The Second Indonesian-Japanese Conference on
Knowledge Creation and Intelligent Computing (KCIC) 2013 *ISBN : 978-602-9494-68-6*

Comparative Analysis Spread Spectrum and Parity Coding Steganography in E-commerce

Mike Yuliana, Fatchul Bari Hikmawan, Brahim Wijaya and M. Zen Samsono Hadi
Division of Telecommunication Engineering, Dept. of Electrical Engineering,
Electronic Engineering Polytechnic Institute of Surabaya (EEPIS), Surabaya, Indonesia.
EEPIS Campus, Jalan Raya ITS Sukolilo, Surabaya 60111
Tel: +62(31) 594 7280; Fax: +62(31) 594 6114
Email : mieke@eepis-its.edu, chul.moveon@yahoo.com, joya_brahim@yahoo.com, zenhadi@eepis-its.edu

Abstract

The transaction data online has increased compared to the previous communications that mostly in the form of voice and text messaging. To improve the security, data must be protected such a way that it cannot be attacked by unauthorized parties. In this case, a good security system must be able to transmit the original information to the second party without having to know the existence and validity by a third party. One of the security systems that can be used is steganography. In this paper, we will compare the performance of Spread Spectrum and Parity Coding in e-commerce based on Android in case of processing time between insertion and retrieval information, and the changing image size during the insertion process. Our experimental results show that parity coding has better performance on client side that use low performance smart phone based on Android operating system and spread spectrum has better performance on blackberry store server that use laptop PC .

Keywords: steganography, spread spectrum, parity coding, insertion.

1. Introduction

Electronic commerce, commonly known as e-commerce, is the buying and selling of product or service over electronic systems such as the Internet and other computer networks. Electronic commerce draws on such technologies as electronic funds transfer, supply chain management, Internet marketing, online transaction processing, electronic data interchange (EDI), inventory management systems, and automated data collection systems. Modern electronic commerce typically uses the World Wide Web at least at one point in the transaction's life-cycle, although it may encompass a wider range of technologies such as e-mail, mobile devices social media, and telephones as well [3].

The transaction data online has increased compared to the previous communications that mostly in the form of voice and text messaging. Various information even more accessible, both upload and download, which includes data that require or do not require a high level of privacy. To improve the security of data access privacy, it would require a system that is not easy to determine the content of the information by third parties that are not desirable. In this case, a good security system must be able to transmit the original information to the second party without having to know the existence and validity by a third party. One of the security systems that can be used is steganography.

Steganography is the science and art of hiding messages in other media so that the existence of the first message is unknown[7]. Steganography comes from the word "Steganos" meaning to hide, and "grapien" which means writing, then Steganography is a technique of data hiding important information into other media file such as image, audio, video or document file. Steganography is different from cryptography. Cryptography secret meaning of the message while the existence of the message persists [1][2], whereas steganography cover the presence of message. Steganography can be seen as a continuation of Cryptography. In practice, the message encrypted and then hidden into other media file so that third parties are not aware the existence of the message.

There are various methods in performing steganography techniques, among others: LSB, spread spectrum and parity coding. Spread spectrum method has been done in previous research using audio files as cover object [6]. How to insert a secret message into a cover-object also vary, some using watermarking method, as FFT (Fast Fourier Transform) and the IFFT (Inverse Fast Fourier Transform) [5] and others using LSB (Least Significant Bit) . Several research that used Parity Coding and Spread Spectrum show that the processing time for embedding and extracting are quite fast [5], that's why this two methods will be used in our research, because on the client side we use android smart phone that has low performance.

In this paper, we will compare the performance of Spread Spectrum and Parity Coding in e-commerce based on Android in case of processing time between insertion and retrieval information, and the changing image size

during the insertion process. The experimental results will indicate which method has better performance on the client and blackberry store server side.

2. Steganography

Steganography is the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message, a form of security through obscurity. The word *steganography* is of Greek origin and means "concealed writing" from the Greek words *steganos* meaning "covered or protected", and *graphei* meaning "writing". Generally, messages will appear to be something else: images, articles, shopping lists, or some other *covertext* and, classically, the hidden message may be in invisible ink between the visible lines of a private letter[7].

Although steganography and cryptography are related there are some differences between them. Steganography hides a message within another message and tries to offer the impression that nothing but normal text, audio or video message is being exchanged. In cryptography the message is encrypted, it looks like a meaningless jumble of characters. Therefore, whereas cryptography protects the contents of a message, steganography can be said to protect both messages and communicating parties.

Steganography includes the concealment of information within computer files. In digital steganography, electronic communications may include steganographic coding inside of a transport layer, such as a document file, image file, program or protocol. Media files are ideal for steganographic transmission because of their large size. As a simple example, a sender might start with an innocuous image file and adjust the color of every 100th pixel to correspond to a letter in the alphabet, a change so subtle that someone not specifically looking for it is unlikely to notice it.

2.1 Spread Spectrum

Spread Spectrum method is divided into two section, where the first section is embedding (insertion) data information to cover image which can be seen in Fig. 1 and the second section is the extraction (removal) from stego-image which can be seen in Fig. 2. In the section, there are phases of generating PRN (pseudorandom number) using the LCG (Linear Congruential Generator) to make the semi-random numbers [6].

LCG is defined in (1) as:

$$X_n = (AX_n - 1 + B) \bmod M \qquad (1)$$

Where Xn is the random number n, Xn-1 is a random number before, A is the multiplier factor, B is the increment, and M is the modulus. The biggest LCG period is M and in most cases the period is less than M. Random number generated by the LCG will be repeated with the same value at period-M, but it is possible series of random numbers generated not more than the modulo value. LCG will have the full period if fulfill the following requirements:

1. B is relatively prime to M
2. A-1 can be divided by all the prime factors of M
3. A-1 multiples of 4 if M multiplied by 4
4. M value is greater than max (A, B, X0)
5. $A > 0$ and $B > 0$

The main steps in the process of embedding information into the cover image are as follows:

(a) Generating pseudorandom number row using a key

$$P = A_{(k)} \qquad (2)$$

Where P is the result of generation LCG in the form of binary bits and k is the key, while $A_{(k)}$ is the result of generation LCG in the form of array bytes.

(b) XOR process between spreaded message with the previous generated pseudorandom number, resulting noise ,

$$Q = P \oplus m \qquad (3)$$

Q is a form of binary bits pseudo noise and m is the message that has been converted into binary bits.

(c) Adding noise into the cover image using LSB method so it turned into stego-image (n). The process of adding noise includes two steps:
   a. Taking the color components of image in (4).
   b. Pseudo noise insertion into the LSB of each color component in (5).

$$B_{(r,g,b)} = I_{(m,n)} >> \{16_{(r)}, 8_{(g)}, 0_{(b)}\} \qquad (4)$$

$B_{(r,g,b)}$ is byte array that contains all the color component values in each pixel, *r* is red, *g* is green, *b* is blue, *m* is the length of image, *n* is the width of image, $I_{(m,n)}$ is image that has dimensions *m* x *n*, and $\{16_{(r)}, 8_{(g)}, 0_{(b)}\}$ is shifted by 16 bits for red, 8 bits for green and does not shift to blue.

$$E = Q \oplus B_{(r,g,b)} \qquad (5)$$

E is byte array in the form of color components that have inserted messages, Q is pseudo noise bits, and $B_{(r,g,b)}$ is byte array that contains all the values of each pixel color components.

The steps in the process of retrieval information are as follows:

1. Stego-image processing, taking bits LSB from all color components, resulting series of bits LSB like noise

2. Generating the same pseudorandom number, with the same key from insertion process.
3. XOR processing between noise that has been taken from step (1) with pseudorandom number that has been generated from step (2) to generate the original message back through despreading process.

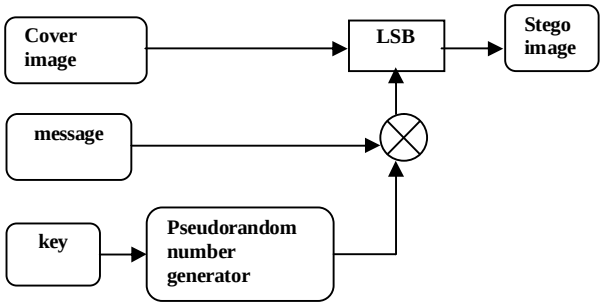

Fig 1. Flow diagram insertion (embedding) process

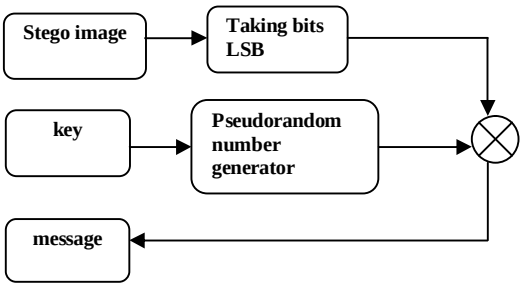

Fig 2. Flow diagram retrieval information process

2.2 Parity Coding

Steganography technique using parity coding is the process of counting bits with even parity conditions [4]. The results of the calculation are checked, if bit 1 an odd number, then the value of parity bit is 1. If bit 1 an even number, then the value of parity bit are 0 [7]. This method is divided into two sections, where the first section is embedding (insertion) data information to cover image which can be seen in Fig. 3 and the second section is the extraction (removal) from stego-image which can be seen in Fig. 4.

The main steps in the process of embedding information into the cover image are as follows:

a) Message file and the cover image are converted to binary form.
b) Cover-image are sorted and counted according RGB with even parity, so the result can be combined with the message file.
c) If the sum of parity bit is not equal to one bit of the message then it needs to be changed the value of the LSB from RGB (if 1 is replaced by 0, and vice versa), the insertion can be done because both have the same bits, and produce stego-image.

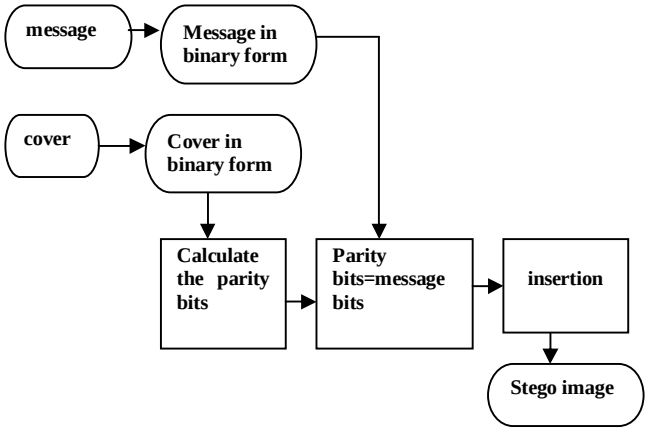

Fig 3. Flow diagram insertion (embedding) process

The steps in the process of retrieval information are as follows:

a) Choose stego-image.
b) Stego-images are classified according to the arrangement RGB.
c) After classification of RGB, recounting every bit RGB with even parity, and the results are the value of the secret message.

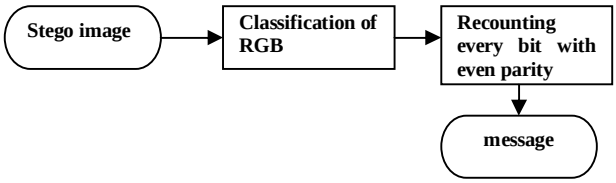

Fig 4. Flow diagram retrieval information process

3. Implementation

In this section, we will explain the whole scheme of the system. The explanation of the workflow system in Fig. 5 is as follows:

(1) From the client side, there is an Android application that display list of products, handle and store member registration, and processing steganography for securing credit card data (number and password).
(2) Overall data in the form of total expenditure and stego-image is sent to the server via the provider of each user.
(3) On the server, stego image is processed to remove confidential information such as credit card number and password. Then the detail data of credit cards are stored in database.
(4) If the connection success, the success expenditure confirmation is sent back to the user.

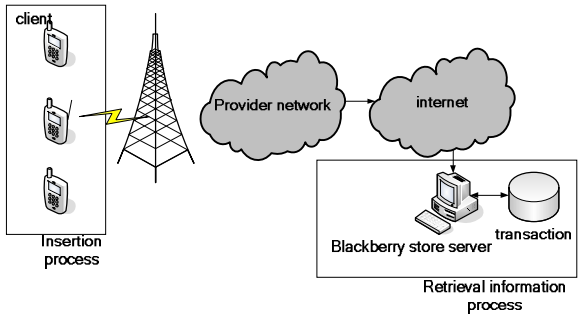

Fig 5. System Integration

Fig. 6 shows the overall system that made in the form of two-column chart. It show that the user insert credit card number and password into the cover image using key variable and generate an image E. Then on the server side retrieve information from the image E using the same key and the result is D that contains credit card number and password. BlackBerry Store is a web application based on JSP (Java Server Pages) containing BlackBerry products. The communication between client and BlackBerry Store is a sample of e-commerce transaction.

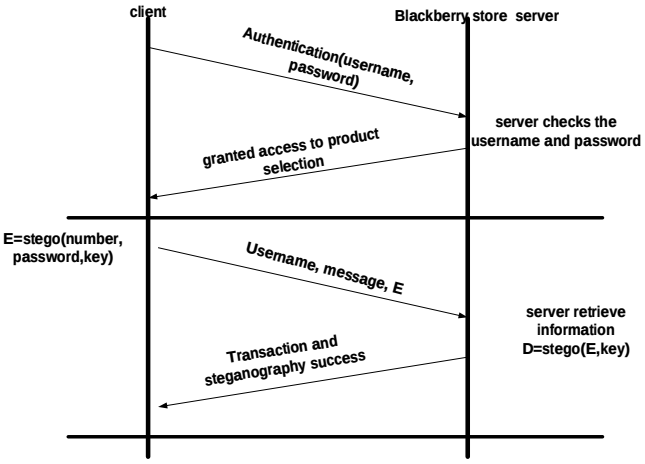

Fig 6. Workflow system

4. Experimental Results

Laptop PC specifications that used as Blackberry Store server are:

- TOSHIBA Satellite M505-S4940
- Processor Intel Pentium Dual-Core T4200 2.00GHz
- Memory DDR2-SDRAM 2GB
- LAN on-board Realtek PCIe FE Family
- Wireless LAN on-board Realtek RTL8192E

While the smart phone device specifications that used as client are:

- Samsung Galaxy Gio GT-S5660
- OS Android 2.3.3 Gingerbread
- Chipset Qualcom QCTMSM7227-1 Turbo
- CPU 800Mhz
- Display TFT Capacitive 3,2 inches, 16M colors, 320x480 *pixel*s (~180ppi density)
- Memory 158MB internal storage, 278MB RAM
- Size 110,5 x 57,5 x 12,2 mm

Software specifications that used in Blackberry Store server are:

- Apache Tomcat 7.0.21
- JAVA Run time Environment 1.7 build 05
- MySQL 5.0.8
- MySQL Connector JAVA 5.1.19

While software specifications that used as client are:

- OS Android 2.3.6 Gingerbread
- Baseband S5660DXKT8
- Kernel 2.6.35.7-px-based-kernel@Phiexz #20120421

The results obtained from this research are computation time of insertion and retrieval information process, and the changing image size during the insertion process.

4.1 Computation Time of Insertion and Retrieval Information Process

Insertion process is done when the payment is made by the user in the client side. There are several steps in the spread spectrum insertion process among others:

1. Spreading information
2. Generate PN bits with key “fatchul”
3. Modulation message and LCG
4. XOR process of each LSB image

Whereas several steps in the parity coding insertion process among others:

1. Converting secret message
2. Capturing RGB values of each pixel
3. Parity coding calculation on cover image
4. Matching parity coding with a secret message

Retrieval Information process occurred on the Blackberry store server side. There are several steps in the spread spectrum retrieval information process among others:

1. Capture all of LSB bits
2. Generate PN bits with key “fatchul”
3. Demodulation LSB and LCG
4. Despreading bits

Whereas several steps in the parity coding retrieval information process among others:

1. Classification of RGB
2. Recounting every bit RGB

 ISBN : 978-602-9494-68-6

In this research, processing time between insertion and retrieval information measured by:

1. Giving command to measure computing time between statements in the program that will be measured.
2. Saving the computation time
3. Repeat step 1 and 2 for ten times, and average the result

The test results in Fig 7 and 8 show that on the spread spectrum insertion section, XOR process of each LSB image takes the longest time because in this step there are pseudo noise insertion process into the pixel image, where the extract process done one by one from pixel (0,0) until the last pixel. So, automatically the looping process takes a long time. While on the spread spectrum retrieval information section, Despreading takes the longest time because in this process there is despreading process four times to get information bit. Previously information bit also have spreading four times. In addition there is process of converting despreading result from bytes to String.

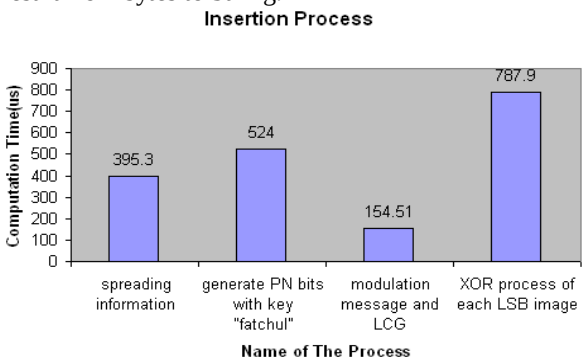

Fig 7. Computation time on the spread spectrum insertion process

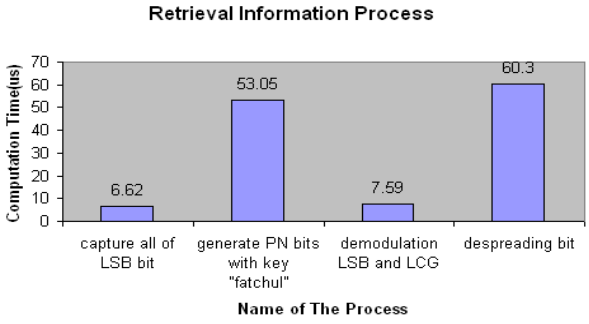

Fig 8. Computation time on the spread spectrum retrieval information process

The test results in Fig 9 and 10 show that on the parity coding insertion section, capturing RGB values of each pixel takes longer than the others because there is process separation of basic colors Red Green and Blue from cover image. Separation process is done one by one from row 0 column 0 to row n column n, and the computation time is 524.8 µs. While on the parity coding retrieval information section, recounting every bit RGB with even parity takes the longest time because there is process collecting from parity coding result into 8 bit (1 byte) and the next process is changing from byte to String, with the computation time is 186,3 µs.

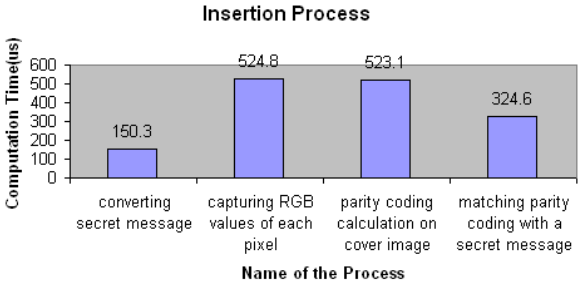

Fig 9. Computation time on the parity coding insertion process

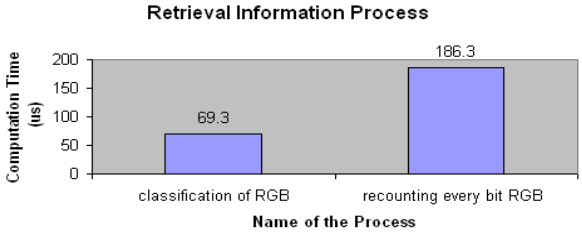

Fig 10. Computation time on the parity coding retrieval information process

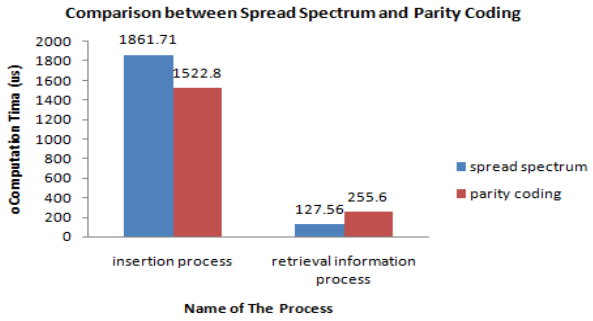

Fig 11. Comparison of computation time between spread spectrum and parity coding

The test results in Fig 11 show that the spread spectrum insertion process that occurs on the client side takes longer time than the other, and the computation time is 1861.71 µs. It happen because XOR process of each LSB image takes the longest time instead of the other process in spread spectrum and parity coding insertion process. Whereas parity coding information retrieval process that occurs on the server side takes longer time than the other, and the computation time is 255.6 µs. Process collecting from parity coding result into 8 bit in

recounting every bit RGB process has different significant time with other process in spread spectrum and parity coding retrieval information process.

4.2 Testing on Several Types of Images

Purpose of this test is to determine the final size of the image file and the average computing time insertion on the different type of image as shown in Fig 12. The test results in Table 1 showed that all of types of images are change, the change is decrease and increase in the final file size. By using spread spectrum, image size on the end result tends to be greater than Parity Coding. The largest final size is type of image BMP and the smallest is type of image JPG. For computation time, spread spectrum

average computation time is faster than Parity Coding.

Fig 12. (a) Cover image (b) Stego Image

Table 1. Changing Image Size

Name of Cover Image	Type of Image	Initial size	Final Size		Change of size(%)		Computing time	
			Spread spectrum	Parity coding	Spread spectrum	Parity coding	Spread spectrum	Parity coding
cross	bmp	1.194	933	786	-21.9	-34.2	112.89	146.13
fb	jpg	700	743	618	+6.14	-11.7	91.21	141.29
help	png	3.336	839	711	-74.85	-78.7	96.29	132.26

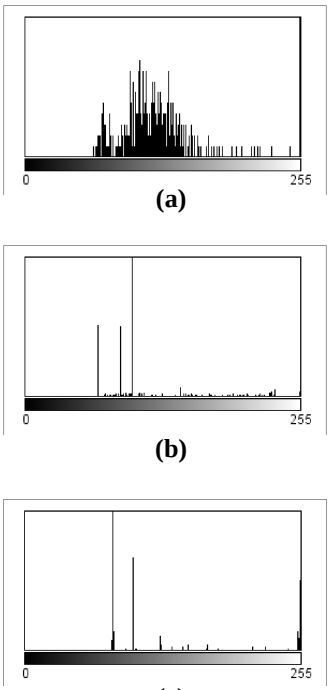

Fig 13. *Histogram*(a)cross.bmp (b)fb.jpg (c)help.png

Computational time testing shows that there is significant time difference between three types of images because there are different colors of histogram composition. Cross.bmp has the most extensive distribution of color then it requires the longest computation time. Fb.jpg has the least extensive distribution of color then it requires the fastest computation time. Fig 13 Show that the distribution of color image affects the computing time need, the wider distribution of the color the longer computation time required.

5. Conclusion

In this paper we have presented an implementation of spread spectrum and parity coding steganography in E-commerce. The experimental results showed that the spread spectrum insertion process that occurs on the client side takes longer time than the other, and the computation time is 1861.71 µs, while parity coding information retrieval process that occurs on the server side takes longer time than the other, and the computation time is 255.6 µs. Image final size on the end result with spread spectrum tends to be greater than Parity Coding, but spread spectrum average computation time is faster than Parity Coding. But overall, parity coding has better performance and better suited implemented on low performance smart phone.

References

[1] A. Cheddad, J. Condell, K. Curran, P. Mc Kevitt, "Securing information content using new encryption method and steganography", *Proceedings of the Third IEEE International Conference on Digital Information Management*, University of East London, UK, 13–16 November 2008, pp. 563–568.

[2] Bruice Schneier, *Applied Cryptography Protocols, Algorithm and Source Code in C. Second edition*, Wiley India edition 2007.

[3] Chaudhury, Abijit and Jean-Pierre Kuilboer, *e-Business and e-Commerce Infrastructure*, McGraw-Hill, ISBN 0-07-247875-6, 2002.

[4] Ch.rupa, P.S Avadhani, E. SrinivasReddy, "An Efficient Security Approach using PGE and Parity Coding", *International Journal of Distributed and Parallel Systems (IJDPS)* Vol.3, No.6, November 2012.

[5] Ingemar J. Cox, Joe Kilian, Tom Leighton dan Talal Shamoon, "Secure Spread Spectrum Watermarking for Multimedia", *IEEE Trans. On Image Processing*, 6, 12, 1663-1687, 1997.

[6] Stefan Katzenbeisser and Fabien A.P.Petitcolas. *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House, Boston, London , 2000.

[7] Venkatraman, S., Abraham, A. and Paprzycki, M., "Significance of Steganography on Data Security", *Proceedings of the International Conference on Information Technology: Coding and Computing*, 2004.