PYHSICAL SCIENCES

ISSN (Print): 24490954
ISSN (Online): 26364972

ENSURING SECURITY AND TRUST FOR DATA IN THE CLOUD

[1]Sule Mary-Jane, , [2]Yakmut D.I. and [3]Maozhen L. I.

[1]Department of Computer Sceince
University of Jos
[2]ICT Directorate,Federal University Lafia
[3]Department of Electronic and Computer Engineering
Brunel University, London

Corresponding Email: daniel.yakmut@fulafia.edu.ng

Manuscript Received:23/07/2018 Accepted:30/07/2018 Published: December 2018

ABSTRACT

Cloud computing has a unique ability to provide on-demand metered elastic computing resources to multi-tenant users. However, given the nature of the cloud platform and its present limited ways to detect unauthorized access or modifications to data,cloud users and data owners therefore do not trust that their data is adequately secure on the cloud platforms. This paper presents a technique of data coloring for securing data on cloud platforms based on establishing and using concatenated fingerprints for cloud watermarking. Using the technique, cloud users and data-owners secure their data by first coloring it offline before uploading onto any cloud platform and subsequently they can detect any unauthorized modifications or identify paths of any unauthorized data modifications.

Keywords: *Cloud computing, cloud security, data coloring, cloud watermarking, data fingerprinting, steganography.*

INTRODUCTION

Security and trust issues have remained a critical concern and major obstacles for the full deployment and complete acceptance of cloud computing services by prospective usersdespite its unique attributes which include providing on demand metered self-service, shared, scalable and elastic virtual computing resources to multi-tenant users across a broad network (Internet / Local Area Network - LAN) (Mell and Grance, 2011; Sun *et al.,* 2011; Nelson, 2009).

Cloud computing services are usually provided through the following three service models - infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS) and Software-as-a-Service (SaaS) (Armbrust *et al.,* 2010).Depending on the need of the users, these service models are then deployed in a number of ways - Private, Public, Community or Hybrid cloud deployments.

These models require different security considerations even for major issues like confidentiality, data integrity and availability. Trust issues between providers, individual users and group users of cloud services are further pronounced because cloud users and data owners have no way of securing or confirming if their data has been tampered with during storage or processing on the cloud platform (Armbrust *et al.,* 2010; Nelson, 2009; Kaur and Mann, 2014).

Trust and security would greatly be enhanced in cloud computing when cloud users and data owners are able to secure their data before uploading onto the cloud platform and are stillable to trace and confirm any distortion to the data and the exact path of distortion.

For any data owner or organisation (no matter the size), securing data is a major priority. Therefore measures are usually put in place to protect data from beingtampered with, limit unauthorised access or illegal usage. One common measure put in place is the process of rendering data unreadable to unauthorised users through scrambling (encryption) of data, although this process could attract undue attention to the data (Zhang *et al.,* 2011)

Another measure frequently used is watermarking as various digital formats especially those for images including the portable document format (PDF) already support the easy embedding (and removal) of visible watermarks. These watermarks are usually located in well-defined sections thereby making identification and unauthorized removal easy.

However, while cloud users often assume thatthe cloud provider should secure their data, the Service Level Agreement (SLA) most times implies that data security is the duty of the user or data-owner and this is often never clearly stated or documented especially as existing methods for securing data have not been fully researched for cloud use (Sun *et al.,* 2011; Armbrust *et al.,* 2010; Hwang and Li, 2010).

This research workdescribes how data is secured using a technique of data colouring. The technique allows cloud users and data owners to secure their data by first colouring it offline before uploading onto any cloud platforms and subsequently, they can detect unauthorized modifications and identify paths of such modifications. A basic shell-script implementation of the technique based on steganography is presented along with some sample results of its deployment/use on an experimental cloud platform.

MATERIALS AND METHODS

The process of watermarking and datacolouringand its implementation through fingerprinting technique,which allows a different watermark for each distributed copy of a data set is described as follows:

Watermarking and Data Colouring.

Watermarking is a security feature that prevents and discourages counterfeiting through the addition of an identification image/pattern with varying visibility. In digital watermarking, a digital mark (pattern) is embedded in a digital file. The digital watermark, which may sometimes be hidden, serves to identify ownership (and copyrights) thereby verifying the authenticity and integrity of the digital file. An extension of the watermarking concept known as fingerprinting ensures that different watermarks are embedded in every copy of the distributed data-sets (digital files) and this aids the detection and tracking of both perpetrators and the path of data distortion(Kessler, 2014). Digital fingerprinting (watermarking), also include information that is useful for identifying unauthorised modifications to the content.

Data colouring on the other hand is a method or a process by which data owners can secure or protect their data through digital watermarking (Hwang and Li, 2010).

The data coloring process in Figure 1 according to Hwang and Li (2010) and Liu (2011) shows where the colour drops are a combination:

a. an *"expected"* value - □□ known only to the data owner,
b. the *"entropy"* value – □□ known only to the users in a particular group,

c. and the *hyperentropy* value □□ known to all the users of the cloud infrastructure .

□□, □□ and □□are combined together to generate a collection of color drops that forms a unique color that neither the cloud providers nor other cloud users can detect. Sandosh and Uthayashangar (2012),argued that the computational complexity in obtaining□□, □□and □□ is lower than that in conventional encryption and decryption process.

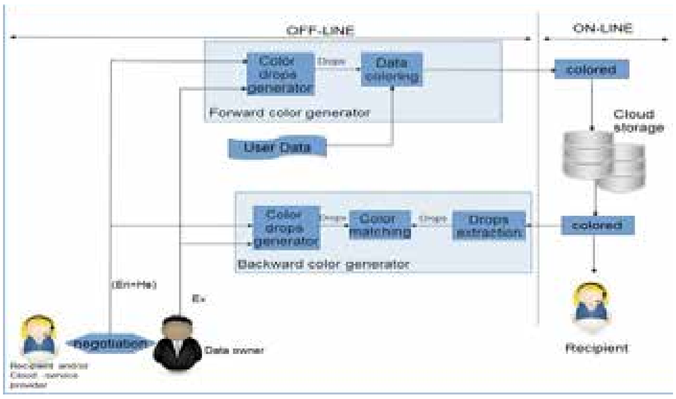


Figure 1:Data colouring implementation.

Further description of Figure 1 reveals that;
□□+□□ represents information that is agreed and exchanged between a cloud provider and a data-owner such as the public-key component of a cryptographic key-pair.
□□ is the information that is only known to the data owner such as the private component of personal cryptographic key-pair and an encryption password.

The forward color generator is composed of two distinct operations, these are the color drops generator and the data coloring process.

The color drops generator is responsible for producing a sequence of bits from the combination of the quantities□□+□□+□□.In traditional watermarking, the color drops would be the unique watermark.

The second operation inserts the generated bit sequence (watermark) into the user-data to obtain the colored data. Thecolored (watermarked) data may be subsequently stored or processed on the cloud platform or acopy maybe delivered to a recipient.

In data coloring, the colored (or watermarked) data retains all the functionality of the original data but contains additional identification bits that is included within the data in a manner that does not permit easy detection or removal of the color drops (unique watermark).

The backward color generator verifies the inserted color drops (watermark). It consist of 3 separate operations; they are - the extraction of color drops from the colored data, the generation of the color drops based on the same input parametersinitially passed to the forward color generation and an operation to compare the generated color drops to the color drops extracted from the colored data.

The coloring and/or verification of the color drops is carried out by the data-owner as they would require knowledge of □□.

In data colouring, the color drops (or watermarks) are embedded within the data (or data-set) to provide integrity and identification without impacting the functionality of the data. The presence of colour drops should be invisible (or transparent) during regular use of the data set. The process of ccolouring or embedding the colour drops within the data sets should also be resilient against unauthorized reversal while reliably supporting the authorized location/extraction of color drops.

Steganography, the art of hidden writing (Morkel, 2005), is used as the primary technique for embedding color drops into the data sets in the data ccolouringimplementation. The aim in steganography is to embed and hide the existence of a message within another carrier message from a third party. Steganography is different from Cryptography in the sense that it does not make the message unreadable from third party but just embeds and hides a message (secret communication) within it. An advantage of steganography is that it doesn't attract undue attention (Morkel, 2005; Collberg and Thompson, 2002), as the original message continues to function as normal (the hidden message is invisible or transparent).

Sometimes, the hidden message may be pre-encrypted, compressed or encoded before embedding in the carrier message (file). Also, sometimes, the hidden message may be split among a set of files but then all files must be available, unmodified and processed in the right order in other to retrieve the hidden data/message. In steganography, the security of the hidden message is cryptographically enhanced when the secret messages are first encrypted before embedding into the carrier. The hidden message is usually embedded as bit-level in the redundant space of the carrier message most times, in a statistical manner to avoid possible detection or modifications.

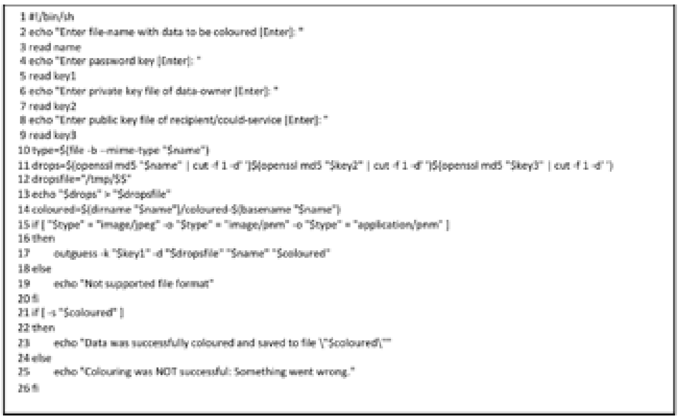
Implementation:
The implementation presented here expects that colouringbe carried out completely offline to enhance security. Therefore, this means only coloured data

[1]NIST defines Cloud Infrastructure (CI) as the combination of the hardware and software that enable cloud computing

should be uploaded to cloud platform(s). The colouring of data-files before uploading to various cloud-service models is expected to improve the integrity of the cloud-based resources as it enables data-owners to detect, trace, report and document unauthorized access and use of uploaded data/data-files to respective cloud providers or users.

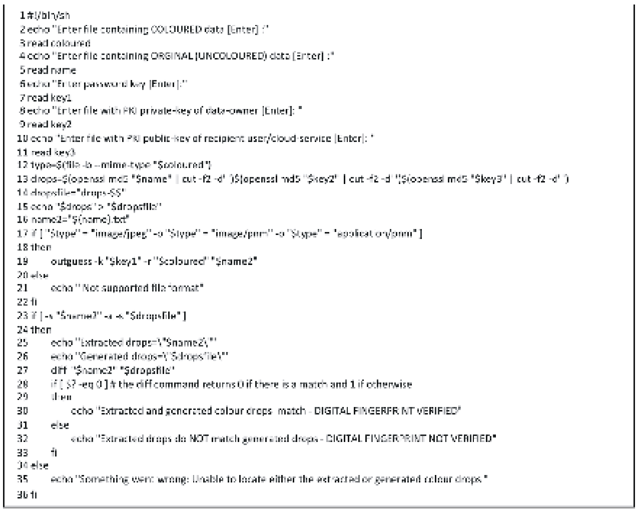
The shell-scripts used for implementation depend on the free and open-source steganography tool OutGuess for colouring data files (embedding hidden data into redundant bits of a carrier file) or extracting colour drops from already coloured files (i.e. extracting the hidden data from redundant bits). Out Guess relies on specific data handlers that would identify and modify redundant bits to carry the secret message. Out Guess is able to handle different data formats as long as a suitable handler is available.

The shell-scripts generated include the forward colour generator - fcg.sh (Figure 2) and the backward colour generator - bcg.sh (Figure 3). The fcg.sh generates the colour drops used for colouring the original data and the bcg.sh extracts the color drops from the colour file and compares with thecolour drops generated directly from the input parameters.



```
#!/bin/sh
echo "Enter file-name with data to be coloured [Enter]: "
read name
echo "Enter password key [Enter]: "
read key1
echo "Enter private key file of data-owner [Enter]: "
read key2
echo "Enter public key file of recipient/could-service [Enter]: "
read key3
type=$(file -b --mime-type "$name")
drops=$(openssl md5 "$name" | cut -f 1 -d' ')$(openssl md5 "$key2" | cut -f 1 -d' ')$(openssl md5 "$key3" | cut -f 1 -d' ')
dropsfile="/tmp/$$"
echo "$drops" > "$dropsfile"
coloured=$(dirname "$name")/coloured-$(basename "$name")
if [ "$type" = "image/jpeg" -o "$type" = "image/pnm" -o "$type" = "application/pnm" ]
then
      outguess -k "$key1" -d "$dropsfile" "$name" "$coloured"
else
      echo "Not supported file format"
fi
if [ -s "$coloured" ]
then
      echo "Data was successfully coloured and saved to file \"$coloured\""
else
      echo "Colouring was NOT successful: Something went wrong."
fi
```

Figure 2: The forward colour generator script (fcg.sh)



```
#!/bin/sh
echo "Enter file containing COLOURED data [Enter] :"
read coloured
echo "Enter file containing ORGINAL (UNCOLOURED) data [Enter] :"
read name
echo "Enter password key [Enter]:"
read key1
echo "Enter file with PKI private-key of data-owner [Enter]: "
read key2
echo "Enter file with PKI public-key of recipient user/cloud-service [Enter]: "
read key3
type=$(file -b --mime-type "$coloured")
drops=$(openssl md5 "$name" | cut -f2 -d' ')$(openssl md5 "$key2" | cut -f2 -d' ')$(openssl md5 "$key3" | cut -f2 -d' ')
dropsfile="drops-$$"
echo "$drops" > "$dropsfile"
name2="$(name).txt"
if [ "$type" = "image/jpeg" -o "$type" = "image/pnm" -o "$type" = "application/pnm" ]
then
      outguess -k "$key1" -r "$coloured" "$name2"
else
      echo " Not supported file format"
fi
if [ -s "$name2" -a -s "$dropsfile" ]
then
      echo "Extracted drops=\"$name2\""
      echo "Generated drops=\"$dropsfile\""
      diff "$name2" "$dropsfile"
      if [ $? -eq 0 ] # the diff command returns 0 if there is a match and 1 if otherwise
      then
            echo "Extracted and generated colour drops match - DIGITAL FINGERPRINT VERIFIED"
      else
            echo "Extracted drops do NOT match generated drops - DIGITAL FINGERPRINT NOT VERIFIED"
      fi
else
      echo "Something went wrong: Unable to locate either the extracted or generated colour drops."
fi
```

Figure 3: The backward colour generator script (bcg.sh)

RESULTS AND DISCUSSION

The results of the implementation of the data colouring technique from Table 1shows the sources of color drops used in the data-coloring implementation. The cryptographic hash of a Public Key Infrastructure (PKI) privatekey of the dataowner guarantees the color drops contain information that ascertains ownership, while the hash of the PKI publickey of data recipient or cloudservice is useful to trace/highlight path of data loss or theft and the hash of the data content itself is useful for detecting unauthorised modifications.

Table 1:Data sources for colour drops generation

Item	Contribution
Data-file to be coloured	Fingerprint to detect unauthorised modifications to content
Private-key of data-owner	Fingerprint to identify owner or data-colourer
Public key of recipient or cloud-service	Fingerprint to trace path of data-loss/theft

Furthermore, a password is used during the embedding process to encrypt the colour drops thereby securing them against unauthorized modification and/or removal.

The use of the original data-file as well as suitable PKI keys such as Pretty GoodPrivacy (PGP) keys for creating the digital-fingerprint (watermark) guarantees uniqueness (and entropy) while also satisfying other defining conditions of □□, □□ and □□ such as, knowledge limited to data-owner and association to defined group of users (or cloud-platform).

Theft and Loss Responsibilities

An important feature of the data colouring implementation is its ability to highlight a path of data-loss/theft based on finger printing. In Table 2, a simple matrix is presented to show how the theft/loss responsibilities (path) may be determined from the corresponding inputs used during the data colouring process. Row 1 represents the classical watermarking process - as only the identity ofthe owner is verifiable from the colour drops (watermark). Rows 2, 3 and 5 suggest that colour drops based on the corresponding combinations would not carry owner information and in such cases, the drops cannot be used to prove ownership of the data. Row 4 and 6 suggests combinations for which the drops may also be used to identify either a Cloud Service Infrastructure (CSI) or a Cloud Service Provider (CSP) or single-recipient. Row 7 highlights the combination for which drops are capable of also identifying individual CSP, CSI and recipient.

As described in Figure 1, the verification of color drops are expected to be carried out by the data-owner.

Table 2: Theft/Loss Responsibilities

	Private Key of data-owner	Public key of cloud-service	Public key of data recipient	Information obtained from Drops
1	YES	NO	NO	Identity OF data-owner
2	NO	YES	NO	Identity OF CSI
3	NO	NO	YES	Identity of recipient (CSP)
4	YES	YES	NO	Identity of both owner and CSI
5	NO	YES	YES	Identity of both CSI and recipient (CSP)
6	YES	NO	YES	Identity of both owner and recipient (CSP)
7	YES	YES	YES	Identity of owner, CSI and recipient (CSP)

Cloud platform and testing

Cloud platform

A trusted cloud-computing platform was deployed using Eucalyptus - is both paid and open source cloud hosting computing environment. The deployment was enabled by the Trusted Platform Module (TPM – ISO/IEC 11889) – astandard for a secure cryptoprocessor, a dedicated microcontroller designed to secure hardware through integrated cryptographic keys.The cloud platform integrates end-user accessible TPM integrity measurement/verification without the need for "custom" software or patches. Furthermore, on the platform, security is enhanced by the inclusion of an instance-level file and directory integrity checker for selected files and directories. In this cloud deployment approach, individual organisations or users share a common cloud platform and sometimes not necessarily retaining control over their sensitive data or applications deployed on (foreign) infrastructure (Sule *et al.,* 2015).

The data-colouring implementation reported inthis research work is aimed at providing integrity/protection of uploaded data, as it would ensure that each operator/user is able to retain and verify ownership of sensitive data with a flexible access model based on data sharing needs.

Testing

The script described in Figure 2 would colour the data file and the script described in Figure 3 would extract and verify the coloured file.

The first input item to the script that would colour the data includes the data file to be coloured followed by the encryption password. The data-owner is identified by a DSA private-key taken from the secure shell (SSH) application; while a PGP public-key belonging to the cloud-service provider is used to identify the cloud-platform/service.

The set of scripts would require the original file as well as the coloured version for successful verification. The steganography tool (outguess) can extract the colourdrops from the coloured file once the right password is provided. The original file is needed for the generation of a new set of colour-drops.

CONCLUSION

The general objective of this study is to show that data in the cloud can be secured and trusted when the appropriate security measure is implemented. Particularly, the study has demonstrated a technique of data colouring for securing user data resources in cloud platforms.The implementation of this security technique creates colour drops from concatenated fingerprints that allow the verification of dataowner, cloud service provider or recipient while also protecting against unauthorised modifications.

The concept of datacolouring can be applicable to all data formats; however based on the OutGuess steganography tool, the present implementation described this studycan only be used on the following digital image formats: Joint Photographic Experts Group (JPEG), Portable Pix Map (PPM) and Portable Any Map (PNM). Future work could investigate other data formats and steganography toolsthat can be implemented and supported.

REFERENCES

Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G.,Patterson, D., Rabkin, A., Stoica, I. and Zaharia, M. (2010). Clearing the clouds away from the true potential and obstacles posed by this computing capability,*Communication of ACM. 53(4):*50-58.

Collberg, C.S. and Thomborson, C. (2002). Watermarking, tamper-proofing, and obfuscation - tools for software protection. *IEEE Transactions on Software. Engineering 28(8)*:735-746.

Hwang, H. and Li, D. (2010). Trusted cloud computing with secure resources and data coloring, *Internet Computing, IEEE. 14(5):4– 22.*

Kaur, S. and Mann, P.S. (2014). A review on cloud computing issues and challenges. *International Journal of Research in Computing Application and Robotics. 2(5)*:63-68.

Kessler, G.C. (2014).An overview of steganography for the computer forensics examiner. *Forensic Science Communicaion.6(3).*

Liu,Y., Ma,, Y., Zhang, H. and Li, D. (2011). A method for trust management in cloud computing: data coloring by cloud watermarking,*International. Journal of Automation and Computing . 8(3):*280–285.

Mell, P. and Grance, T. (2011). The NIST definition of cloud computing [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf [Last Accessed: 22 June 2018].

Morkel, T. (2005). Steganography and Seganalysis. [Online]. Available: http://www.seralliance.com/enews/vol2no1/pdfs/steganography_jan05.pdf. [LastAccessed 24th June 2018].

Nelson, M. (2009). Briefing paper for the ICCP technology foresight forum cloud computing and public policy. [Online]. Available: http://www.oecd.org/sti/ieconomy/43933771.pdf [Last Accessed: 6th June 2018].

Nelson, M.R. (2009). The cloud, the crowd, and public policy.issues in science and technology, *Issues in Sci. and Technology, 5(4).*

Sandosh, S. and Uthayashangar, S. (2012). An authentication in cloud through data coloring using progressive approach.*International Journal of Current Research and Review. 4(21):*179 – 182.

Sule, M., Li, M.,Taylor, G.A, and Furber, S. (2015). Deploying trusted cloud computing for data intensive power system applications. 50th Golden Int. Universities Power Eng Conf. (UPEC 2015), Stoke on Trent.

Sun, D., Chang, G., Sun, L. and Wang, X.(2011). Surveying and analyzing security, privacy, and trust issues in cloud computing environments. *Proc. Eng., 15*:2852 – 2856.

Zhang, X., Du, H., Chen, J. and Zeng, L. (2011). Ensure data security in cloud storage. Proceedings of Int. Conf. on Network Computing and Information Security. 2011