International Journal of Management, IT & Engineering
Vol. 8 Issue 12, December 2018,
ISSN: 2249-0558 Impact Factor: 7.119
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

ENHANCEMENT IN SECURITY OF FRAMEWORK FOR MOBILE CLOUD COMPUTATIONAL OFFLOADING

Aradhana[1], Samarendra Mohan Ghosh[2]
Department of Computer Science Engineering
Dr. C. V. Raman University Bilaspur, C.G.

Abstract

Code Obfuscation is one of the prevention techniques of computation offloading on clouds and distributed environment. Computational offloading is vulnerable to risks. To enhancing the security of computation offloading, we propose a hybrid approach that renders the byte code and change the structure of original code using code mixing and dead code insertion method. The approach make more hardens the original code, unreadable and difficult to reverse-engineer. Based on our analysis, the obfuscation is able to increase the complexity of the code, confidentiality and integrity during computational offloading on clouds using mobiles. We consider that hybrid approach of obfuscation methods with cryptographic hash function may affect the size of program and cost of execution time.

Keywords: *Code Obfuscation, Encryption, Computation Offloading, Reserve Engineering.*

1. INTRODUCTION

Bytecode obfuscation is method of modifying the byte code or instance of executable file so that it is become a harder to read and understand for an attackers but it remains fully function. At present there are many platform that can be reverse engineering with enough skill hands and efforts like JAVA,.Net and Android. Present available java decompiles can be easily reversed engineered back into source code from complied code, they may very risky for distributing environment.

Bytecode obfuscation provides higher level of ob-security that makes reverse-engineering a program more difficult and economically unfeasible. Other advantages of obfuscation method on distributed environment include helping to protect unauthorized access, code optimizing, hiding vulnerabilities and shrinking the size of the programme code. Protecting intellectual property of any code is a must and not easily achievable; unless the understanding of the intellect present in the system is unidentified. One way of securing the intellectual property is by obfuscating the system. Many types of obfuscation can be done while the func-tionality of the system remains same.

2. OBFUSCATION TECHNIQUES

These are some techniques for java code obfuscation. They can help create a complex defence against reverse engineering and code tampering.

- **Renaming**: this alter the name of class, methods and local and global variables,

identifiers to make the decompiled source much harder for a human to understand.

- **Control Flow Obfuscation**: This method changes the sequence of execution of code and flow of data.
- **String Encryption**: It transforms strings using encryption and only calls their original value when required.
- **Structural obfuscation**: This approach convert common instructions to other form of construct potential for confusing the decompilers.
- **Dummy Code Insertion**: This method inserts null code inside the source code. It breaks decompilers to makes reverse-engineered code harder to analyze and extract the original form of code.
- **Unused Code and Metadata Removal**: This technique reduces the unreachable code and meta data of the source code for preventing the information available to an attacker.
- **Binary Linking/Merging**: It combines many library file and executable lie into one or more than file.
- **Code Mixing**:-In this approach we scrambled the code to harden the reverse engineering.
- **Anti-Tamper**: It infuse the application self protection code into original code verify. If tampering is detected, it can stop the process.
- **Class file Encryption**: This method encrypts the java executable before running confusing decompilers.

Java has an advantage for obfuscation due to its architecture that uses byte code to run on any machine that could run JVM (Java Virtual Machine). Obfuscating the byte code or instance of executable code does not affect the code architecture that will still enable the reusability and maintainability aspects of the system, because only the deliverable element is obfuscated and maintaining the functionality of the system with overhead that depends on the type of obfuscation applied [1]

In this work, we are combined the obfuscation technique on Mobile application codes that involves a small overlaid architecture. We built a tool in Java environment to explore the real-world tradeoffs in transforming and obfuscating Java executable files or byte-codes during the computation offloading using available any mobile devices. For enhancing the security of code two bytecode obfuscation methods is evolved. These techniques involve obfuscation that is dummy code insertion and code mixing transposition. This Tool has considerably more power than other available open source obfuscators on internet. The server platform like clouds can de-obfuscate programs and run on own platform and outputs will back to client's platform.

Successful program obfuscators have a number of benefits:-

1. Protection of transient secrets in programs,
2. License management for networked (e.g., client/server) software,
3. Software-based tamper resistance, and
4. Protection of mobile agents

3. PROBLEM STATEMENT

Security, privacy, integrity and trust issues exist since the evolution of cloud computing. It is big issue to trust between client and cloud such that the cloud provider ensures that the user is an authorized and the user can be assured of data consistency, data storage [2] and the instance he/she is running is not malicious. Hence the

necessity for developing trust models/protocols is demanding. Data are placed in the form of various nodes on different locations on cloud. This causes the security breach thus intrusions are easily affecting the environment. [3].cloud providers do not provide facility of monitoring security, location monitoring, authenticity and integrity of hardware, software and data. [4]. cloud providers do not provide facility of monitoring amount of resources used by customer. In clouds user cannot identify the location of data .so the issued that who controls the integrity of data on virtual machine and difficult to maintain the consistency of security and ensure audit ability of records [5].

This agent is worked like security agent by implementing trusted computing infrastructure through authenticating hardware/software integrity. [6] has analysed some security threats, risks and vulnerabilities associated with cloud computing.

The seven threats identified are:

- Abuse and Nefarious Use of Cloud Computing
- Insecure Application Programming Interfaces
- Malicious Insiders
- Shared Technology Vulnerabilities
- Data Loss/Leakage
- Account, Service
- Traffic Hijacking Unknown Risk Profile

4. PROPOSED FRAMEWORK

The framework provides the secure channel of computational offloading in a fully automatic and end-to-end fashion. The framework will propose a security model for both, the client and server platform during computational offloading. Here any mobile devices are used as client, cloud environment is considered as server environment and resource pool which is under control of the secure infrastructure provider.

In this work, we create an authenticate virtual shadow of mobile application will execute that byte code on cloud. It is helpful in building the trust between client and server communicating with each other via a secure and reliable communication. It produces the result according to request and sends back to client's environment.

This concept is all about to save the battery life time as well as increases the processing capability and storage capability with security. The virtual shadow will be temporary on cloud and will control by mobile device. Once shadow is created it will work only for mobile applications. Cloud can control the request and response of mobile shadow.

Methodology :-

The proposed framework can be partitioned in to three parts:-

1) **Platform Authentication**: - To protect the Client and Server Environment.
2) **Code Integrity and Confidentially**:-To protect the Bytecode which is off-loadable
3) **Sever Side Execution for Computation offloading**:- Secured code will be run on server side and sends back to result on client environments.

The whole procedure of this framework can be performed in the following steps:-

Step 1: Start

Step 2: Mobile Phones are requested to cloud for computation offloading.

Step 3: Cloud accepts the request and sends back response to client and generate virtual shadow for Apps.

Step 4: For secure channel and authorized end parties an ID with crypto-graphic hash function are applied on virtual shadow states.

Step 5: When both ends are found authorized then code offloading process will start.

Platform Authentication

Step 6: All executable code of host platform will wrap up as java class file.

Step 7: Apply Hybrid approach using obfuscation methods to obfuscate the class file for code integrity and confidentially.

Step 8: Compress the obfuscated file.

Step 9: Apply the hash Algorithm on compressed code and generate MD1.

Code Integrity and Confidentially

Step 10: Send the compressed obfuscated code with MD1 to Virtual Shadow of mobile.

Step 11: Received the compressed Obfuscated code with MD1 by Virtual Shadow of mobile.

Step 12: calculate the MD2 from received code

Step 13: compare IF MD1=MD2

Step14: Uncompress the received file.

Step 15: Deobfuscate the obfuscated file and extract original class file.

Step 16: Run the code on virtual shadow in cloud environment

Step 17: Result come back to client Platform.

Else:

Step 18: Cancels the process.

Step 19: **End the Process**

Server Side Execution for Computation Offloading

5. FLOW CHART OF PROPOSED FREMEWORK

The framework propose a security model for both, the Client and Server platform during computational offloading as shown in Fig: 1. Here any mobile devices are used as client, cloud environment is considered as server environment and resource pool which is under control of the secure infrastructure provider.

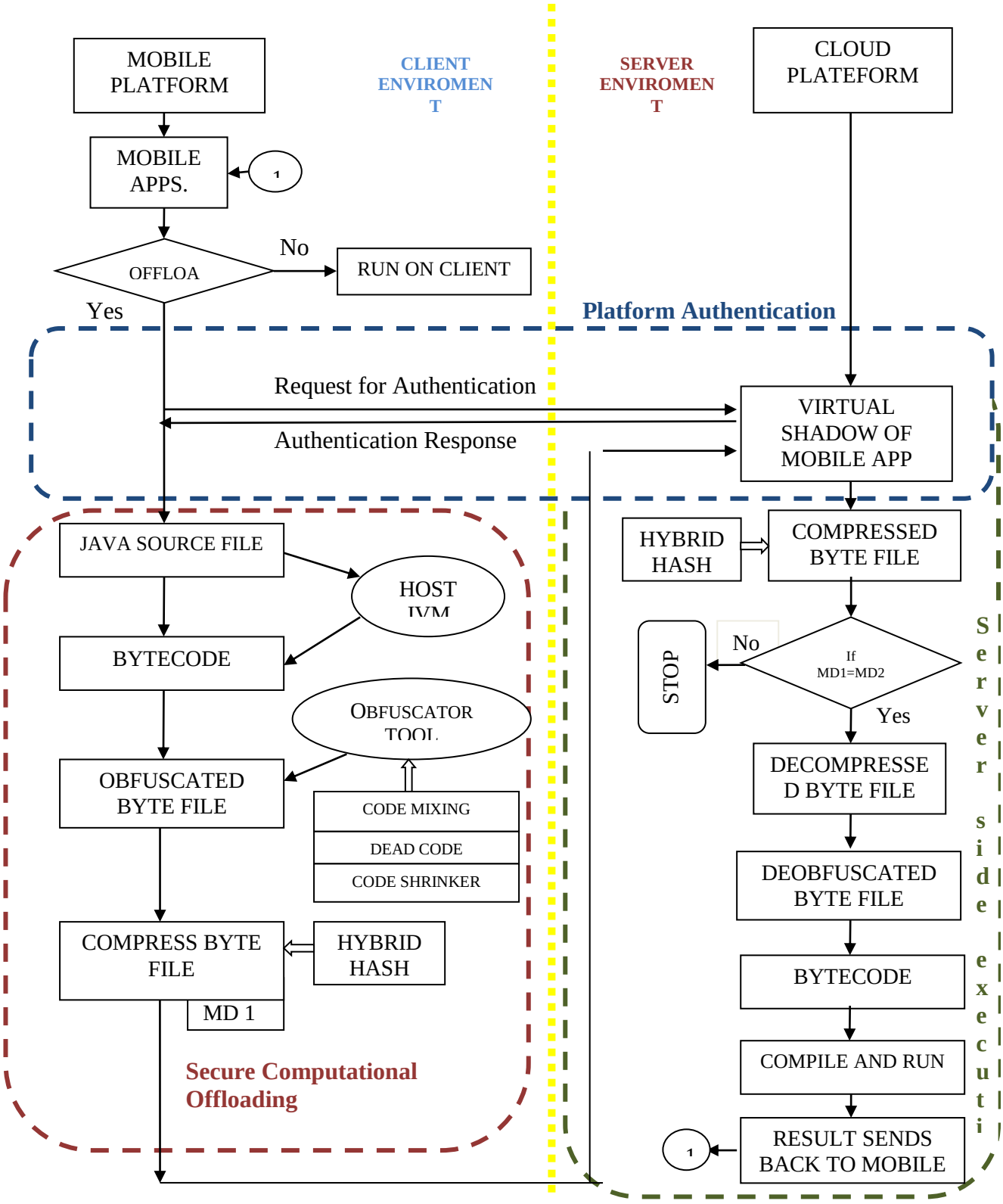

Fig 1: Flow Chart of Proposed Framework

6. FEATURES OF PROPOSED FRAMEWORK

The framework has the following advantages:

- **Ease of use:** virtual shadow of mobile app provides a cross-platform interface to everyone, so that the user may access the virtual phone through the different mobile platforms such as Android, iOS,

or Firefox OS. Which should be easy to use since many people are already familiar with the user interface of smart phones

- **Cost effective**: The system can be easily constructed with available existing devices. It only requires the server supports virtualization technology.

- **High performance:** The framework, the Provides higher computational power and bigger storage capacity in order to execute these heavy weight applications.

- **Flexibility**: Virtualization is a hot topic of computer science. It provides a method to deploy devices dynamically.

- **Intelligent control and management:** The framework will give a way of control and management for mobile application. There is an interface to configure and manipulate the virtual shadow framework on clouds. Illegal accesses can be blocked out automatically by configuring rules and policies.

- **Security and isolation**: Running mobile applications on clouds under the virtual shadow is more secure than executing them on a physical phone because most of malicious activities will be detected immediately in a centralized controlled environment.

7. FEATURES OF PROPOSED TOOL WITH OTHERS OBFUSCATORS

There are many obfuscator tools are available in internet but most the obfuscator are not freely available .We listed the features of two best obfuscator with our proposed tool.

Table: 1 Features of Proposed Obfuscator Tool

Obfuscator Tools	License	Obfuscation	Features
Proguard Tool	**Open Source**	Bytecode Optimizations	• It reduces code the 5% smaller than original. • Up to 20% faster. • Minimal Protection against reverse engineering.
		Flow Obfuscation	
		Incremental Obfuscation	
		Obfuscation using Reserved Keywords	
		Stack Trace Translation	
yGuard Tool	**Free**	Name obfuscation	• Up to 22% approx faster. • Minimal Protection against reverse engineering
		Code Shrinker	
Proposed Obfuscator Tool	**Free**	Code Mixing	• It reduces the size of code. It depends upon the size of file. • Up to minimal faster execution than client. • Enough protection against reverse engineering • It reduces the heavy load on client's battery. • It enhances the processing capabilities. • Detect and Prevent malware and other vulnerable attacks
		Dead Code Insertion	
		Code Shrinker	

8. CONCLUSION

The current mobile environments come with serious secure threats and management issues which require innovative solutions. Inspired by the behaviour of human societies and federalism, this proposed Framework enhances the integrity and security of off-loadable client's code and provides secure platform using mobile agent during end to end offloading of computation on clouds. In this work we implements Advance Cryptographic hash Algorithm with hybrid approach on obfuscation methods for code integrity and confidentially to development of secure environment with virtualization technique, malware detection and informal behavioural of monitoring.

9. REFERENCES

I. *Vasudevan M. et al, An Architecture of Class Loader System in Java Bytecode Obfuscation, Indian Journal of Science and Technology, Vol 8(S2), 291–294, January 201, ISSN (Print) : 0974-6846 5*

II. *Cachin, C., Keidar, I., and Shraer , A. Trusting the cloud. ACM SIGACT News, 20:4 (2009), pp. 81- 86.*

III. *Popovic, Kresimir, Hocenski, Zeljko, "Cloud computing security issues and challenges", MIPRO, 2010 Proceedings of the 33rd International Convention, pp.344-349, 24-28 May 2010.*

IV. *Grobauer, B., Walloschek, T., Stocker, E., "Understanding Cloud Computing Vulnerabilities", Security & Privacy, IEEE, vol.9, no.2, pp.50-57, March-April 2011.*

V. *Anthony Bisong and Syed (Shawon) M. Rahman, "An Overview of the Security Concerns in Enterprise Cloud Compuitng", International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.1, January 2011.*

VI. *Amit Sangroya, Saurabh Kumar, Jaideep Dhok, Vasudeva Varma, "Towards Analyzing Data Security Risks in Cloud Computing Environments". International Conference on*

Information Systems, Technology, and Management (ICISTM 2010)

VII. *Madhulika, G. and C.S. Rao. Generating digital signature using DNA coding. in Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014. 2015. Springer.*

VIII. *Baratto, R. A. B., Potter, S., Su, G., & Nieh, J. (2004). MobiDesk: Mobile virtual desktop computing. In Proceedings of the 10th annual international conference on mobile computing and networking (pp. 1–15).*