Volume 3, Issue 8, August 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Edifying the Issue of Software Security and Solution by Software Watermarking in Cloud Computing Environment

Navneet Singh
Nitttr
Bhopal, India

Professor Shailendra Singh
(member IEEE) *Nitttr*
Bhopal, India

Abstract:* With the rapid development in the technology of cloud computing all the resources are available on demand which can be according to the requirement. But with the growing technology and trend, the security threat also resides as a big issue in parallel with the advancement of the technology in the stream of cloud computing. In this paper we present a descriptive survey on various threats that can attack the software user as well as provider and some solutions that by watermarking technology the software can be secured even in the cloud computing environment as we apply watermarking for a generic copyright protection. We also present various algorithms available for watermarking the software in accordance with either the security or for efficiency that is according to the priority of either the owner of the software or the provider who is deploying the services over their cloud framework.

Keywords: software watermarking, watermark embedding, cloud protection, graph based watermarking

I. INTRODUCTION

Now a day's applications are migrated on to the virtualization environment rather than being stored physically on a local machine. By drifting the applications towards the cloud storage, it gains advantages over storing it locally, one is of space complexity and another one is the usage time, only pay for what you have used. Software distributions of any cloud are subjected to breaching the security, various other privacy issues and many access violations also is a major concern for the cloud providers like amazon's version of cloud service i.e. amazon ec2 (elastic compute cloud). Even the insider threat i.e. security breach inside an organization is still not eliminated completely by taking respective methods such as cryptography [3, 4], various authentication mechanism and digital signature. The global revenue loss due to software piracy on an approximation is round about $50 billion, software companies are taking regulatory action to protect the whole development process by integrating various security checks but still we are facing the illegal accessing of the software [3] and its underlying code. Illegal copying is also a major issue that motivates the methodology for protecting the software. It lowers the overall profits of the manufacturers, distributive authority and also the service providers and most importantly affecting the governmental tax revenues now this is something really serious and some preventive measures has to be taken against this. Furthermore when copyright holder and suspected infringer are in different social grouping, it creates some difficulties for copyright holder to enforce a sanction. This reason motivated to find technical solutions for detecting copyright infringements and responding to suspected ones. This is a statistics of a survey, the data provided by business software alliance (May 2012), it states that people who acquire the unlicensed software all the time. Pirates are more likely to occur than non-pirated users this is generally done by installing one copy of software on more than one personal computer. The business analyst admits that business decision makers and the general user involve in software piracy.The graph is among always, mostly or rarely with their percentage ratio's. This is the scenario about the piracy of software for the year 2011-2012. Now what actually is the watermarking is the next issue to be discussed for, it involves embedding a unique identifier within a piece of software that helps in discouraging the piracy.The key can be a signature of the organization or the name of the user to which the software is been deployed it can be anything depending upon the algorithm or the individual decision of the organization.

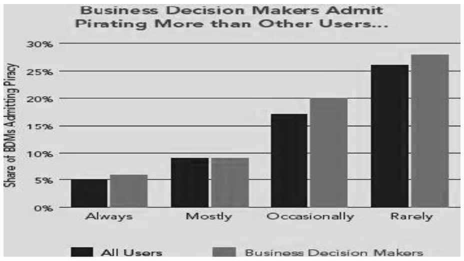

Figure 1: analysis of software piracy

Now the embedded watermark can be retrieved later on to authenticate the owner of the software. The whole process is called is embedding the watermark and detection of the watermark.

II. Watermarking Technology & Cloud Computing

In the iaas scenario the outsider threat can be mitigated by encrypting the software under keys that helps to coordinate the two collaborating organization. By using the encryption the framework can be secured up to an extent if keys [11, 12] are impossible to guess and if the software is decrypted on a trustworthy platform.

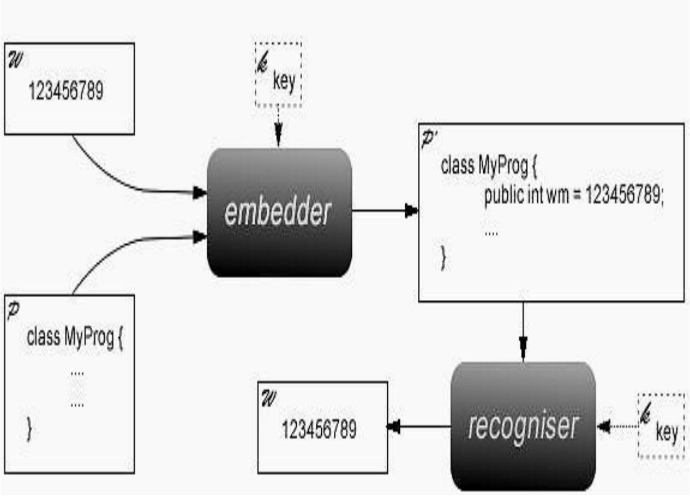

Figure2: code watermarking

There are mainly two types of watermarking techniques [1] when classified broadly, one is static watermarking whose depiction is in figure2 and another one is the dynamic watermarking which needs the runtime information or execution state of a program to embedded a secret key or a piece of secret information. In this figure the code i.e. 123456789 is embedded with a key that can be any key with the help of an algorithm that is called as embedding algorithm like for example using opaque predicates for this scenario. On the reverse end another algorithm i.e. called as detecting algorithm or a recognizer compares the program against the key provided, when the watermark is extracted with correct parameters it means that the user of the software is a legitimate user.

The argument that the cloud is an appropriate location for a trusted user would mitigate insider threat in various software using organizations. The cloud has great powers of observations as well as control to respect the services and infrastructure it provides to client. To be more trusted for sharing software any cloud provider should offer an additional pair of services for embedding and detecting the watermark. According to cloud security alliance the cloud must be from a trusted source irrespective to weather it is audited by a TPA or they take some security measures itself but must be a trusted source or else it will start to Detroit. It is the responsibility to each organization to determine on what aspects a cloud should be trusted.

These are the six question that cloud security alliance ask to each and every cloud service provider

- What will happen if assets are unavailable for a certain period of time?
- What measures will be taken if data is unexpectedly changed?
- What counter actions are taken when any function failed to provide expected results?
- If any outsider manipulate the data then what are the consequences?
- If any insider employee accessed the assets?
- What will happen if assets become highly public or available?

Security problems in cloud have caused significant problems for business and government, the infrastructure resources, platforms as well as services are provided dynamically, on demand provisioning the hardware, software as well as computing or cpu utilization. This makes the normal desktop user gaining massive access to various resources that also includes databases [2]. With the development of the services the cloud risk also increases in parallel. There are some non-technological methods such as education and legal remedies, and also the measures in which the virtualized servers used to limit the damage that is done by some rogue programs [8].

There are three major commercial cloud platforms for security vulnerabilities:

- Google cloud platform also known as Google app engine
- IBM blue cloud
- Amazon elastic compute cloud

TABLE 1: SECURITY THREATS AND COUNTER MEASURES

Threats in cloud		**Counter measures**
Insiders	**Inadvertent abuses**	**Alerts, methods**
	Intended abuses	Encryption, authentication, authorization, digital signature
Secondary users	Unauthorized distribution	Firewalls, network service management, watermarking
Outsiders	Illegal access	Digital signature, software watermarking
Non-human factors	Software and hardware failure	Software management, system vulnerabilities analysis

Four types of disclosure threats and their principal countermeasures are depicted. Non-human factors can also be a type of threat such as failures or bugs in software and hardware. Encryption and watermarking can be used as countermeasures. Here outsider intrusions are defined as unauthorized accesses either through the network or through physical attack. Outsider threat agents are angry customers or vindictive former employees and can also be competitors and thieves. Insiders may inadvertently release data they may be curious they may be bribed or they may be greedy enough to betray the trust placed [13, 14] in them by their organization. Insiders are employees of the software producing organization of the cloud service provider and of the firm who act as security auditors. A second tier of insider represented by the software consuming organization called as secondary users. These semi trusted threat agents and can be controlled by a digital rights management system such the one described here with oversight by external auditors and cloud service providers to lessen the threat of a disclosure abuse that is sanctioned by the software consuming organization [3].

III. The Software Watermarking Model

Let P denote the set of programs that are accepted by a watermarking system, W the set of watermarks for this system and K the set of keys. A watermark is a message of bits expressed by 0 and 1 with a finite length.

Embedding function
$P \times K \times w \rightarrow P^I(1)$

The watermark embedding function Embed is used to insert a watermark into a program. For the program **P** that belongs to the set of programs P, **K** that belongs to the set of keys K whereas the watermark **W** will be taken from the set of watermarks W is so
P^I = embed (P, K, W) this combines to form a watermarked program

Extract function
$P^I \times K \times W \rightarrow W$ (2)
The extract function retrieves the watermark from the watermarked program, this helps in verifying the ownership of the software.

$$\forall P \in \mathrm{P}, \forall K \in \mathrm{K}, \forall W \in \mathrm{W} : Extract(Embed(P,K,W),K) = W$$

A. *Practical problems arising in watermarking*

- ***False negatives:*** these errors can arise due to imperfections in the watermarking embedding and extractions. When encountering the false negatives it is important of specify the test conditions, when a watermarked program is attacked by an intruder the observer's false negative ratio will be much higher than running a software watermarked program on a test bed.
- ***False positives:*** this can be caused by the imperfections in algorithms or the clever ability of the attacker that can mimic the embedding function.
- ***Fail to mark:*** this condition occurs when the embedded program is semantically not equivalent to un-watermarked program. The input is not equivalent to the output where $P \ != \ P^I$

IV. Algorithms For Software Watermarking

There are almost 11 different types of algorithms that help in securing any application, they broadly falls into the category of embedding, blindness, where the extractor is unaware of the original copy of the program whereas in non-blind algorithms the extractor is provided with the original copy of the program in order to compare it with the watermarked copy. In R1 static based algorithm the watermarked is retrieved by examining the watermarked program.

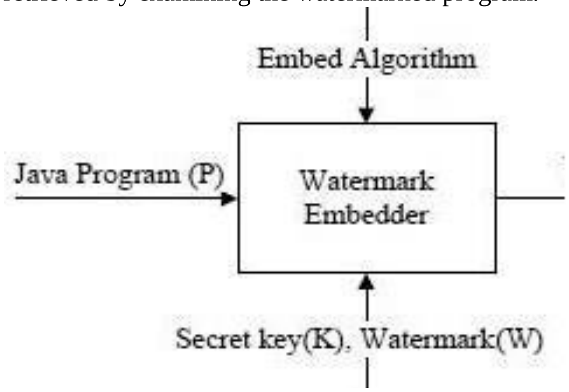

Figure 3: watermark embedding model

This module is somewhat similar to image watermarking or audio/video watermarking, the software watermarking also work in same fashion which has the components like embedding algorithm, secret key and a watermark. Here the example in figure 3 is shown for watermarking java based software and various other applications. In dynamic watermarking by extracting the state of program like memory state, registers when the program is executed with a secret key.In this section we will discuss about two algorithms that will help in securing the code and will provide copyright protection.

A. Graph based watermarking

This algorithm uses the encoding function that converts a watermarking number say W into a graph ***encode (w) -> G***and a decoding function that converts a graph back to the watermarking number ***w, decode (G) -> W.*** this pair is also called as(encode, decode) and along with the graph G. the equation is called as (encode, decode)$_G$ called as graph codec system.

Encoding algorithm:

- Encoding a self-inverting permutation to a directed graph ***encode (SIP -> reducible permutation graph)***
- ***π^* -> F(π^*)*** using the domination
- *Construct a di-graph* ***F(π^*)*** on ***n+2*** nodes using adjacency relation of the nodes.
- For every element I create a vertex and add it in the vertex set V(D[π^*])={v1, v2….v_n}
- Compute the denomination relation of each element i
- For every pair of vertices (Vi, Vj) add the edge in E(D[π^*]) if i denominates j
- Create two dummy vertices S and T and add the set of edges for every in degree and out degree

Decoding algorithm:

- ***Decode (RPG -> SIP)***
- delete the directed edges (Vi+1, Vi) from E(F[π^*]), ***1 <= I <= n***
- *now flipping all the remaining directed edges of the graph* ***F(π^*)***
- *perform a DFS search on a tree T which is constructed by algorithm 1*
- *ordering of nodes s= v0, v1….v_n*
- *start deleting node s from the order* ***π***
- *return* ***π^*= π***

B. Opaque predicate algorithm

This algorithm is based on inserting some predicates to make difficult for adversary to analyze the control flow of any application. There are mainly two methods for opaque predicates GA1 and GA2 [6,7].

The embedding process is dependent on identifying a set of possible branching points. This set is identified through preprocessing each method in the application. For each W_i that belongs to W anopaque predicate is added to a branching point which belongs to the set of branching points. In this technique it is not always possible to identify a local variable that contain integers around that branching point.

To embedded a watermark using GA2 k new methods must be added to the applications. This could increase the size of the program so this is done by using the W_i by rank then reusing the methods to add to the application.

The recognition algorithm varies slightly as the embedding technique. The GA1 technique involves the exhaustive search of the method identifying the set of instructions by creating control flow graph as well as by creating expression trees. Each predicate will end with ***if*** instruction that is encountered as the last instruction of the basic block. The instruction that comprise of the expression tree for that particular ***if*** instruction is compared to the entries in opaque predicate library. If the watermark was embedded using GA2 then each method is scanned looking for invoke instructions which call a method that has the same signature as one of the opaque methods [15].

V. Conclusion

In this paper we have presented the way in which software piracy is increasing due to the non-compliance and low security features present either in the organization or within the application. To protect it in a way that is reasonable and efficient was the biggest challenge. Still we have tried to provide a methodology for enhancing the security of an organization's security measure. In this paper we have counted the types of algorithms available and in what conditions they can be used.The above suggest two algorithms i.e. graph based as well as opaque predicates can be implemented is a distributed environment or can be tested using various prototypes available for simulation of cloud based applications which measures the performanceof the watermarked program with un-watermarked program and then depicts the difference between thetwo. Though watermarking large number of applications is not an issue if the application is small in size, but in case of large applications some special measures must be taken. For simulation in cloud or distributed environment apache hadoop provides its cluster based model whereas java network launch protocol is also a good tool to implement this scenario

Acknowledgment

This work is a combined effort of me and my guide professor shailendrasingh who helped me in this survey work by giving his deep knowledge about the security issue of cloud computing. Without his dedication it wont be possible for me to succeed in this survey

References

[1] C. Collberg and C. Thomborson. Watermarking, tamper-proofing, and obfuscation – tools for software protection. Technical Report 170, Department of Computer Science, University of Auckland.

[2] D. Grover. Program identification. The protection of computer software: its technology and applications, The British Computer Society monographs in informatics.

[3] Business Software Alliance. Seventh annual BSA and IDC global software piracy study, May 2012

[4] Tharaud J, Wohlgemuth S, Echizen I, Sonehara N, Muller G, Lafourcade P. Privacy by data provenance with digital watermarking - a proof-of-concept implementation for medical services with electronic health records. Sixth International Conference on Intelligent Information Hiding andMultimedia Signal Processing (IIH-MSP), 2010; 510–513, doi:10.1109/IIHMSP.2010.130

[5] Hwang K, Kulkareni S, Hu Y. Cloud security with virtualized defense and reputation-based trust mangement. Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC '09), 2009; 717 –722,doi:10.1109/DASC.2009.149.

[6] Collberg C, Thomborson C. Software watermarking: Models and dynamic embeddings. POPL, 1999; 311–324.

[7] Collberg C, Thomborson C. Watermarking, tamper-proofing, and obfuscation — Tools for software protection. IEEE Trans. Software Eng. 2002; 28(8):735–746

[8] Thomborson C. A framework for system security. Handbook of Information and Communication Security, StampM, Stavroulakis P (eds.). Springer, 2010; 3–20, doi:10.1007/978-3-642-04117-4 1

[9] Arboit G. A method for watermarking JAVA programs via opaque predicates. Fifth International Conference on Electronic Commerce Research (ICECR-5), 2002

[10] Myles G, Collberg C. Software watermarking via opaque predicates: Implementation, analysis, and attacks. Electronic Commerce Research 2006; 6(2):155–171.

[11] Ronald L. krutz, "cloud computing fundamentals", what is cloud computing, Indianapolis, Indiana, 2010,ch.1, sec.1, pp.26, 30-32.

[12] AmandeepVerma, SakshiKaul, "cloud computing security issues & challenges: A survey", springerverlag berlin Heidelberg, part IV, ccis 193, pp.445-454, 2011

[13] Michael Arnold, martin schmucker, "applications of digital watermarking", copyright protection, libraryof congress cataloging, isbn 1-58053, ch.3, pp.40.

[14] joshiakshay "enhancing security in cloud computing", information & knowledge management, vol.1,no.1, pp.40-43, 2011

[15] Cyril bazin, jean marie, "a novel framework for watermarking",springer-verlag berlin Heidelberg,pp.201 217, 2008