Journal of Software Engineering

ISSN 1819-4311

Academic Journals Inc.

www.academicjournals.com

Journal of Software Engineering 9 (2): 410-418, 2015
ISSN 1819-4311 / DOI: 10.3923/jse.2015.410.418

Software Watermarking in the Cloud: Analysis and Rigorous Theoretic Treatment

Sun Guang, Fan Xiaoping, Fu Sha, Song Yingjie and Luo Huifang
Department of Information Management, Hunan University of Finance and Economics, Changsha, Hunan, Fenlin 2nd Road, No 139, 410205, People's Republic of China

Corresponding Author: Sun Guang, Department of Information Management, Hunan University of Finance and Economics, Changsha, Hunan, Fenlin 2nd Road, No 139, 410205, People's Republic of China

ABSTRACT

This study offers an analysis on "What's new" in software piracy when software applications move to the Cloud and develop rigorous theoretic treatment on applying software watermarking in the Cloud. The "new" analysis focus on the changes emerged with X-is-a-Service. The rigorous theoretic treatment of software watermarking involves ADT and trusted extract function. In particular, a DFA is employed to formally define embedding and extracting process. Rigorous treatment is the first and significant step to fight the "new" in software piracy. It's very vital to develop approaches that protect Cloud software from piracy.

Key words: Software piracy, software watermarking, cloud computing, formal definition

INTRODUCTION

Cloud computing, refers to both the services produced by the Cloud applications via Internet and the services provided by hardware and systems software in the datacenters. There are many new opportunities and advantages of Cloud computing, such as infinite computing resources made available on demand, elimination of an up-front commitment, scalability of virtual machines (CSA., 2011), therefore plenty of people are ready to or already jumping on the Cloud (CSA., 2010). The steady growth of Cloud computing is undoubted. However, security is still an unavoidable barrier to the success of Cloud computing. Academia, industry, government and organizations have expressed inadvisable or insecure parts of Cloud computing frequently (CSA., 2013). It's well worth concerning some aspects which are new compare to conventional Computing.

SaaS (Software as a Service) means software applications are shifting onto Cloud storage for enabling service and on-demand network access. Many people include some members of BSA, believe that SaaS model will greatly minimize software piracy (CSA., 2012). The reason is that, migrating programs like Microsoft Office from the desktop to the Cloud separates adversary with source code, software piracy will be harder to bear off (Yu *et al*., 2012). Adobe Systems CEO Shantanu Narayen gives us another reason, SaaS applications will continuously require Internet connection, are able to gain basic control over their applications, thus most likely solving piracy problem on its own. Kevin Lalor, Business Intelligence 101 CEO and Founder and David Hoff, Cloud Sherpas Chief Technology Officer, share the same point with Narayen regarding software piracy. According to them, Cloud's subscription model will eventually eliminate software piracy (Hashizume *et al*., 2013).

Nevertheless, believing in Cloud is a vanguard in the fight against unauthorized copying or distribution of copyrighted software turns Cloud users into a Pollyanna (Rosado *et al*., 2012). Considering the scene of IaaS (Infrastructure as a Service), one uploads software to the Cloud, others download this software from the Cloud. The outsider threat to software copyright in this scenario could be effectively solved by modern cryptography (Grobauer *et al*., 2011), that is one uploads encrypted software, delivers the key to legal users via trustworthy channels, others download decrypted software. However, the insider threats turn to more difficult to tackle. Who guards the guards? We have no way to monitor the security arrangements in the Cloud yet (Jansen, 2011). David Finn, associate general counsel for anti-piracy, said that the rise of Cloud Computing will make software pirates even tougher to clamp down on the copy-crazy crooks. The Cloud can't be proved trustworthy in all relevant ways (Zissis and Lekkas, 2012). Another proof is sharing of login credentials. New survey released by BSA (2012) showed that 42% of respondents share login credentials for paid Cloud computing applications within their organization (Bisong and Rahman, 2011). Simultaneous logins using one account is a clear abuse of terms of paid Cloud service or licensing. What is worse, "Dark Clouds" which can be used to provide with pirated SaaS and "gray Clouds"aims to share legal software or login credential pose a big problem as over 21% of organizations do implement private Clouds which are high incidence area of changing the color of private Cloud into dark or gray (Dai *et al*., 2012).

We argued that Cloud computing complicates anti-piracy problem compare with conventional computing. Cloud application services are often outsourced to a third party, so SaaS providers can also be a SaaS users. For example, a mashup provider of rental maps usually is a user of the Google maps. These relationships between Saas providers and SaaS users are recursive, more complicate to understand than conventional computing. Cloud's large scale serious damages the ability to distinguish these recursive relationships, results in lots of uncertainty about how software copyright is moved to Cloud computing (Chroni and Nikolopoulos, 2012). A software owner must know the actual use of his digital asset and the impact that moving to the Cloud will have on those assets. Adopting Cloud architecture without properly addressing piracy considerations can result in serious errors (Yu *et al*., 2012). The goal of the present study is to clarify some terms, provide our view of applying software watermarking in the Cloud, to detail comparisons between of software copyright protection issues in the Cloud and conventional computing environment and reveal our mathematical definitions of software watermarking used in the Cloud. Finally, a DFA is employed to formally define watermark embedding and extracting process. Rigorous treatment is the first and significant step to identify the "new" in software piracy (Wu and Wu, 2012). It's very vital to develop preferable approaches protecting Cloud software application copyright.

Software watermarking is already being adopted broadly within on-premise systems. But, to date, little practical study is available on why and how to apply software watermarking in a Cloud environment. The further rigorous treatment is rare. Zhu (2007) gave a brief overview of software watermarking. It describes the taxonomy, attack models and algorithms of software watermarking (Zhu, 2007). Zhu (2007) work filled a void and provides guidance on concepts researcher interested in successfully making first step to develop software watermarking approaches used in the traditional computing environments. Furthermore, the formal definition of the representative extracting algorithm is used to a general software watermarking embed function (Zhu, 2007). But even the principles of software watermarking remain unchanged, the usage and piracy model, other specific technical considerations in the Cloud differ fundamentally from those in traditional IT environments. Software watermarking needs to develop from the basic concept definition.

With software applications are moving to Cloud, software distributions within the Cloud is supposed to mark the beginning of new worries about security breaches, privacy abuses and access

control violations (Yu *et al*., 2012). Software watermarking should review its traditional threats and understand the new challenges when protect the software copyrights in the Cloud. While the goal of software watermarking does not change with the Cloud, the "how" of software watermarking does need to be adapted for Cloud environments.

Yu *et al*. (2012) identified an insider threat to access control to collaborative sharing of copyright software which requires special attention when a move to the Cloud. They address this threat using software watermarking and propose a design for a Cloud-based watermarking service. However, there are no whole threats to Cloud software, no general rigorous formal treatment on adapting software watermarking to the Cloud.

Wu and Wu (2012) developed a formal definition of p trust pair given as the base for the further concepts of trust pair of coded circuit. The concept of trust pair stems from the theory called 'Relativism of security'. The relativism addresses security property is relative under different circumstances which is related to a verifier or a judge. They given a rigorous formal definition of trusted computing based on the concept of trust pair (Wu and Wu, 2012). More important is, they reveal that software watermarking has a deep relation with trusted computing and construct software watermarking on trust pair. They argue that if watermark needs to specifically address private, watermark and verifier is a trust pair. Their works differ in what environment is to ours. The trust pair is used to integrity of watermark, regardless of applying background. The present study has a premise, must adapt the treatment on Cloud environment.

ANALYSIS ON SOFTWARE USAGE AND PIRACY IN THE CLOUD

Software usage for the cloud: The use of Cloud systems shaping the way software is delivered and employed which roughly involves three Cloud computing service models. This study excludes infrastructure software from item software; focus on software applications, because we don't intend to set any presumption on the infrastructure software which is at one end of the spectrum. In Fig. 1, there is summorized a the model that how to use software in the Cloud. The process, from

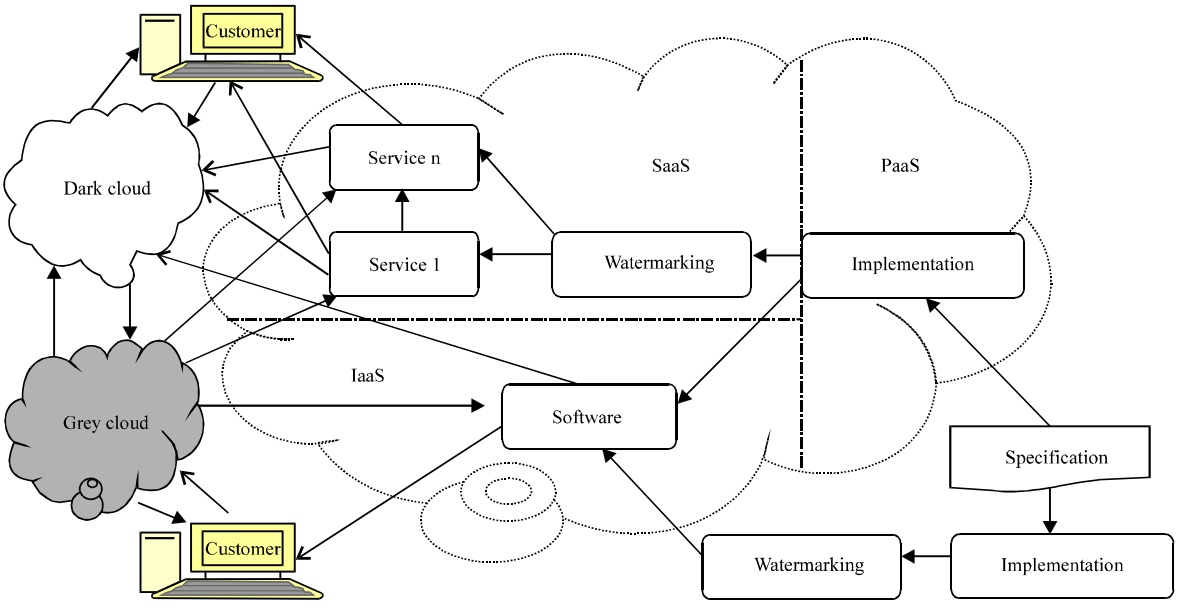

Fig. 1: Software usage and piracy in the Cloud

the requirement specification to the implementation, can be achieved in two ways. First, Cloud providers offer a computing platform, including operating system, programming language execution environment, database, web server and according hardware. Developers develop and run their applications on the platform, the underlying computer and storage resources scale automatically to match application demand so that the Cloud user can offer software service conveniently. This utility model is what we call "PaaS" (Platform as a Service). We draw PaaS part on the right of Fig. 1 to show the fact that the software applications implemented in terms of PaaS model could be delivered to either SaaS or IaaS.

Second, developers upload software to the Cloud, after that, customers download expected software from the Cloud. In this scenario, developers complete their software with own computers and install operating-system images and their application software on the Cloud infrastructure latterly. Cloud providers supply the resources on-demand such as a virtual-machine disk image library, file-based storage, load balancers and software bundles to help developers to distribute the applications, make software available to general public. Without the on-demand resources, from software developer and customer's view, Cloud infrastructure would more likes a huge transfer station; the developers rent storage space from a Cloud provider and customer then access the stored software. We refer this Cloud-service model to "IaaS".

The last model is SaaS, SaaS means the applications delivered as a service over Internet, make Cloud Computing attractive to both Cloud users (SaaS providers) and SaaS users (customers). Cloud providers manage the Cloud infrastructure and platforms that run the pay-per-use applications, Cloud users deploy their applications in the Cloud, customers are provided access to on-demand software application. Cloud Computing providers many often expose the interfaces or APIs of software that other service providers can build upon these interfaces to manage, interact with their own software and offer value-added services to their customers.

Analysis on software piracy occurred in the cloud: When the IaaS model runs, the outsider threat to software copyright could be effectively solved by modern cryptography (Yu *et al*., 2012). Developer uploads encrypted software, delivers the key to legal users via trustworthy channels, user download the decrypted software from Cloud. However, the insider threats come from the Cloud turn to more difficult to tackle. Who guards the guards? Apparently, Cloud users have no way to monitor the security arrangements in the Cloud (Ren *et al*., 2014). The situation of PaaS is similar, insider threats almost concern all.

Given SaaS model, two outstanding factors might influence the strategy against the software piracy in the Cloud. First, SaaS can be recursive, by reason of SaaS providers might be a SaaS users. For example, a mashup provider of rental maps likely is a user of the Google maps services. The recursive relationships are of extensive scale, increases monitoring of Cloud services risk, so tracing pirate behavior would leads us through the maze of caves. Equally kittle, as the only plausible solution to very high availability of SaaS is multiple Cloud computing providers, Cloud computing service management is impossible to carry out by a single company. Further, independent software stacks are usually provided by different companies. To justify creating and maintain two stacks by one company in the name of software dependability is very hard. Even if the company has multiple datacenters in different geographic regions using different network providers. Apparently, such situation makes the supervision on software piracy in the Cloud difficult.

In addition, the BSA (2012) released its new survey reported the possible for "Dark Clouds" and "Grey Clouds" to proliferate in the Cloud. Dark Clouds refer to private or public Clouds that deploy

pirated SaaS. The pirated SaaS run in dark Clouds derived from both SaaS and IaaS. Also, "Gray Clouds" is somewhere people purchase software legally and share their licenses to other users even outside of their organization. In InformationWeek's 2013 Private Cloud Survey, when 21% of organizations move their operations to the Cloud, do implement private Clouds, dark or gray Clouds will pose a big problem.

FORMAL DEFINITION OF SOFTWARE WATERMARKING IN THE CLOUD

Can Cloud Computing Stop Software Piracy? If the answer is positive, who guards the guards? We have no way to monitor the security arrangements in the Cloud yet. The rise of Cloud Computing will make software piracy even tougher to clamp down on the "dark Clouds" and "gray Clouds." Sharing of login credentials, Cloud services outsourced to a third party and Cloud's large scale worse this situation, results in lots of uncertainty about software copyright protection in the Cloud.

To research applying software watermarking in the Cloud, we give rigorous definition of software watermarking in the Cloud in section 3 and 4. We argued that identifying what software piracy appears "new" in the Cloud which is relative to "traditional" is the first step. No rigorous definition of software watermarking in the Cloud, the "new" could not be understood and treated. This is why software watermarking is shifting onto Cloud needs rigorous treatment on its concepts to give out a clear understanding of usage in the Cloud. At first, we define ADT (abstract data type) as follow:

Definition 1: ADT of software watermarking

ADT software watermarking

{

Data object:

$D = \{p_i, w_i, k_i \mid 1 \le i \le n, n \ge 0, p_i \text{ is a program}, w_i \text{ is a watermark inserted into } p_i, k_i \text{ is key to extract } w_i\}$

Data relationship:

$R = \{< (p_i, w_i, k_i), (p_{i+1}, w_{i+1}, k_{i+1}) > \mid (p_i, w_i, k_i), (p_{i+1}, w_{i+1}, k_{i+1}) \in D, i = 1, \ldots, n-1\}$

Basic computing:

Embed (P, W, K): Insert a watermark into a program, construct a watermarked program P'

Extract (P', W, K): Extract the watermark w from P' with k

}

Software watermarking is a software protection technique commonly used in the process after implementation. If software is implemented on developer's computer and delivered to IaaS, software watermarking is used as traditional protection tool. If software is implemented on the Cloud platforms, software watermarking is sold "as a service" to watermark Cloud software application. Ideally, extract function will have no false negatives (Zhu, 2007), it will always extract trusted watermark w from p' that was inserted into P. This requirement is formally expressed as definition 2.

Definition 2: Trusted extract function

$\forall p \in P, \forall k \in K, \forall w \in W: \text{Extract}(\text{Embed}(p, k, w), k) = w$

This trusted extract function of software watermarking becomes more difficult to guarantee in the Cloud. As presented in the Fig. 2, software watermarking is sold "as a service" to watermark Cloud software application. Conceptually, we assume software watermarking service runs under

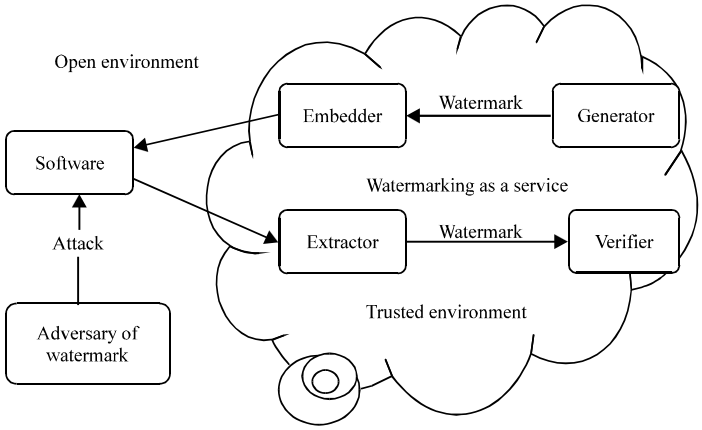

Fig. 2: Software watermarking as a service

trusted environment. Every part of software watermarking service, watermark generator, embedder, extractor and watermark verifier is closed to outsiders and keeps out any possible adversaries. As for the software application which is the data object to watermark possibly runs in the remote Cloud with dangerous adversaries in it. This open Cloud environment makes the guarantee of trusted extract function harder for real.

Trust pair (U, V) is the pair of computing unit run in open environment and verifier model work in trusted environment, used to analysis the measure V can reveal the status of computing units U. This concept is proposed by Wu and Wu (2012) who gave out a sound measure theory of trust. Trust pair is rigorous treatment on trusted computing. Trust pair grasps the concept of trust and its measurement. We refer to the definition of trust pair to introduce formal construction of software watermarking on trusted extract function in the Cloud.

Definition 3: Trusted extract function in the Cloud

$$\forall A \in PPT, Pr[Extract1(P'_A(U, x_n, w, k)) \neq Extract2(P'_A(U, x_n, w, k))] < \frac{1}{p(n)}$$

For any adversary to attack on U which runs program P', using any algorithm A, with input of a computing configuration of U which has the length of p(n), in the lifetime of U, for any polynomial p, for every input x: p(n) denotes a polynomial of n; PPT is the abbreviation of probabilistic polynomial time Turing machine; P'_A (U, x_n, w, k) is the attacked program P' by A; x_n is the input; w is the watermark; k is the key merged with w; Extract 1 () is the watermark extracting function works at U, Extract 2 () is the watermark extracting function works at the service. This definition means software watermarking service can extract the same watermark as works on computing of U. In brief, open environment has no effect on the remote watermark extraction process.

DEFINING EMBEDDING AND EXTRACTING PROCESS WITH DFA

A software watermarking process is a logically related pair of embedding a unique identifier within a piece of software, extracting the ownership information from the watermarked software.

Both watermark embedding and extracting is a sequentially process aims to manipulate the software code. Deterministic Finite Automaton (DFA) is a appropriate tool to present the serial behaviors of software watermarking.

The watermark e mbedding is a DFA defined by 6-tuple $(Q, \Sigma, \Psi, \Delta, \wedge, q_0)$ where $Q = \{q^{\circ}, q^{i}, \ldots\ldots, q^{m-1}\}$ is the states in the embedding process, every state denotes a step in the embedding process.

$\sum = \{C_0, C_1, \ldots, C_{n-1}\}$ is the input elements which stand for the instructions of the software p.

$w = <w_0 \ldots, w_i, \ldots w_{n-1}>$ is the output elements which stand for the instructions of the software p'.

$\delta_i = \Sigma \rightarrow Q$ is the next-state function for state q_i. If C_i is not a watermark carrier, , state stay the same. If C_i is a watermark carrier, the state change depends on the watermarking algorithm, the start state equal to the end state, state execution trace is a loop.

$\Delta = \{\delta_0, \delta_1, \ldots\ldots, \delta_{l-1}\}$ is the set of all next-state functions.

$\lambda_i = \Sigma \rightarrow \Psi$ is the output function for state q_i. For the same way as next-state function, if C_i is not a watermark carrier, $\lambda_i = c_i \rightarrow c_i$, embedder do nothing with C_i. If C_i is a watermark carrier, the output depends on the watermarking algorithm.

$\wedge = \{\lambda_0, \lambda_1, \ldots\ldots, \lambda_{l-1}\}$ is the set of all output functions.

$q_0 \in Q$ is the starting state of the watermark embedding process.

Also, the watermark extracting process is a DFA. If watermark embedding process is a complier, extracting process is a decomplier accordingly. The DFA of extracting process defined by 6-tuple $(Q', \Sigma', \Psi', \Delta', \wedge', q_0)$, where $Q' = \{q_0, q_1, \ldots\ldots, q_{m-1}\}$ is the states in the extracting process, every state denotes a step in the extracting process. There is no such requirement $m' = m$, that is we can't ask embedding process finish its work in same steps to embedding process.

$\Sigma' = \Psi$ is the input elements which stand for the instructions of the software P'.

$\Psi' = w$ is the output alphabet which is the watermark w of the software P' if extracting process is success.

$\Delta' = \{\delta'_0, \delta'_1, \ldots\ldots, \delta'_{l-1}\}$ is the set of all next-state functions of embedding process. Likewise, there is no such requirement $l' = l$, that is we can't ask embedding process finish its work in same steps to embedding process. $\delta'_i: \Sigma' \rightarrow Q'$ is the next-state function for state q_i. If C'_i is not a watermark carrier, $\delta'_i: c'_i \rightarrow q_i$, state stays the same. If C'_i is a watermark carrier, the state change depends on the watermarking algorithm, the start state equal to the end state, state execution trace is a loop.

$\wedge' = \{\lambda'_0, \lambda'_1, \ldots\ldots, \lambda'_{l-1}\}$ is the set of all output functions. $\lambda'_i = \Sigma' \rightarrow \Psi'$ is the output function for state q_i. For the same way as next-state function, if C'_i is not a watermark carrier, $\lambda'_i = c'_i \rightarrow$ null extractor do nothing with C'_i. If C'_i is a watermark carrier, $\lambda'_i = c'_i \rightarrow w_i$, $w = <w_0, \ldots, w_i, \ldots w_{n-1}> n, \geq 0$. More often, each watermark carrier hides a piece of watermark.

$Q_0 \in Q'$ is the starting state of the watermark extracting process.

CONCLUSION

The work of this study could be used to fill a void and provide primary guidance on challenges software owners face in integrating software protection within their Cloud computing environments. Software protection plays an important role in Cloud computing, it's a considerable attempt to enhance the Cloud information security. Preventing pirate software will have a huge influence on economic development. There are various threats in the Cloud, thorough threat analysis and rigorous treatment such as mathematical definition become very necessary. This study analyses the software usage in the Cloud. In this part we strive to frame the full space of Cloud Computing software usage issues, attempting to separate j ustified concerns of the threats to

software in the Cloud from possible over-reactions. We argue that SaaS model is fundamentally new for software piracy which should received more attention. On the other hand, the formal definitions were introduced in this study such as ADT of software watermarking and trusted extract function in the Cloud. We argue that rigorous treatment on software watermarking in the Cloud is the first and significant step to anti-piracy in the Cloud. DFA is employed to formally define embedding and extracting process in software watermarking. The present study tried to take a first step to against software piracy in the Cloud, it is very vital to develop approaches that protect software from threats.

REFERENCES

BSA., 2012. Emerging markets ready to adopt paid cloud services-and likely to share log-in credentials, BSA survey finds. Business Software Alliance (BSA), Washington, DC., USA., July 18, 2012.

Bisong, A. and S.S.M. Rahman, 2011. An overview of the security concerns in enterprise cloud computing. Int. J. Network Secur. Applic., 3: 30-45.

CSA., 2010. Top threats to cloud computing V1.0. Cloud Security Alliance (CSA), March, 2010, pp: 1-14. https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf.

CSA., 2011. Security guidance for critical areas of focus in cloud computing V3.0. Cloud Security Alliance (CSA). https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf.

CSA., 2012. Security guidance for critical areas of mobile computing, V1.0. Mobile Working Group, Cloud Security Alliance (CSA), Washington, DC., November 2012, pp: 1-60.

CSA., 2013. The notorious nine: Cloud computing top threats in 2013. Top Threats Working Group, February 2013, Cloud Security Alliance (CSA), pp: 1-21.

Chroni, M. and S.D. Nikolopoulos, 2012. An efficient graph codec system for software watermarking. Proceedings of the IEEE 36th Annual Computer Software and Applications Conference Workshops, July 16-20, 2012, Izmir, Turkey, pp: 595-600.

Dai, P., C. Wang, Z. Yu, Y. Yue and J. Wang, 2012. A software watermark based architecture for cloud security. Proceedings of the 14th Asia-Pacific Web Conference on Web Technologies and Applications, April 11-13, 2012, Kunming, China, pp: 270-281.

Grobauer, B., T. Walloschek and E. Stocker, 2011. Understanding cloud computing vulnerabilities. IEEE Secur. Privacy, 9: 50-57.

Hashizume, K., D.G. Rosado, E. Fernandez-Medina and E.B. Fernandez, 2013. An analysis of security issues for cloud computing. J . Internet Services Applic. (In Press). 10.1186/1869-0238-4-5

Jansen, W.A., 2011. Cloud hooks: Security and privacy issues in cloud computing. Proceedings of the 44th Hawaii International Conference on System Sciences, January 4-7, 2011, Koloa, Kauai, HI., USA., pp: 1-10.

Ren, C., K. Chen and P. Liu, 2014. Droidmarking: Resilient software watermarking for impeding android application repackaging. Proceedings of the 29th ACM/IEEE International Conference on Automated Software Engineering, September 15-19, 2014, New York, USA., pp: 635-646.

Rosado, D.G., R. Gomez, D. Mellado and E. Fernandez-Medina, 2012. Security analysis in the migration to cloud environments. Future Internet, 4: 469-487.

Wu, H. a nd G. Wu, 2012. On the concept of trusted computing and software watermarking: A computational complexity treatise. Proceedings of the International Conference on Solid State Devices and Materials Science, Volume 25, April 1-2, 2012, Macao, pp: 465-474.

Yu, Z., C. Wang, C. Thomborson, J. Wang, S. Lian and A.V. Vasilakos, 2012. A novel watermarking method for software protection in the cloud. Software: Pract. Exp., 42: 409-430.

Zhu, W., 2007. Informed recognition in software watermarking. Proceedings Pacific Asia Workshop, Intelligence and Security Informatics, Volume 4430, April 11-12, 2007, Chengdu, China, pp: 257-261.

Zissis, D. and D. Lekkas, 2012. Addressing cloud computing security issues. Future Gener. Comput. Syst., 28: 583-592.