INTERNATIONAL JOURNAL OF COMPUTERS
Issue 4, Volume 6, 2012

Advanced User Authentication Process Based on the Principles of 4th Degree Multibrot Fractal Structure

Ivo Motýl, Roman Jašek, Pavel Vařacha

***Abstract*—** This article is focused on authentication of users inside and outside the information systems. For this purpose is widely used hash function. The proposed process is based on the elements of the fractal geometry. The algorithm here uses a wide range of the fractal sets and the speed of its generation. The system is based on polynomial fractal sets, specifically on the 4th Degree Multibrot. The system meets all the conditions for the construction of hash functions.

***Keywords*—** HASH, hash function, fractal geometry, information system, security, information security, authentication, protection, fractal set.

I. Introduction

Hash functions have in the world of information technology an important role. They are represented in many areas of the information system. Hash functions can be used for example in password section of information system, data identification, integrity control, database comparing and many others solutions.

Hash functions are one-way functions, which must meet defined conditions. A hash function maps string of arbitrary length string of constant length (from a given large amounts of data returns a much smaller amount of data, but it clearly shows the contents of the document). The resulting impression is dependent on all bits of the string. [2] Hashing is the process of taking any input and transforming it into a fixed length string. This output which is obtained is called the hash value/message digest. In informal terms, a hash is a sort of signature/identification for some stream of data which represents the value of the data. It is a one way transformation [3].

The aim of this study was describe how to use the principles of fractal geometry like hash function for secure authentication inside of the information system.

Hash function is a mathematical function that takes a string of arbitrary length and transforms it into a block of fixed-length string. Input block of data for the hash function can be any length, from zero length, over a few dozen bytes to several GB, and much more. All the current modern hash functions use Merkle - Demgard construction iterative hash calculation, which is based on the idea of progressive processing of the message blocks of fixed length. The input message is first completed the required number of bits - that the total length of the entry divisible by the length of one block. Furthermore, the report is written by blocks, the calculation results of each block is called context - Intermediate outturn, which is an input to the next block processing, along with another block of input data [11].

Ivo Motýl is with the Department of Informatics and Artificial Intelligence, Faculty of Applied Informatics, Tomas Bata University in Zlín, nám. T. G. Masaryka 5555, 76001 Zlín, CZECH REPUBLIC; e-mail: motyl@fai.utb.cz

Roman Jašek is with the Department of Informatics and Artificial Intelligence, Faculty of Applied Informatics, Tomas Bata University in Zlín, nám. T. G. Masaryka 5555, 76001 Zlín, CZECH REPUBLIC; e-mail: jasek@fai.utb.cz.

Pavel Vařacha is with the Department of Informatics and Artificial Intelligence, Faculty of Applied Informatics, Tomas Bata University in Zlín, nám. T. G. Masaryka 5555, 76001 Zlín, CZECH REPUBLIC; e-mail: varacha@fai.utb.cz.

II. Problem Formulation

Using of fractal geometry for the authentication process is an alternative to authentication process using the hash function. For proper function of the process is necessary to ensure the following parameters:

- One-way function – For a given message M is very simple compute h = H (M), but the h is computationally impossible to calculate M. This characteristic is very desirable and is used for example for storing passwords. We do not store password, but only stored hash code. When authentication is the point of direct comparison entered and saved passwords compare their hash codes.
- Non-collision function – impossibility to find a variety of M and M ', then the M≠ M ' so that h (M)= h (M '). Two input strings are not allowed to apply the same hash.
- Random oracle – Output of the hash function must be random [1].

A. Principle of the Classical HASH Function in the Authentication process

Fig. 1 shows the basic principle of the HASH function in password processing. This is only basic illustration case of the HASH process. In the modern information systems are items for more sophisticated solutions. One of them is for example Salting process. This is one of many possibilities how to

increase the password security. The password processing in this case is very simple. The password is extended by the several random characters. All string is converted by the hash function to HASH. This HASH string is stored to database.

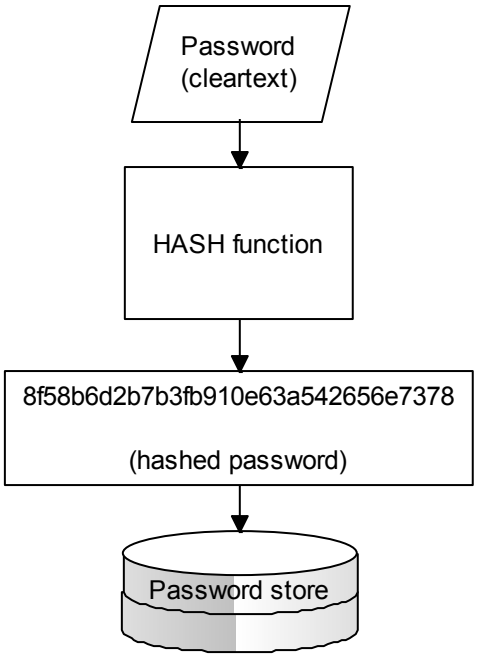

Fig. 1 Password hashing [3]

Fig. 2 shows the login process with the hashed password. The information system contain database with users password. In the first step user enter his secret password into the login form. The entered password is converted by the hash process to hash string. This string is compared with the record in database. If the entered password is correct, the user is successfully logged.

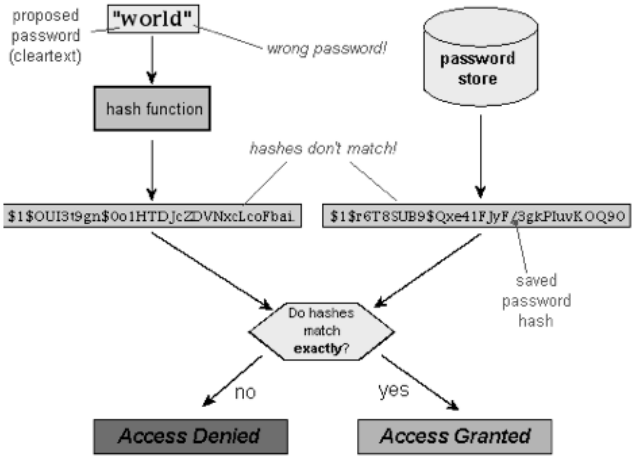

Fig. 2 Login process [3]

B. Alignment of input data

Input string of data processing functions that can be any size. Hash function but only processes data blocks of a length, it is necessary to align the input string with a number of bits that contains the block size. To avoid the possibilities of multiple collisions using a simple conclusion, perhaps the zeroes must be every message, including alignment, unique. Therefore, today, virtually all modern hash functions to align it also enter the length of the original input string. Alignment is performed mostly by the chain to the input data complement binary number one, the required number of zeros, and finally the length of input string in terms of binary bits, all so that the total length including the final alignment is divisible by block size. To save the length of input data is used to align the fixed number of bits, usually 64, for example functions MD4, MD5 and SHA1. It also implies, inter alia, that the input block of data must be less than, in this case, 264bits. But this is a very large number, about 2*1018 bytes, which for present needs enough. Adding itself the length of the input-process chain is called the so-called Merkle - Demgard amplification [11].

C. Merkle – Demgard construction

Compression functions are the very core of hash functions. Compression function takes the input block mi input and the output produces a block of data Hi, which is called context. This context is then input to the compression feature in the next step. Compression function is receiving input earlier context Hi -1 a new block of data mi. Compression function itself is therefore two inputs and one output. The initial value is called the context initialization vector IV, which is one for each hash function firmly determined.

Compression function can be described as a function f: 0.1 x 0.1 m h → (0.1) h while H 0 = I V a H i = f (H i -1, m i). This structure, used by virtually all modern hash functions, called Merkle - Demgard construction, according to two authors who do independently introduced in 1989. In addition to this design, using a compression function, there are structures based on current ciphers, which use two, respectively four compression functions in a single block processing (MDC-2, respectively MDC-4) [11].

D. Construction of compression function

High quality compression function is a good base hash function. Compression function should be very robust to ensure perfect mixing bits of news and one-way. Hash function should be like random oracle. Since the quality of the block cipher F k (x) when the hard key to expected to behave as a random oracle. Furthermore, it guarantees that we know if there is any set of input-output, open-ciphered texts (x, y), then it is not possible to determine (due to complexity) key k. Due to the key block cipher is so one-way. More specifically, for each x is a function $k \rightarrow F$ k (x) one-way. Hence follows the possibility of construction of compression functions as follows: H i = F m i (H i -1), where F is a good block cipher. The advanced functions are used once again the operation. This is the so-called Davies-Meyer compression function design, which amplifies the property undirected even before adding the context of the previous output: H i = f H i -1 m i = F m i H i -1 x o r (H i -1). Output then there is also masked input, which makes it even more difficult for any reverse. An alternative is to use the Matyas – Meyer - Oseas construction,

which added to the input block of data before departure, and Miyaguchi - Preneel construction, which is added to the previous context and the input block of data (g denotes the context of pre-entry / key) [11].

E. Random oracle

Random oracle is a hypothetical device for each entry corresponds to a randomly selected outcome of their field values, but also with the proviso that the same given input represents the same answers. The aim of the proposal compression feature is that such functions behave like random oracles [11].

F. Second level of HASH protection

It is a first level of HASH protection improvements. When you save the password server takes the string, which is called salt. The chain server in some way combine with a password and only the result makes a hash stored in the database. Furthermore, the hash is of course saved the salt. User authentication is then executed so that the server takes input from the user, combines it with salt, the result makes a hash and compare with the database. Salt can be deposited uniformly in the code of the application or database for each user. Stored passwords in a database for each user mean the disadvantage that when obtaining an extract from the database an attacker gets both the hash, and salt. On the other hand, however, an advantage that even if users have the same password will have a different salt, so the resulting hash will be different. Thanks to not know which users have the same password [11].

III. PROBLEM SOLUTION

Polynomial fractals are between the most popular. Their design takes advantage of the attractiveness of areas for various solutions of nonlinear systems. The coordinate system is tested at points belonging to it, whether the rule meet the specified condition. Evaluation of equations, which are based on polynomial fractals, happens iteratively. Iterative cycle can be terminated either after a specified number of iterations, or after the evaluation of test conditions. After the process is the appropriate point in the coordinate system indicated by the ink. Here, depending on the specific application of fractal, if required by the resulting fractal monochrome to colour, such as shade or equal to the number of iterations performed in the evaluation algorithm. [4]

A. Principle of the Classical HASH Function in the Authentication process

To generate fractal impressions of password can be used polynomial fractals described in section 3. For this experiment was used 4th Degree Multibrot. The 4th Degree Multibrot set is a set of complex numbers defined in the following way: [5]

$$D = \left\{ c \in \mathbb{C} \mid \lim_{n \to \infty} Z_n \neq \infty \right\} \tag{1}$$

$$Z_0 = c$$
$$Z_{n+1} = Z_n^4 + c \tag{2}$$

The 4th Degree Multibrot is the set of all complex numbers which fulfilled the condition described above, that is, if the value of the (recursive) function Zn for the value c is not infinite when n approaches infinity, then c belongs to the set. Attractors are related to the "orbit" of the function. This orbit is defined by the path formed by the values of Z at each step *n*. The orbit of Z for a certain value c either tends towards the attractor or not. In this type of fractals a value c causing the orbit of Z to go to the attractor point is considered to be outside the set. [5]

B. Parameters for Fractal Construction

For the construction of fractal hash is necessary to set the initial conditions. Table 1 shows parameters for construction of fractal set. Parameters X and Y specify the coordinates of fractal field. Parameters were found by experimental process. The experimentally determined parameters are used as the basis for creating new parameters for the user authentication experiment. User authentication password (key) can be stored on the security token after the generating process. User will use it in the authentication process.

Table 1 Fractal parameters

X - real part of the operating quadrant	*-0,712578125*
Y - imaginary part of the operating quadrant	*-0,303984375*
Range	*0,00390625*
Number of iterations	*250*

Fig. 3 shows the output of the fractal structure used in the algorithm for advanced user authentication. It is part of the 4th Degree Multibrot. The coordinates for generating this picture was used from Table 1.

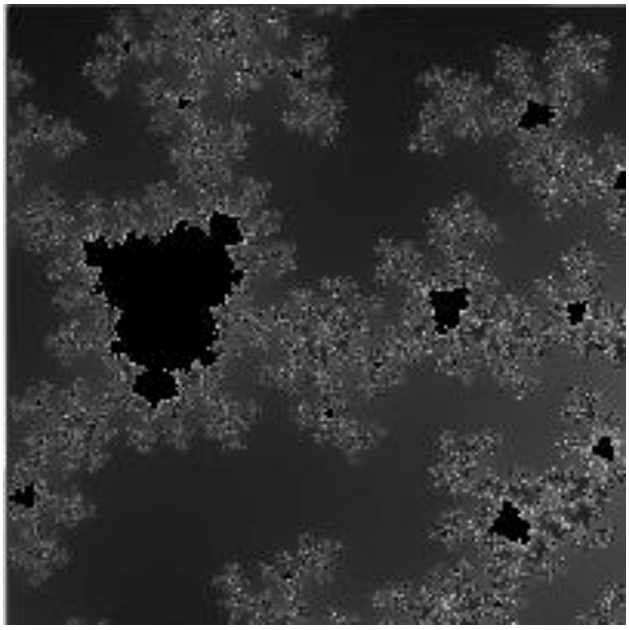

Fig. 3 fractal structure for advanced user authentication

C. TEA algorithm

Group created using fractal algorithms TEA is a very widespread. Competent fractal of this group is defined by its equation. The equation is iteratively carried out according to specified conditions. There is a large set of initial conditions for the creation of fractal this group. Conditions are often intertwined with each other and the fractal generation process depends on their mutual combinations. One case may be, for example determining the number of iterations, while the value of the parameter equation, which must not be exceeded. For this group of fractals is investigated by point location on the desktop or in space. After the process is the relevant point of shade drawn proportional to the number of iterations performed, which was to be done to leave the selected boundaries. Examples include Julia sets, Mandelbrot set and their modifications.

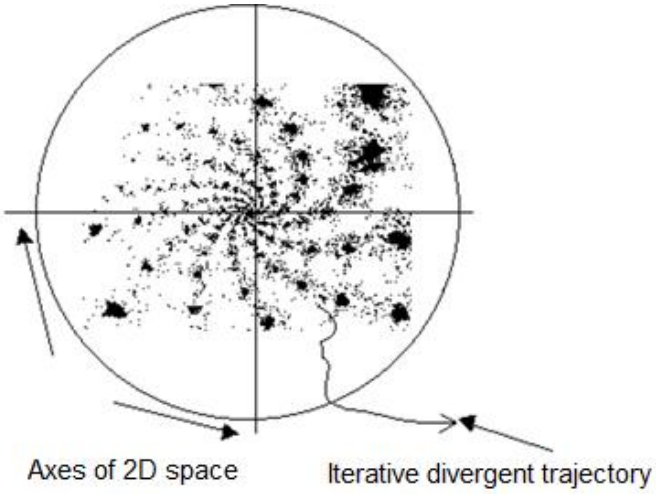

Fig. 4 TEA algorithm [4]

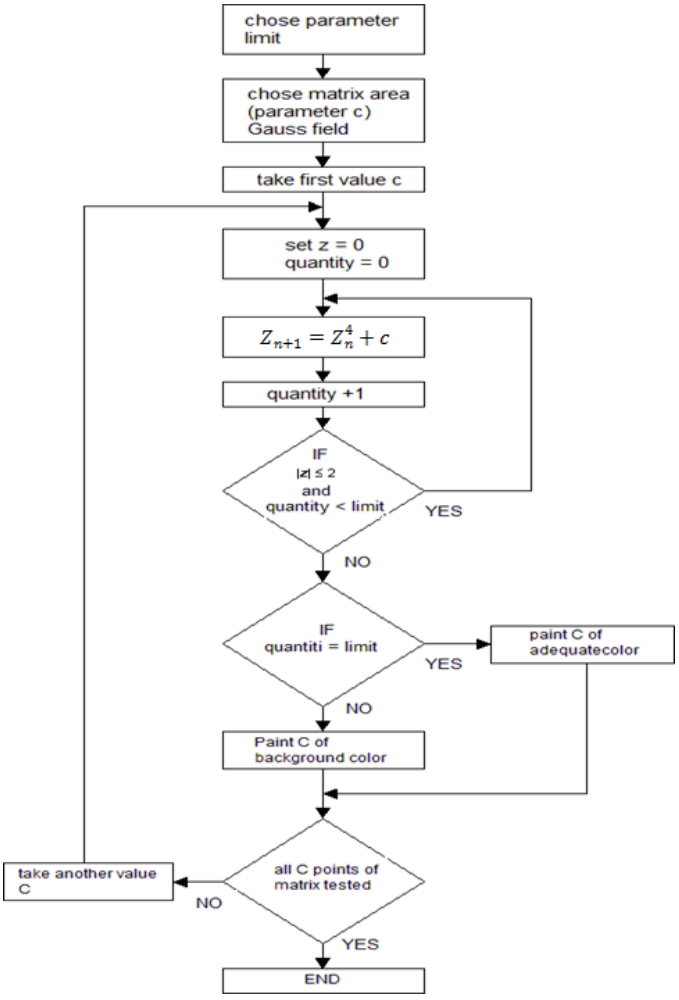

Fig. 5 Generating algorithm

D. Fractal HASH Functions in the Authentication Process

Fig. 6 shows the login process with the fractal hashed password. The information system contain database with users password in the form of fractal. In the first step user enter his secret password into the login form. The entered password is converted to parameters for fractal algorithm. The fractal algorithm produces fractal images in agreement by the initial conditions. This image is compared with the record in database. If the entered password is correct, the user is successfully logged. This is caused by the conformity of the fractal images.

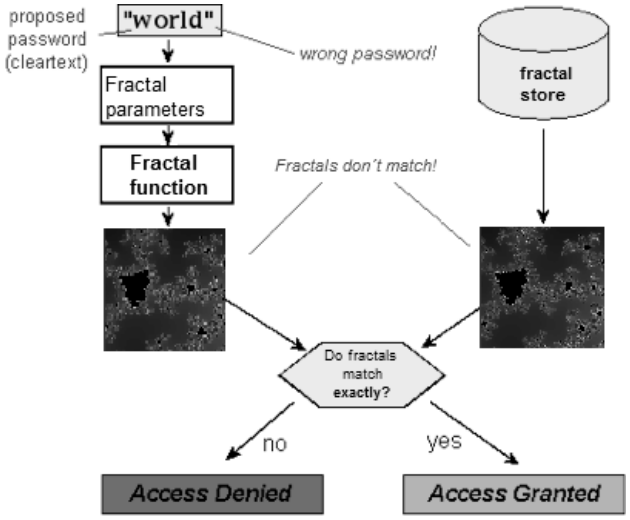

Fig. 6 Fractal authentication process

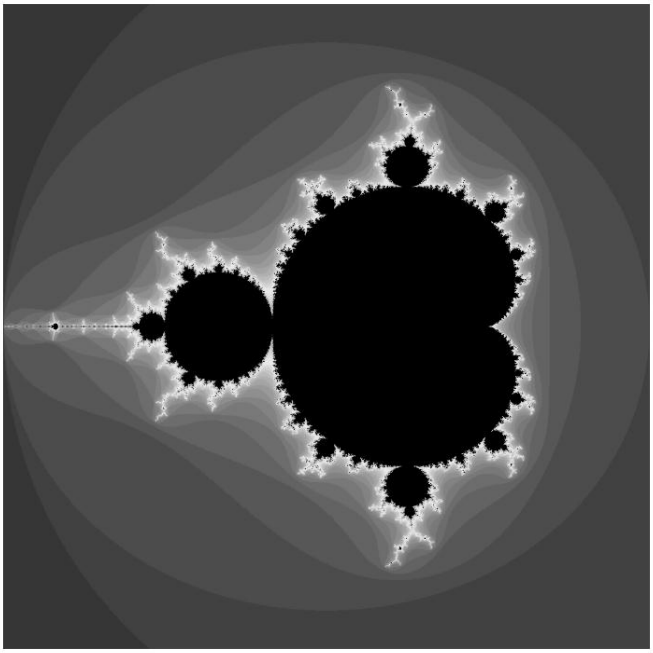

Fig. 7 Mandelbrot set

IV. Conclusion

This article was focused on the possible use of fractal geometry for secure authentication of users. This process is an alternative for the now widely used hash function. The process meets the requirements spoken in the second chapter. The process of authentication and its principles are described in chapter three. If we compare the authentication process using hash function and fractal geometry, we find that in many ways are similar. The size of fractal object can be selected by modifying the function generating the initial conditions for the creation of fractals. The system uses the advantages of fractal geometry, in particular the wide range of fractal sets and the speed of its generation.

The first part of research of HASH function powered by fractal geometry was focused on the Mandelbrot set, described on the previous conference. Present research describes this problematic solved by 4th Degree Multibrot. The research confirmed my expectations. The 4th Degree Multibrot can be applied in the HASH authentication process based on fractal geometry.

Appendix

A. *Fractal structures used along the research – Mandelbrot set*

$$M = \left\{ c \in \mathbb{C} | \lim_{n \to \infty} Z_n \neq \infty \right\} \tag{3}$$

$$\begin{aligned} Z_0 &= c \\ Z_{n+1} &= Z_n^2 + c \end{aligned} \tag{4}$$

B. *Fractal structures used along the research – Julia sets*

$$J = \left\{ c \in \mathbb{C} | \lim_{n \to \infty} Z_n \neq \infty \right\} \tag{5}$$

$$\begin{aligned} Z_0 &= c \\ Z_{n+1} &= Z_n^2 + K \end{aligned} \tag{6}$$

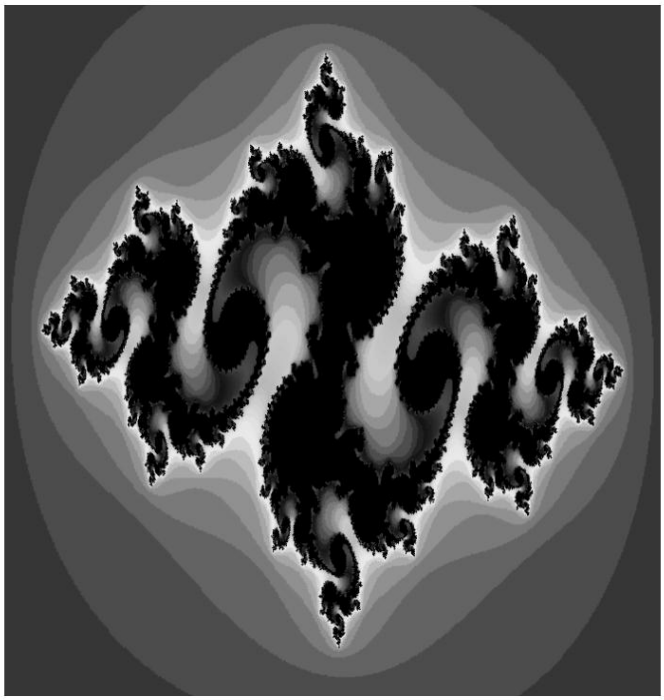

Fig. 8 Julia sets

C. Fractal structures used along the research – Burning Ship

$$B = \left\{ c \in \mathbb{C} | \lim_{n \to \infty} Z_n \neq \infty \right\} \tag{7}$$

$$Z_0 = c$$
$$Z_{n+1} = [|Re(Z_n)| + i|Im(Z_n)|]^2 + c \tag{8}$$

Fig. 9 Burning Ship

D. Fractal structures used along the research – Bird of Prey

$$P = \left\{ c \in \mathbb{C} | \lim_{n \to \infty} Z_n \neq \infty \right\} \tag{9}$$

$$Z_0 = c$$
$$Z_0 = c[|Re(Z_n)| + i|Im(Z_n)|]^3 + c \tag{10}$$

Fig. 10 Bird of Prey

E. Fractal structures used along the research – Water Plane

$$W = \left\{ c \in \mathbb{C} | \lim_{n \to \infty} Z_n \neq \infty \right\} \tag{11}$$

$$Z_0 = c$$

$$Z_{n+1} = Z_n^3 + Sin(Z_n) + c \tag{12}$$

Fig. 11 Water Plane

F. Fractal structures used along the research – 4th Degree Multibrot

$$D = \left\{ c \in \mathbb{C} | \lim_{n \to \infty} Z_n \neq \infty \right\} \tag{13}$$

$$Z_0 = c$$

$$Z_{n+1} = Z_n^4 + c \tag{14}$$

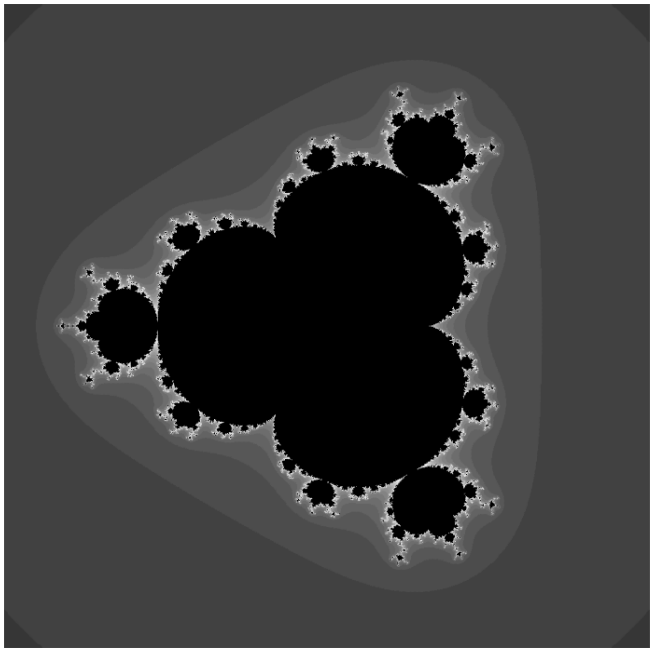

Fig. 12 4th Degree Multibrot

ACKNOWLEDGMENT

This article was supported by the European Regional Development Fund under the Project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089 and Internal grant agency (IGA/FAI/2012/019).

REFERENCES

[1] PILLER, I., Hashovací funkce a jejich využití při autentizaci, Vysoké učení technické v Brně, 2009
[2] TŘÍSKA, D., Kryptografická ochrana, Univerzita Tomáše Bati ve Zlíně, 2009.
[3] MANDELBROT, B. B., Fractals nad chaos: the Mandelbrot set and beyond. New York: Springer, 2004. 308 s. ISBN 0-387-20158-0.
[4] ZELINKA, I. Fraktální geometrie – principy a aplikace, BEN Praha 2006.
[5] The Mandelbrot set, available from: <http://warp.povusers.org/Mandelbrot/>

[6] LOFSTEDT, T. Fractal Geometry, Graph and Tree Constructions, Umea University, Sweden 2008.
[7] PIPER, F., MURPHY, S. Kryptografie – průvodce pro každého. 1. vyd. Praha:Dokořán, 2006. 157s. ISBN: 80-7363-074-5.
[8] BOSE, S. Information theory, coding and cryptography. Tata McGraw-Hill Education, 2008. 326s. ISBN: 0070669015.
[9] SPROTT, J. C., Chaos and time-series analysis aplikace. Oxford University Press, 2003. 507 s. ISBN:978-0198508403.
[10] STAIR, R. M., REYNOLDS, G. W. Principles of Information systems. 7. vyd. Course Technology, 2005. 808 s. ISBN-10: 9780619215613.
[11] MOTÝL, I., PÁLKA, J., JAŠEK, R.: Application of hash function to increase security level of the information system. In Int. 2010. Internet, bezpečnost a konkurenceschopnost organizací. Zlin 17-18. 3. 2010. ISBN 978-83-61645-16-0.
[12] *CAREY, J., M.* Human Factors in Information Systems: *The Relationship Between User Interface Design and Human Performance. 1. vyd. USA: Intellect Books, 1996. 254 s. ISBN: 9781567502862.*
[13] *FEIL, T., SINKOV, A.* Elementary Cryptanalysis. *2. vyd. USA: Michigan university, 2009. 226 s. ISBN: 9780883856475.*
[14] *GALBRAITH, S. Blockwise – Adaptive Chosen – Plaintext Attack and Online Modes of Encryption.* American Journal of Applied Sciences 4. *1st ed. Heidelberg: Springer, 2007, vol. 23 , p. 129-151.*
[15] KAHATE, A. Cryptography in the database. 2. vyd. New York: Tata McGraw-Hill Education, 2008. 792 s. ISBN: 9780070648234.
[16] KIZZA, J., M. Computer Network Security. 1. vyd. USA: Springer, 2005. 534 s. ISBN: 9780387204734.
[17] LESMOIR-GORDON, N., ROOD, W., EDNEY, R. Introducing Fractal Geometry. 3. vyd. United Kingdom: Icon Books, 2002. 176 s. ISBN: 9781840467130.
[18] LU, N. Fractal Imaging. 1. vyd. London: Academic Press, 1997. 412 s. ISBN: 0124580106.
[19] NAGEL, CH., EVJEN, B., GLYNN, J. Professional C# 2008. 1. vyd. USA: John Wiley & Sons, 2011. 1848 s. ISBN: 9781118059463.
[20] POUR, J. Informační systémy a technologie. 1. vyd. Praha: VSEM, 2006. 492 s. ISBN: 9788086730035.
[21] SCHNEIDER, B. Applied cryptography: Protocols, algorithms, and source code in C. 2. vyd. USA: Michigan university, 1996. 758 s. ISBN: 9780471128458.
[22] STAIR, R., M., REYNOLDS, G., REYNOLDS, G., W. Fundamentals of Information Systems. 5. vyd. USA: Cengage Learning, 2006. 457 s. ISBN: 9781423925811.
[23] SUTTON, R., J. Secure Communications: Applications and Management. 1. vyd. USA: J. Wiley & Sons, 2002. 322 s. ISBN: 9780471499046.
[24] WHITMAN, M., MATTORD, H., J. Principles of Information Security. 4. vyd. USA: Cengage Learning, 2011. 617 s. ISBN: 9781111138219.
[25] LONG, J., MITNICK, D. No Tech Hacking: A guide to social engineering, dumpster diving, and shoulder surfing. Syngress, 2008. 285 s. ISBN 1597492159.
[26] GODBOLE, N. Information systems security: Security management, metrics, frameworks and best practices. 1. vyd. India: Wiley India Pvt. Limited, 2008. 1020 s. ISBN: 9788126516926.

Ivo Motyl is a member of doctoral study program on the Department of Informatics and Artificial Intelligence, Faculty of Applied Informatics. He was born in Zlin, Czech Republic on 11.3.1984. He received his bachelor's degree in Faculty of Applied Informatics in Zlín in the year 2006. Later, he was awarded master's degree in the same place in the year 2008. He started doctoral study programe in 2008. He was on the short term attachment in the University in Vigo in 2010. He made degree examination in 2011. He has published 24 technical papers in international journals and conferences.