A Technical Reflection towards Understanding the Future of Data Storage Appliances: How to Store and Retrieve Data without Storage Appliance?!

Adel M. Agina [1], Salah M. Talha[2], Saad M. Alhossin [3] and Aseel I. Ageena [4]

[1] University of Twente/Faculty of Electrical Engineering, Mathematics and Computer Science
(Human Media Interaction) - P. O. Box 217, 7500 AE, Enschede, The Netherlands
[2] Database Consultant at the Company of MT5IT Ltd., & Head of IT Dept. at RFID-Company Ltd., Tripoli/Libya
[3]College of Electronic and Computer Science & Head of RFID-Company Ltd., Tripoli/Libya
[4]Community Revival UK (Further Education College)

Abstract

The present study was conducted to introduce a new technique for storing and retrieving data without using any storage appliance/device, any of the common compressed/zipped or decompressed/unzipped software application as no database environment was used to store or retrieve data. The outcomes showed that the mechanism of the new technique AMA-TECH is able to store the entire given data in one single code and retrieves that data, as stored, from that code itself. AMA-TECH was able to manipulate all data types at once (image, text, or mixture data). What makes AMA-TECH different was that the generated code was not an index to such a record in a computer database's table or a name of a compressed file. Technically, the code, per se, becames storage appliance to store and retrieve data as stored. Accordingly, the measurement of data size was not by Kilobyte, Megabyte, and so on. Instead, data was measured by how many nanoseconds, seconds, or minutes that AMA-TECH needs/consumed to store and retrieve data in-and-from the code itself. To our knowledge so far, this solution has never introduced before. However, many and many other issues were raised that still we are working on them.

Keywords: *Compressed Software Application, Data Storage Appliance, Database Environments, Data Size's New Measurement*

1. Introduction

Nowadays, database environments (e.g., SQL, and ORACLE) and the storage appliances (e.g., hard-disks, flash memories, CDs, DVDs, tapes, etc.) still are the backbone to store and re-trieve data. Accordingly, can we think about such a system, which acts without storage appliance to store and retrieve any given data? Undoubtedly, this question sounds like magic, in-sensible, unbelievable, and simultaneously very appealing and interesting. The contribution to shade the first light to answer that question is what the present study is undertaking.

A. *The Need for Storage Appliances*

New kinds of data and their analysis increase the demand for flexible data models supporting data of different degrees of structure [1]. As our life continues, the world is rapidly running to develop new and more sophisticated storage appliances with software applications for data encryption, decryption, com-pressed, and decompressed to be stored and retrieved perfectly and securely with secrecy. Recently, there is a substantial influx in the enterprise storage market [2]. For instance, large-scale storage appliances like Teradata's Extreme Performance Appli-ance [3] and Oracle's Exadata [4] harness the technology of SSDs to realize very high I/O operations per second (IOPS). Moreover, there are a host of positive forecasts for SSDs in the enterprise market [5]. Therefore, as an inevitable demand to protect data, a huge number of security software applications and storage de-vices are currently researched, tested, and used to secure data because of many difference and various problems such as dam-aging, crashing, burning, stealing, hacking, penetration, and so forth- though, the latter issues still pose a greater threat to stored information. To date, there is exists no hack-proof solu-tion against those problems rather than obscurity encryption data, protecting data with passwords, and keeping many back-ups on many different storage appliances. Technically, there is still no way to avoid using storage appliances to store and re-trieve data so far.

2. Rational of the Present Study

The Problem Underlying Core

The most significant problem that faces the innovative and burgeoning technologies is the size of data that should be stored in a limited storage appliance (e.g., storing information of the market's goods in barcodes and information of the objects on small memory chips like RFID's tags [6]). Another big problem faces the technology nowadays is the security concerning tracking and tracing objects where the code that saved in the object's small chip memory is actually an index-code to a computer database center to get more detailed information about that object [7]. Up to date, the use of databases and storage appliances still are the main components of any system all over the world [8].

THE RELEVANCE OF THE CURRENT STUDY PROBLEM

The use of storage appliances is eventually dangerous and insecure. This is extremely true given the simple fact that any data can be accessed through manipulation of the system whether it could be human error or a technical flaw within. Simply, accessing any storage appliance can be done by the professional programmers, betrayer and intruder employees, may happen accidentally (e.g., any of the natural disasters), or coincidentally (unintentionally deletion or overwrite... etc.). Specifically, the main problem of the storage appliances can be seen through it is inevitable use to store and retrieve data. Thus, the dangerous remains as long as the storage appliances still in use for whatever the security techniques used for data protection. This, in turn, requires the security level to be always updated and upgraded for the associated software and hardware respectively. In sum, whatever the security level available in any storage appliance, the system remains insecure, friable, and penetrable.

WHY THE PRESENT STUDY SHOULD RAISES?

Every year there is a huge financial support that has to be spent to develop more software applications, creating higher storage appliances, developing advanced communications, and managing the database environments. Most of those costs went to the security developments especially for protecting data and the associated storage appliances. Despite the present study is just a reflection, the outcomes were very motivated and promising that the future of the storage appliances might be changed and the world might save those millions. In other words, it is the time to think about new ways to manipulate data digitally.

I. Critical Critiques on Data Compression Techniques

In the late 1970s and early 1980s, the idea of executing database operations in storage devices was extensively explored in the context of database machines [9]. The idea was shown to have the potential of achieving great performance improvements for simple operations (such as “scan”) with processor per head, track, or disk. As summarized in [9], however, it has failed because of several difficulties. First, most database machines used special-purpose hardware (such as associative disks and magnetic bubble memory), but the performance gains were not enough to justify the additional cost and design time. Second, improvements in host-side processor architecture were much faster than disk bandwidth, leading to the underutilization of the special-purpose hardware. Finally, there was little performance gain for complex operations like join. Nowadays, the common process used to process any huge data is regularly done through three mains steps. First, convert the given data into an image format. Second, save the image's pixels as a file using binary base or any other base and then compress the file in a computer database (i.e., save the compressed-data using one of the common database environments such as SQL and ORACLE). Third, generate a unique identifier (i.e., an index-code) and use it as a primary key to retrieve the related data from computer database. For instance, a human fingerprint can be scanned, converted into pixels, and then saved in the tag's memory as an index-code to the rest of the related data in computer database [6]. Nevertheless, the storage appliances, database environments, and compressed software applications are still inevitable to store and retrieve data

A. Critiques on Data Hiding Techniques

With the rapid development in multimedia and the Internet, a huge amount of digital information, i.e. digital images, videos, audio, and text, is transmitted and stored and made accessible on the Internet, resulting in several security issues, e.g., interception, duplication, and modification [10]. As a result, ensuring the security and secrecy of transmitted information over the Internet has become increasingly critical. Nowadays, data hiding (obscurity) is an essential technique that hides secret data into cover images with minimal distortion. This means that the embedded image has close visual quality to that of the cover image to avoid attracting the attention of malicious attackers. Data hiding can be divided into two categories, namely, irreversible [11]-[13] and reversible data hiding [14]-[30] where each category has many advantage and disadvantages. Recently, to save the storage and bandwidth space of the embedded image, many data hiding schemes for compressed images have been proposed in the literature. The main reason is that the sizes of compressed images are much smaller than the sizes of the original images before and after data embedding. Various compression techniques, i.e., JPEG [24] and [25], JPEG 2000 [1], block truncation coding (BTC) [26], and vector quantization (VQ) [10]–[23] have been applied for embedding data to achieve both a good compression rate and good embedding capacity. Despite all those techniques, the compressed data still have a size in Kb, Mb … etc., and have to be stored and retrieved in and from one of the storage appliances with such protection applications!

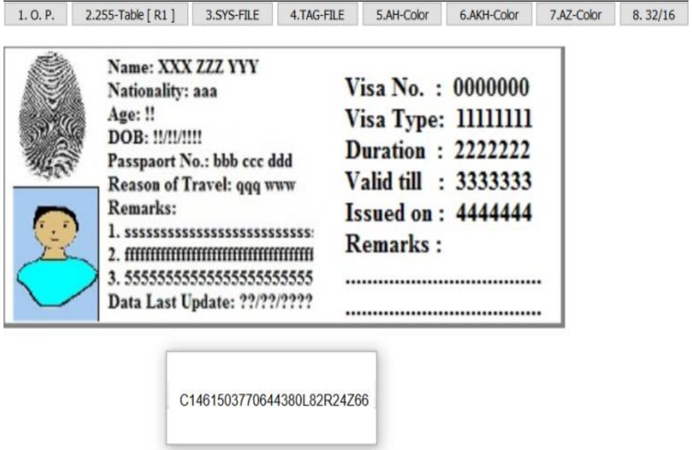
II. AMA-TECH

AMA-TECH is a technique implemented based on the power of math and computer programming. The initial outcomes of AMA-TECH were very motivated and highly promising for brightness future of storing and retrieving data and the associated storage appliances. Specifically, AMA-TECH is a technique works without using any of computer database environments, no storage appliances, and no compressed/decompressed software applications were used to store and retrieve data! Practically, AME-TECH stores the entire given data just in one single code, which can also be used to retrieve the same data as it were stored. This, in turn, means that the current common data measurements (i.e., Kilobyte, Megabyte and so on) will have no meaning to describe the size of data with AMA-TECH.

III. THE EXPERIMENT

A. *Store the Given Data without Data Storage Device*

AMA-TECH-01 manipulated the given data (image, text, or mixture data) to be stored into one single code without using any of the compressed/zipped software applications as shown in Figure (1).



Name: XXX ZZZ YYY
Nationality: aaa
Age: !!
DOB: !!/!!/!!!!
Passpaort No.: bbb ccc ddd
Reason of Travel: qqq www
Remarks:
1. sssssssssssssssssssssssss
2. ffffffffffffffffffffffffffffffffffff
3. 5555555555555555555555
Data Last Update: ??/??/????

Visa No. : 0000000
Visa Type: 11111111
Duration : 2222222
Valid till : 3333333
Issued on : 4444444
Remarks :
....
....

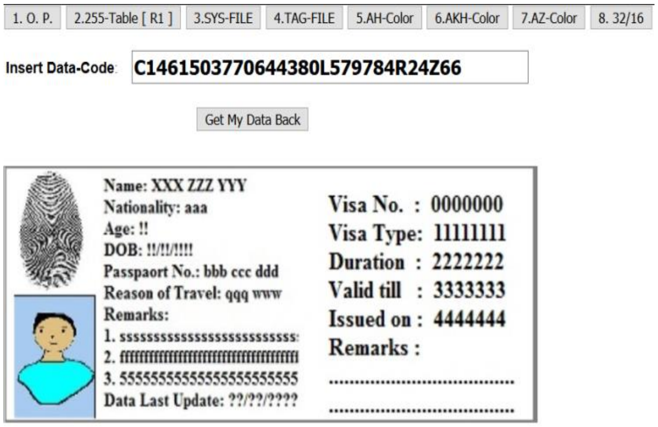
C1461503770644380L82R24Z66

Fig. 1. AMA-TECH-01 for manipulating data into one single code without using any of computer database environments, storage appliances, or compressed software applications

AMA-TECH-01 did not need any storage appliance to store the code (you can keep it in mind or on a piece of paper). The experiment was running on a card that may represent a passport, travelling visa/ticket, personal document, student card, or any other document. The proposed card was assumed to hold personal fingerprint, picture, information, and others as shown in Figure (1).

B. *Retrieve the Original Data from the Code*

Only AMA-TECH-02 was running on another computer, which has never 'seen' the original data before as shown in Figure (2).



Insert Data-Code: C1461503770644380L579784R24Z66

Name: XXX ZZZ YYY
Nationality: aaa
Age: !!
DOB: !!/!!/!!!!
Passpaort No.: bbb ccc ddd
Reason of Travel: qqq www
Remarks:
1. sssssssssssssssssssssssss
2. ffffffffffffffffffffffffffffffffffff
3. 5555555555555555555555
Data Last Update: ??/??/????

Visa No. : 0000000
Visa Type: 11111111
Duration : 2222222
Valid till : 3333333
Issued on : 4444444
Remarks :
....
....

Fig. 2. AMA-TECH-02 for retrieving the original given data from the single code without using any of computer database environments, storage appliances, or decompressed software applications

Just by inserting the generated code (Figure 1) and clicking the option "Get My Data Back", AMA-TECH-02 was able to retrieve the original data back. Technically, AMA-TECH-02 was able to retrieve the original data from the given code itself without any decompressed software application or database environment as no data storage appliance was used with the second computer, which has never seen the original data before.

C. *AMA-TECH Security and Secrecy*

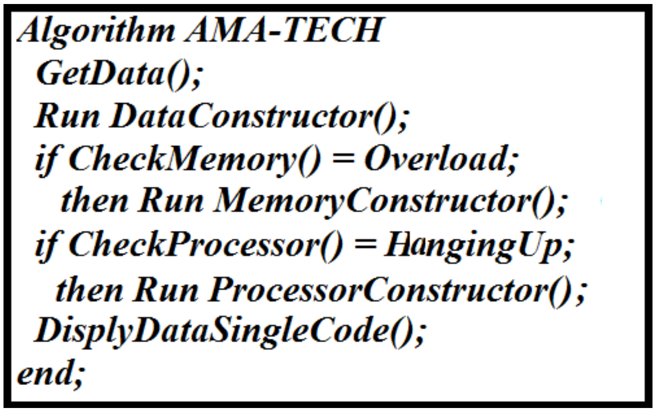
AMA-TECH is, by nature, secure and safe. This is very clear through manipulating the given data into just one single code without using any of the computer database environments, any of the storage appliances, and without using any of the compressed/decompressed software applications. The generated code can also be encrypted many times based on the user's desire, which can be seen as the second level of AMA-TECH security. Thus, there will be no need to use a password to protect the given data as no need to be scared if your code becomes known. Eventually, the given data can be retrieved at any time just by inserting the original code into AMA-TECH-02 (Figure 2) using any computer. As the third level of security, you can keep the code out of the computer so no one will be able to know it.

D. *Measuring the Size of Data by AMA-TECH*

Given that AMA-TECH does not pay attention to the size of any given data (KB, MB, GB …etc.) to start the process, the time that AMA-TECH needed/consumed to manipulate the given data into one single code was the actual size of data. Thus, the new data size is the number of the *nanoseconds*, *seconds* or *minutes* that AMA-TECH consumed. For instant, the data in Figure (1) was exactly 280KB and became "*Less Than One-Second-AMA-TECH*" after generating the code (Figure 2). Remarkably, the time needed to store data in one single code was the same time to retrieve it as stored. However, this cannot be generalized yet as AMA-TECH still missing the ability to determine the exact data size in time after the manipulation process (we are currently working to add this procedure to calculate the exact time in nanoseconds).

IV. AMA-TECH Mechanism

In terms of programming, AMA-TECH was implemented to solve two main problems: Computer's Memory Overloading (CMO) and Computer's Processor Hanging-up (CPH). The two problems (CMO and CPH) were entirely solved using the power of math and computer programming. The solution of CMO and CPH are still under the 'rights reserved', so none of them will be stated in details as a simple algorithm can be displayed as shown in Figure (3).

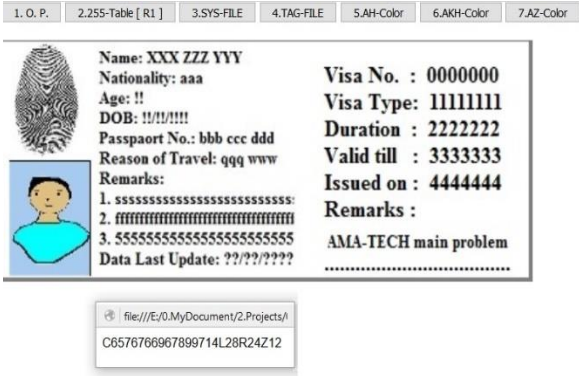


```
Algorithm AMA-TECH
  GetData();
  Run DataConstructor();
  if CheckMemory() = Overload;
    then Run MemoryConstructor();
  if CheckProcessor() = HangingUp;
    then Run ProcessorConstructor();
  DisplyDataSingleCode();
end;
```

Fig. 3. The mechanism of AMA-TECH shows the two main problems (CMO and CPH).

V. AMA-TECH Main Problem

AMA-TECH is currently associated with one problem that still we do not know yet whether it is an advantage or disadvantage as shown in Figure (4).



Name: XXX ZZZ YYY
Nationality: aaa
Age: !!
DOB: !!/!!/!!!!
Passpaort No.: bbb ccc ddd
Reason of Travel: qqq www
Remarks:
1. sssssssssssssssssssssssss
2. ffffffffffffffffffffffffffffffffffff
3. 5555555555555555555555
Data Last Update: ??/??/????

Visa No. : 0000000
Visa Type: 11111111
Duration : 2222222
Valid till : 3333333
Issued on : 4444444
Remarks :
AMA-TECH main problem
....

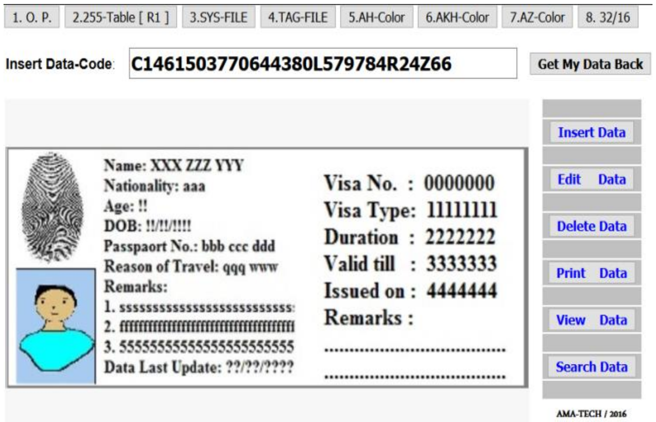
file:///E:/0.MyDocument/2.Projects/
C6576766967899714L28R24Z12

Fig. 4. AMA-TECH main problem that could be an advantage as it could be a disadvantage.

Precisely, after each Editing operation on the given data, AMA-TECH generated another new code with no overwrite operation on the first code (see Figure 4 where some data were added at the field of 'Remarks'). On one hand, this could be a problem because you need to keep the new code each time you make any editing operation. On the other hand, this could be more secure because you do not need to keep the first code (you may simply ignore it) so the security level of your data is always renewed. This issue is currently one of the main considerations of our running investigation and study.

VI. Current and Future Work

Given the fact that a greater data capacity requires larger size of storage appliances that, in turn, leads inevitably to more expensive equipments, our current consideration is the following question: how can the generated code be able to hold a complete data program management to control the main data operations (insert data, delete data, edit data, view data, search data, print data...etc.) as shown in Figure (5).



Insert Data-Code: C1461503770644380L579784R24Z66

Name: XXX ZZZ YYY
Nationality: aaa
Age: !!
DOB: !!/!!/!!!!
Passpaort No.: bbb ccc ddd
Reason of Travel: qqq www
Remarks:
1. sssssssssssssssssssssssss
2. ffffffffffffffffffffffffffffffffffff
3. 5555555555555555555555
Data Last Update: ??/??/????

Visa No. : 0000000
Visa Type: 11111111
Duration : 2222222
Valid till : 3333333
Issued on : 4444444
Remarks :
....
....

Fig. 5. Current work: AMA-TECH as a complete data management system.

In other words, when inserting the generated code into AMA-TECH-02, suppose that AMA-TECH shows such a system with the main data operations to manage the presented data on screen. This, in turn, leads to present the following big question: can data be indexed as regular databases inside the code itself? Specifically, if there are many cards (i.e., different data records), can AMA-TECH -02 store and retrieve all of them based on the user's request? Figure (5) is our current work besides investigating the possibility of manipulating video data.

VII. Conclusion

Given the massive body of the research and techniques regarding data manipulation (storing, retrieving and compression, …), and the rapid development of the sophisticated storage appliances, it might be quite hard to believe that data (image, text, or mixture) can be stored in one single code and retrieved as stored from that code using any computer and without using any database environment, compressed/decompressed software application, or any data storage appliance. Accordingly, many questions are currently rising such as: *how and where data had been stored? How can data be retrieved by any other computer just by entering a single one code? Is it true that the code that I can simply keep in mind holds all my data?* All we can say at this moment: nothing is impossible with research as nothing is impossible with the power of math and computer programming as we are currently paying a solid attention to how can video data be manipulated in just one single code.

acknowledgment

The team work in the present project would like to thank the following companies: Mt5IT and RFID-Libya Ltd for their financial support.

REFERENCES

[1] E. Vasilyeva, M. Thiele, C. Bornhövd and W. Lehner, “Answering “Why Empty?” and “Why So Many?” queries in graph databases,” Journal of Computer and System Sciences, 82 (2016) 3–22.

[2] “Intel, Solid-state drives in the enterprise: a proof of concept,” White Paper, 2009.

[3] Teradata, “Teradata active enterprise data warehouse,” http://www.teradata.com/Active-Enterprise-Data-Warehouse/

[4] Oracle, “Oracle exadata,” White Paper, 2010.

[5] “Objective Analysis, Solid State Drives in the Enterprise, Technical Report”, 2008.

[6] D. Henrici, A. Kabzeva, T. Fleuren, and P. Müller, “Data Storage in RFID Systems,” University of Kaiserslautern Germany (Radio Frequency Identification Fundamentals and Applications, Bringing Research to Practice, Book edited by: Cristina Turcu, ISBN 978-953-7619-73-2, INTECH, Croatia, pp. 278, February 2010.

[7] C. M. Roberts, “Radio frequency identification (RFID),” Computers & Security, vol. 25, no. 1, pp. 18-26, 2006.

[8] K. Sungchan, O. Hyunok, P. Chanik, C. Sangyeun, L. Sang-Won and M. Bongki, “In-storage processing of database scans and joins” Information Sciences, vol. 327, pp. 183–200, 2016.

[9] . Boral, D. J. Dewitt, “Database machines: an idea whose time passed? A critique of the future of database machines,” in: Proceedings of International Workshop on Database Machines, pp.166–187, 1983.

[10] C. Chang, T. Nguyen, M. Lin, and C. Lin, “A novel data-hiding and compression scheme based on block classification of SMVQ indices,” Digital Signal Processing, in press, 2016.

[11] P.C. Su, C.C. Kuo, “Steganography in JPEG2000 compressed images,” IEEE Trans. Consum. Electron., vol. 49, no. 4, pp. 824–832, Nov. 2003

[12] W.J. Wang, C.T. Huang and S.J. Wang, VQ “applications in steganographic data hiding upon multimedia images,” IEEE Syst. J., vol. 5, no. 4, pp. 528–537, Dec. 2011.

[13] C. C. Lee, W. H. Ku and S. Y. Huang, “A new steganographic scheme based on vector quantization and search-order coding,” IET Image Process, vol. 3, no. 4, pp. 243–248, 2009.

[14] J. Tian, “Reversible data embedding using different expansion,” IEEE Trans. Cir-cuits Syst. Video Technol., vol. 13, no. 8, pp. 890–896, Aug. 2003.

[15] C.C. Chang, T.S. Nguyen and C.C. Lin, “A reversible data hiding scheme for VQ indices using locally adaptive coding,” J. Vis. Commun. Image Represent, vol. 22, no. 7, pp. 664–672, Aug. 2011.

[16] C.F. Lee and Y.L. Huang, “An efficient image interpolation increasing payload in reversible data hiding,” Expert Syst. Appl., vol. 39, no. 8, pp. 6712–6719, Jun. 2012.

[17] Y.C. Hu, “High-capacity image hiding scheme based on vector quantization,” Pat-tern Recognit., vol. 39, no. 9, pp. 1715–1724, 2006.

[18] C.C. Chang, T.S. Nguyen and C.C. Lin, “A novel VQ-based reversible data hiding scheme by using hybrid encoding strategies,” J. Syst. Softw., vol. 86, pp. 389–402, Aug. 2013.

[19] S.C. Shie, S.D. Lin and C.M. Fang, “Adaptive data hiding based on SMVQ prediction,” IEICE Trans. Inf. Syst. E 89-D, pp. 358–362, 2006.

[20] W.J. Chen and W.T. Huang, “VQ indices compression and information hiding using hybrid lossless index coding,” Digit. Signal Process, vol. 19, no. 3, pp. 433–443, May 2009.

[21] C.C. Chang, T.D. Kieu and W.C. Wu, “A lossless data embedding technique by joint neighboring coding,” Pattern Recognit., vol. 42, no. 7, pp. 1597–1603, 2009.

[22] J.X. Wang and Z.M. Lu, “A path optional lossless data hiding scheme based on VQ joint neighboring coding,” Inf. Sci. vol. 179, pp. 3332–3348, 2009.

[23] C.H. Yang and Y.C. Lin, “Reversible data hiding of a VQ index table based on referred counts,” J. Vis. Commun. Image Represent, vol. 20, no. 6, pp. 399–407, Aug. 2009.

[24] S.C. Shie and S.D. Lin, “Data hiding based on compressed VQ index of images,” Com-put. Stand. Interfaces, vol. 31, no. 6, pp. 1143–1149, 2009.

[25] J.D. Lee, Y.H. Chiou and J.M. Guo, “Reversible data hiding based on histogram mod-ification of SMVQ indices,” IEEE Trans. Inf. Forensics Secur., vol. 5, no. 4, pp. 638–648, Dec. 2010.

[26] C. Qin, C.C. Chang and Y.P. Ping, “A novel joint data hiding and compression scheme based on SMVQ and image in painting,” IEEE Trans. Image Process., vol. 23, no. 3, pp. 969–978, Mar. 2014.

[27] J.D. Lee, Y.H. Chiou and J.M. Guo, “Lossless data hiding for VQ indices based on neighboring correlation,” Inf. Sci. vol. 221, pp. 419–438, Dec. 2013.

[28] C.C. Chang and T.S. Nguyen, “A reversible data hiding scheme SMVQ indices,” Infor-matica., vol. 25, no. 4, pp. 523–540, 2014.

[29] L. Wang, Z. Pan, X. Ma and S. Hu, “A novel high-performance reversible data hid-ing scheme using SMVQ and improved locally adaptive coding method,” J. Vis. Commun. Image Represent, vol. 25, pp. 454–465, 2014.

[30] C.C. Chang, T.S. Nguyen and C.C. Lin, “A reversible compression code hiding using SOC and SMVQ indices,” Inf. Sci., vol. 300, pp. 85–99, 2015.