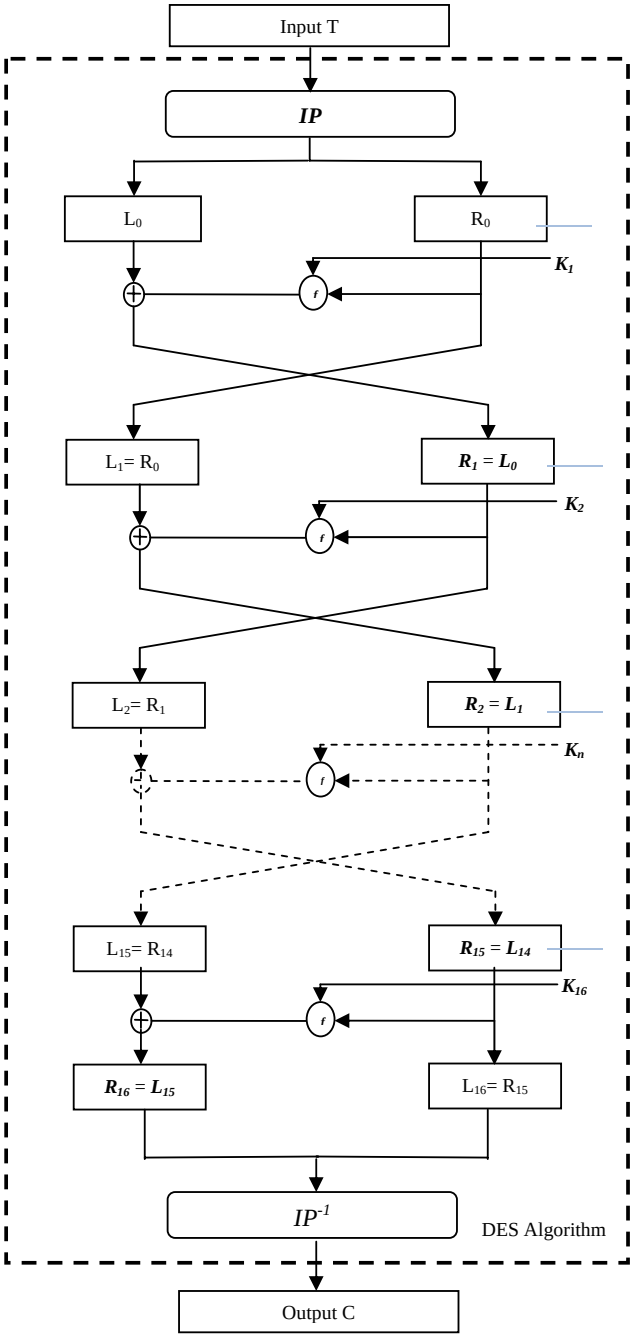

Fig. 1 Encryption algorithm

III. Elliptic Curves (EC)

An elliptic curve is the set of solutions (x, y) for (3)

$$y^2 = x^3 + ax + b \quad (3)$$

where a and b are constants. Note that the right-hand side is a special cubic polynomial. The left-hand side is a quadratic term. The definition of elliptic curve also requires that the curve be non-singular. Geometrically, this means that the graph has no cusps or self-intersections. Algebraically, this involves calculating the discriminant shown in (4) in order to prevent repeated roots on the right-hand side [6].

$$4a^3 + 27b^2 \neq 0 \quad (4)$$

The curve is non-singular if and only if the discriminant is not equal to zero

Despite their simple form, elliptic curves have been studied for many years and have many significant applications in mathematics. Maybe one of the most interesting results related to the application of elliptic curves is that they are used to prove Fermat's Last Theorem [7].

ECC is particularly beneficial for application where:

Computational power is limited (wireless devices, PC cards)

Integrated circuit space is limited (wireless devices, PC cards)

High speed is required [8].

Let's examine some properties of elliptic curves to generate points used in encryption.

A. The group law

One of the interesting and powerful features of elliptic curves is that by using a special 'add' operation, any two points added together will result in a third point on the same curve.

Adding points on an elliptic curve:

Given two points $u = (x_1, y_1)$ and $v = (x_2, y_2)$ on the elliptic curve (see Fig. 2), the point u + v is calculated by the following steps:

Step 1: Drawing a straight line through u and v and finding the third intersecting point w;

Step 2: Drawing a vertical line through w (and O) and finding the third intersecting point u + v [9].

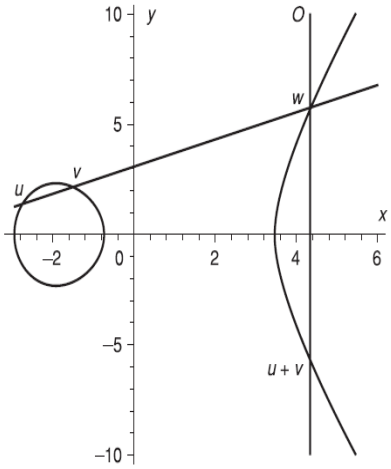

Fig. 2 The addition of two points on EC

To make this rule work, we assume the following:

An extra imaginary point O on the curve, also known as the identity element or the point at infinity. It has no specific (x,

y) coordinates, but one might imagine that its location is infinitely high above the curve where all vertical lines converge.

A line tangent to a point on the curve is said to intersect the point twice. Think of the tangent as the limit of a line through two distinct points as the points approach each other [4].

Fig. 4 shows a picture of examples of a family of standard elliptic curves to study the singularity.

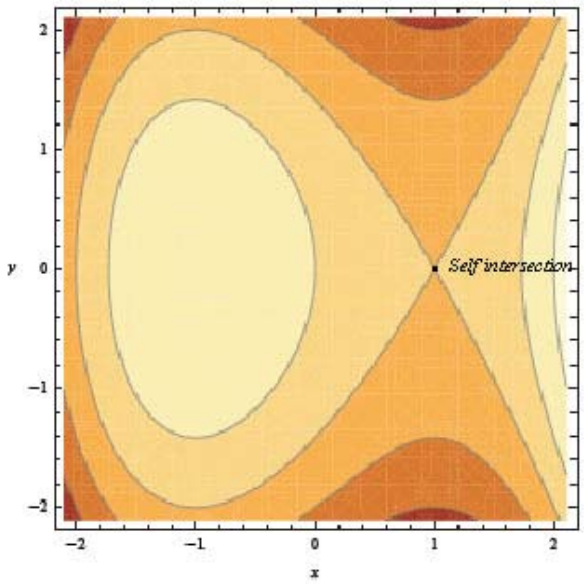

Fig. 4(a) $y^2 = x^3 - 3x + b$

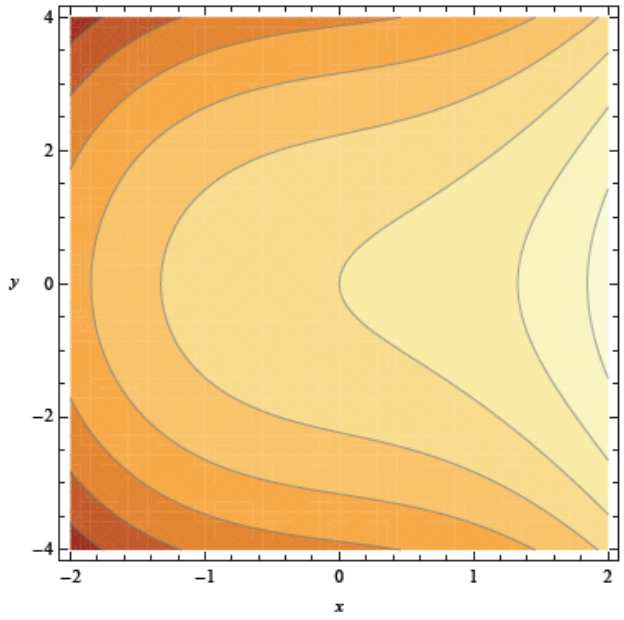

Fig. 4(b) $y^2 = x^3 + 2x + b$

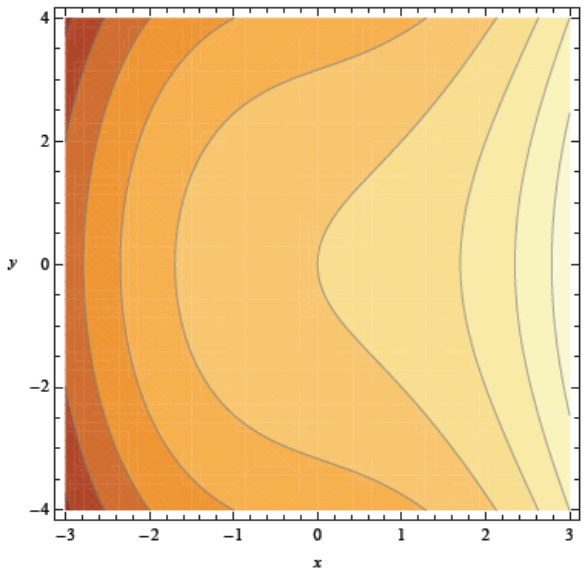

Fig. 4© $y^2 = x^3 + 3x + b$

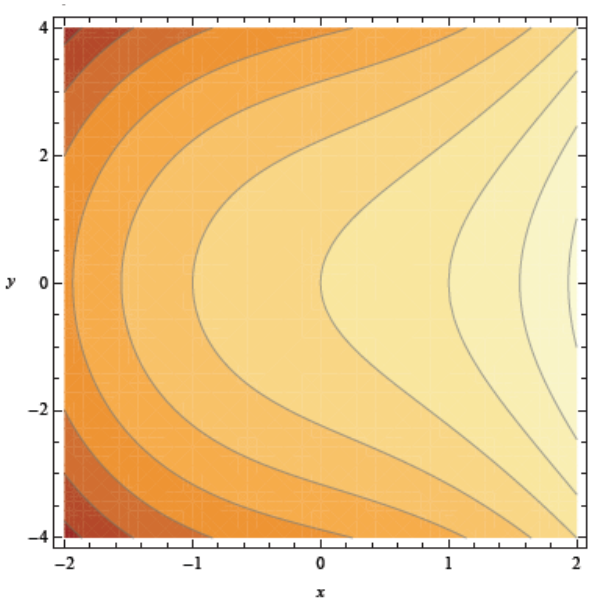

Fig. 4(d) $y^2 = x^3 + 4x + b$

Fig. 4 Family of standard elliptic curves

You can observe in fig. 4(a) that the curve has self intersection if b=2 or -2 so the curve in this case is called singular and the singular curves are not preferred in encryption as it is so simple to be broken. In our algorithm, all stages, we should use non-singular elliptic curves that have no self intersection for encryption. Some of these curves, which are used in our algorithm, are shown in fig. 4(b), 4(c) and 4(d).

IV. Stage 2: S-boxes Modification

We will modify only in the cipher function using EC's as in the following:-

A. The Cipher Function f

A sketch of the calculation of $f(R, K_n)$ is given in Fig. 5.

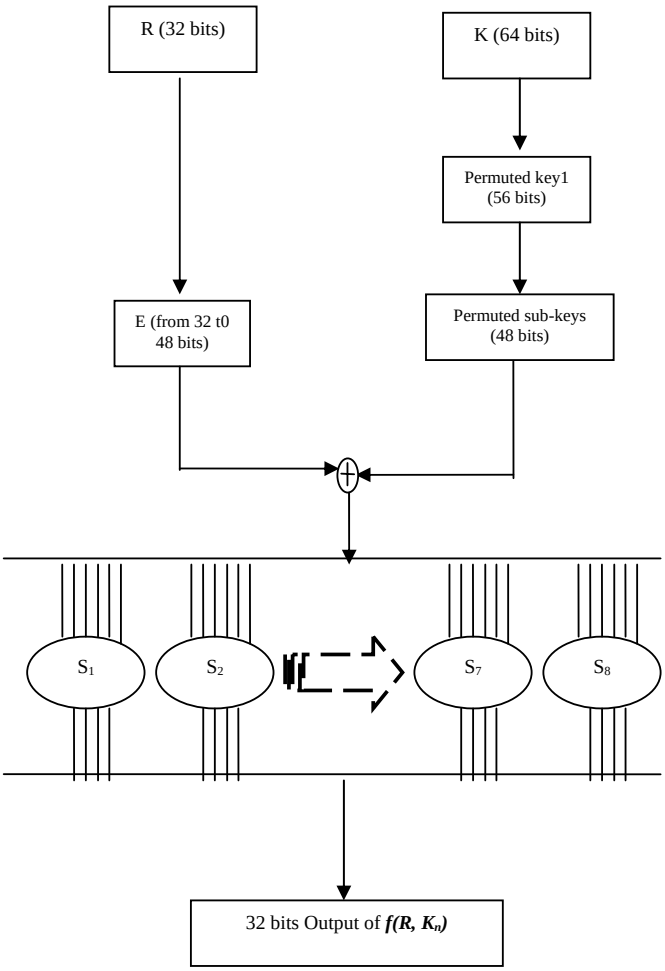

Fig. 5 Cipher function in Modified DES: stage 2

DES operates on the 64-bit blocks using key sizes of 56-bits. The keys are actually stored as being 64 bits long, but every 8th bit in the key is not used (i.e. bits numbered 8, 16, 24, 32, 40, 48, 56, and 64). However, we will nevertheless number the bits from 1 to 64, going left to right, in the following calculations. But, as you will see, the eight bits just mentioned get eliminated when we create sub-keys.

The 64-bit key is permuted firstly to be transformed into 56-bit key. Then it will be subjected to some shifting to the left to generate the 16 sub-keys and finally all sub-keys will be permuted again to be transformed into 48-bit key. This function takes a block of 32 bits (R) XORing with the secret key (K_n) as input written as 8 blocks of 6 bits each. Each of the unique selection functions S_1, S_2,..., S_8, takes a 6-bit block as input and yields a 4-bit block as output. S-boxes in DES are arrays whose cells are known but in this stage, the cells of S_1 for example, are generated from different elliptic curves as the x-coordinates of double of their base points and the cells of S_2 are generated from the same elliptic curves as the y-coordinates of these points. In the same way, we can generate S_3 and S_4, S_5 and S_6, and finally S_7 and S_8.

V. Stage 3: Sub-keys Modification

We will keep everything in stage 2 except we will generate the sub-keys using elliptic curve equation. So we will cancel the regular DES key and all operations applied on it. The 64-bit key will be generated from the elliptic curve and we can generate the sub-keys using the addition of points on the same elliptic curve.

Decryption in all previous steps is simply the inverse of encryption, following the same steps as above, but reversing the order in which the sub-keys are applied exactly as the regular DES.

VI. Stage 4: New DES

We will keep everything in stage 3 except that the input will be represented in another elliptic curve equation. We used an elliptic curve to mask the plaintext.

A. Input masking

To mask an ordered pair of elements (m_1, m_2) with an elliptic curve means to alter the pair by multiplying m1 and m2 with the x and y coordinate, respectively, of some point on the curve.

Consider that $T = (x_1, x_2)$ is the pair of plaintexts, each plaintext (x_1, x_2) represents two alphabetic characters in this case, and "a" corresponds to 1, "b" to 2, "c" to 3, ….., "z" to 26, with the point $(c_1, c_2) = K P$ where P is the selected base point of the used elliptic curve and K is a secret random integer number that is varying per plaintext. It depends on the order of the used elliptic curve.

The masking process is used actually as an encryption method using elliptic curves and it's easier in decryption, so we are going to use the masking process in our algorithm. The 64 bits of the input block to be enciphered are first subjected to masking process.

The input to the DES scheme is $M= (y_1, y_2) = (c_1x_1 \bmod p, c_2x_2 \bmod p)$. M is considered the permuted input to the calculations of 16 rounds to produce the pre-output block. We select a non-singular elliptic curve to carry out this process. Then we complete the DES algorithm as previous, the output (M) is subjected to an initial permutation IP and complete as what happen in the regular DES. Fig. 6 explains the encryption algorithm in this stage.

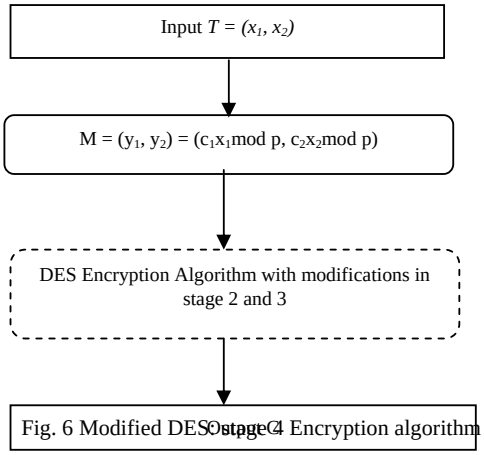

Fig. 6 Modified DES: stage 4 Encryption algorithm

B. *Decryption Algorithm*

Decryption algorithm is also the inverse of the encryption but slightly different as shown in fig. 7

.

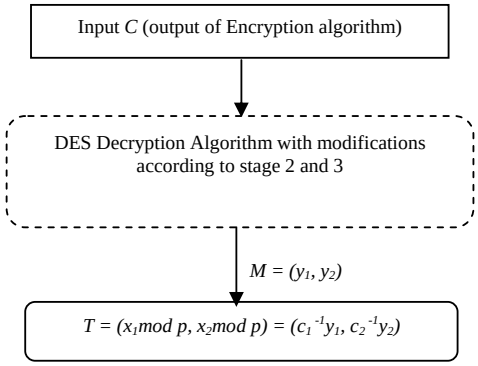

Fig. 7 Modified DES: stage 4 Decryption algorithm

The output of the DES decryption algorithm $M = (y_1, y_2)$ is subjected to the inverse of the masking process according to (5) and (6).

$$y_1 c_1^{-1} = (c_1 x_1) c_1^{-1} = x_1 \bmod p \quad (5)$$

$$y_2 c_2^{-1} = (c_2 x_2) c_2^{-1} = x_2 \bmod p \quad (6)$$

VII. Comparison between stages

From all previous stages, we can see that starting from stage 2 the cipher function varied according to changing of the parameters of EC's also the sub-keys are varied from the 3rd stage. The comparison between all stages will be explained in table I.

In stage 4, we have been built a new DES based on many elliptic curves equations. This new algorithm will be the core of symmetric cryptosystems like Blowfish and Two-fish for example. As soon as, we change the parameters of the equation, we get different elliptic curves, different points and of course, different cipher functions and keys. By this new algorithm, we made a complication in DES computations to be hard to be broken and it is still easy to be implemented on hardware like DSP's, Microprocessors, and any embedded systems.

TABLE I
COMPARISON BETWEEN STAGES

	Time elapsed	Key size In bits	Cipher function	Input format
Stage 1	0.072	56 (fixed)	Fixed	
Stage 2	0.209	56 (fixed)	Varied	64 bits
Stage 3	0.26	64 (varied)	Varied	
Stage 4	0.49	64 (varied)	Varied	128 bits

Acknowledgment

Ghada Abdelmouez would like to acknowledge the financial support of my study at German University in Cairo. Also she would like to express her sincere thanks and gratitude to her supervisors Prof. Dr. Fathy Saad Helail and Dr. Abdellatif Elkouny.

References

[1] Hugo Fruehauf , "*Encryption Fundamentals*," in *Zyfers*, October 2001.
[2] W. Mao, "*Modern Cryptography Theory and Practice (Book style).*," Prentice Hall PTR July 25.
[3] W. Stallings, "*Cryptography and Network Security Principles and Practices*," Fourth Edition, Prentice Hall, November 16, 2005.
[4] N. Torri, K. Yokoyama, "*Elliptic Curve Cryptosystems*," Fujtsu Sci. Tech. J. pp. 140-146, December 2002.
[5] W. M. Daley, "*Specifications for the data encryption standard (DES),*" FIPS PUB 46-3, Federal information processing standards publication, Reaffirmed, October25th, 1999.
[6] R. Zuccherato, "*Practical Cryptology and Web Security*," Entrust, May 9, 2000.
[7] Pk Yuen, "*Elliptic Curve Cryptography Support in Entrust*," Pearson Education Limited 2006.
[8] D. Hankerson, Menezes. A, Vanstone S., "*Guide to Elliptic Curve Cryptography*," Springer, (c) 2004, Springer-Verlag New York, Inc.
[9] Matthew England, "*The Weierstrass Theory For Elliptic Functions*," Department of Mathematics, MACS Heriot Watt University, Edinburgh, The Burn 2007.