Software Obfuscation: A Method for Cryptography

Varun Agrawal, Dharmesh Aghada, Lalitkumar Agarwal, Megha Mudholkar

Abstract— Software Obfuscation is the method for cryptography. Software obfuscation is motivated by the idea that many useful programs can become beneficiary if we can somehow stop people often called intruders from reading the entire software programs, while still letting them possess and run the code on their own computers. It is used to encrypt the source code which makes it hard to understand by the normal person. It is difficult for the programmer to read and write obfuscated code.This paper presents overview, analysis, working and applications of software Obfuscation.

Index Terms— Code Encryption, Cryptography, intruder, Software Obfuscation, Software Protection.

1 INTRODUCTION

IN this increasingly competitive world, there is a threat for the software manufacturers which leads them to create various software protection techniques. Many software protection techniques are available amongst which one well known software technique used is software obfuscation.
Software obfuscation in layman language is the method of making software program codemore difficult to understand. This can be done by encrypting the code so that the other people cannot identify the execution flow of the program and still they can run the code and execute the programnot only on their own computer but also on other machines.
In software obfuscation, we can encrypt the important part of the code temporarily, but we need to send the decryption keys with the program so that it can be run by the other user. There is a tool known as obfuscator which is use to automatically convert the source code into an executable program which runs in the same manner, but it creates the problem for the programmer to read and understand it.

2 LITERATURE REVIEW

According to Aniket Kulkarni in research paper Software Protection through Code Obfuscation [1]. There are various software protection techniques like Code obfuscation, Code watermarking and tamper-proofing. Code obfuscation is a technique used to protect software against malicious reverse engineering attacks. Code watermarking is a technique used to protect software against software piracy attacks. Tamper-proofing is a technique used to protect software against tampering attacks.Classification of code obfuscation techniques are layout obfuscation, control obfuscation, data obfuscation and preventive obfuscation. He has also described the various technique which is use to get the original source code. Attackers can employ these techniques to get the functionality of the software. These techniques are Static analysis, Dynamic analysis, and Code Clone Detection techniques.

According to Chandan Kumar Beheraa and D. Lalitha Bhaskari in their studied research paper Different Obfuscation Techniques for Code Protection [2], they have stated that, Obfuscation consists of code transformations that make a program more difficult to understand by changing its structure, while preserving the original functionalities, not suitable also to reverse-engineering. Encryption and firewalls are some of the common solution to diminish the threat of the attackers who try to crack the application. But, these approaches do not help to protect the software, when the attacker is him/ herself the end-user. Among the various techniques available for protecting code from different attacks, code obfuscation is one of the most popular alternative, for preventing from code comprehension, code tampering etc. So, code obfuscation is a largely adopted solution, and many different obfuscation approaches has been proposed. This is also a type of software protection against unauthorized reverse-engineering. Practically, protection by server-side, hardware-based security solutions, different signed native codes, tamper proofing, watermarking, software aging, packing are some of the most commonly used methods to avoid or challenge the detection engines. However, the provider should estimate how long the obfuscation would resist, i.e., the time taken by an attacker to understand the code. According to that, some other obfuscation methods can be implemented on the original code, so that; the adversary will not be able to get the algorithm or logic of the code.

Further, obfuscation methods include code re-ordering, trans-

- *Varun Agrawal is currently pursuing masters degree program in Computer Application in Mumbai University, Mumbai, India,. E-mail: varun_agrawal1995@yahoo.com*
- *Dharmesh Aghada is currently pursuing masters degree program in Computer Application in Mumbai University, Mumbai, India,. E-mail:. dharmesh.aghada@gmail.com*
- *Lalit Agrawal is currently pursuing masters degree program in Computer Application in Mumbai University, Mumbai, India,. E-mail: lmkagarwal1234@gmail.com*
- *Megha Mudholkar is working as assistant professor in TIM-SCDR,Mumbai,India. E-mail:meghakunte2000@gmail.com*

formation to replace meaningful identifier names in the original code with meaningless random names (identifier renaming), junk code insertions, unconditional jumps, conditional jumps, transparent branch insertion, variable reassigning, random dead code, merge local integers, string encoding, generation of bogus middle level code, suppression of constants, meshing of control flows and many more. Basically, obfuscation is different from encryption in many ways. Primarily, it does not require any inverse transformation. Next, it's not necessary for an attacker to look for the original code all the time, because the attack can be succeeded without having the original code of the software. And at last, cipher text will be worthless without the key, as an obfuscated program can perform without any additional information.

According to N. Sasirekha and Dr. M.Hemalatha in their research paper, a survey on Software Protection techniques against various attacks had evaluated [3]. Software security and protection plays an important role in software engineering. Considerable attempts have been made to enhance the security of the computer systems because of various available software piracy and virus attacks. Preventing attacks of software will have a huge influence on economic development. It is very important to develop approaches that protect software from threats. Digital data of the software is very risky task. Confidentiality and data authenticity are two important concepts in security. Piracy, reverse engineering and tampering have been the major software threats. The achievement of the content/ software security in a huge segment is based on the ability of protecting software code against tampering and identifying the attackers who issue the pirate copies. Code obfuscation focuses to protect code against both static and dynamic study and there exists another approach to protect against code analysis, namely self-modifying code.

In this paper the author has emphasized more on how to protect the software by using these three techniques like Software watermarking, code obfuscation, and tamper proof. These three techniques we have seen in the previous author's paper also. It means that these three techniques are the basic methods to protect the code. But in future these techniques will not be safe or will not be sufficient enough to protect the code as they are doing in recent days.

3 ANALYSIS

If other person tries to copy the code so they won't be able to read or modify the code. Software Obfuscation helps in reduced the size of the software. Software Obfuscation also helps to increase the performance of the application. A variety of tools exist to perform or assist with code obfuscation. These include experimental research tools created by academics, hobbyist tools, commercial products written by professionals, and open-source software. There also exist deobfuscation tools that attempt to perform the reverse transformation. Although the majority of commercial obfuscation solutions work by transforming either program source code, or platform-independent byte code as used by Java and .NET there are also some that work directly on compiled binaries.

4 WORKING

Code obfuscators work by changing the names and (now and again) association of techniques without modifying the conduct. This works a ton like refactoring instruments, aside from while refactoring devices for the most part live in the IDE, obfuscators infuse into the assemble procedure

The most straightforward type of muddling is renaming an all-around named technique or capacity to a seriously named strategy or capacity. For example, suppose a function was called GetNameFromTable. This might be renamed to gn_1ftl.

The fact of the matter is to deliver a capacity whose name provides no insight as to its motivation. Different changes may include changing the names of parameters, including additional parameters that have no reason, pointers-to-pointers and some other things which would confound somebody taking a gander at the code. Think about every one of the things great software engineers don't do in light of the fact that it prompts to ineffectively organized and mixed up code - that is the thing that obfuscators do.

5 TECHNOLOGY

The main aim of software obfuscation is to provide reverse engineering to gain access to source code. When we are doing software obfuscation we should keep in mind that it should not change the software performance. Software obfuscation works by converting the code which humans and decompilers use to understand the functionality of the programs.

Software obfuscation is the process which transforms the source to make it more difficult to be analyzed. This process is implemented in automatic tools called obfuscators. Following obfuscation techniques must be considered while doing this: name obfuscation, data obfuscation, code flow obfuscation, incremental obfuscation, intermediate code optimization, debug information obfuscation, watermarking, and source code obfuscation.

6 APPLICATIONS

There are lots of obfuscation software's available in market.Some of the popular applications of obfuscation used in high level languages areas follow:
For .Net :
1) Smartassembly
- References Dynamic Proxy

- Control Flow Obfuscation
- Dependencies Merging

2) Dotfuscator
- Cross Assembly Renaming
- Silverlight XAML Renaming
- Enhanced Overload Induction

3) Salamander
- Licensing Management Support
-Version Independency
- Resource Protection

For Java :

1) Pro Gaurd
- Shrink
- Optimize
- Preverify

2) yGaurd
-Name Obfuscation
-Code Shrinking

3) Allatori Java Obfuscator
- Minimizes Application Size
- Boosts the Speed

7 Diagramatic Representation

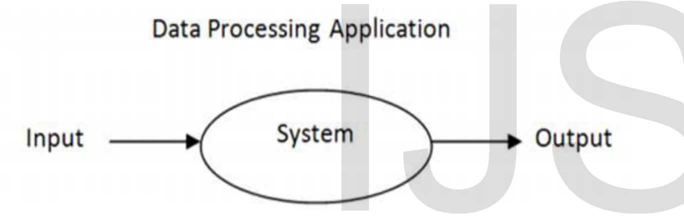
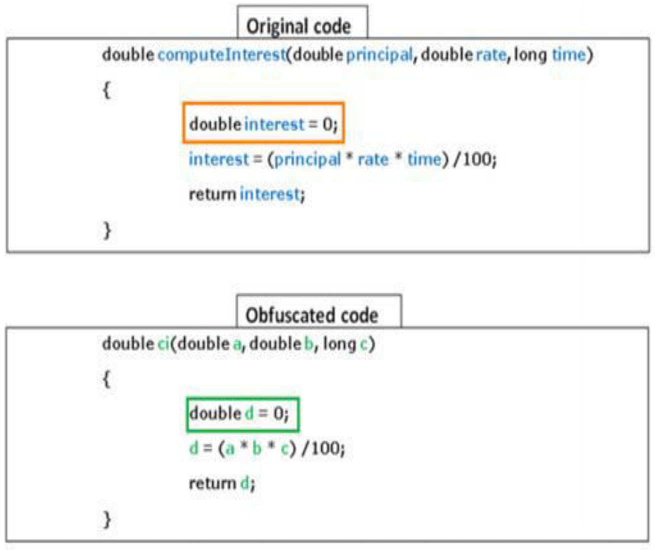


Fig.1. Simple Flow of Code



Original code

```
double computeInterest(double principal, double rate, long time)
{
        double interest = 0;
        interest = (principal * rate * time) /100;
        return interest;
}
```

Obfuscated code

```
double ci(double a, double b, long c)
{
        double d = 0;
        d = (a * b * c) /100;
        return d;
}
```

Fig. 2.Original code and code after Obfuscation

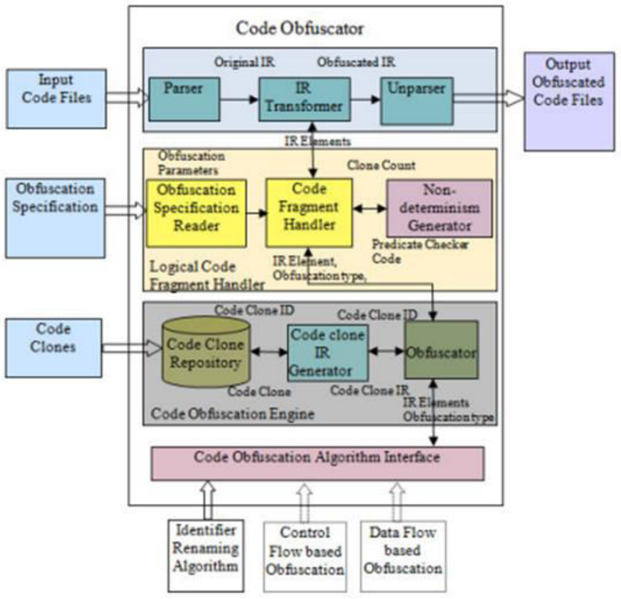

Code Obfuscator
Input Code Files
Original IR
Obfuscated IR
Parser
IR Transfomer
Unparser
Output Obfuscated Code Files
IR Elements
Obfuscation Parameters
Clone Count
Obfuscation Specification
Obfuscation Specification Reader
Code Fragment Handler
Non-determinism Generator
Predicate Checker Code
Logical Code Fragment Handler
IR Element, Obfuscation type,
Code Clone ID
Code Clone ID
Code Clones
Code Clone Repository
Code clone IR Generator
Obfuscator
Code Clone
Code Clone IR
IR Elements Obfuscation type
Code Obfuscation Engine
Code Obfuscation Algorithm Interface
Identifier Renaming Algorithm
Control Flow based Obfuscation
Data Flow based Obfuscation

Fig.3. Block Diagram of Obfuscated Code

7 Future Enhancement

It will take time to make understand all the developers to obfuscate the code after it's done. In some years IDE companies will provide obfuscation feature in-build.It's little time consuming, but it's definitely worth it. This will decrease software cracking and make de-obfuscation hard for reverse engineers.

Techniques we can see and use in future for software obfuscation is dead code insertion, subroutine reordering, instruction substitution, code integration and code transportation.

We have seen from the previous author that we use to protect the code by using three basic techniques like Software watermarking, code obfuscation, and tamper proof. But we cannot rely on these techniques for more long. We need to find the more secure way to maintain the security of the code. We need to create the most secure algorithm so which can be use for the software obfuscation.

8 Conclusion

Many Companies are trying their best to create a future where software developers don't have to struggle.Each of the above techniques and application used above are powerful and effective to prevent the crackers and hackers. If these applications are used together for obfuscation it will be extremely difficult to break the code. If the codes are broken by the expert they will get garbage, encrypted or obfuscated code, names and data as the result. The GNU website states "Obfuscated 'source code' is not real source code and does not count as source code." Thus it's not only a practice of protecting secret information from intruders but also to reserve intellectual property rights from outside environment.

ACKNOWLEDGMENT

It gives us immense pleasure to present this Paper. We grab this opportunity to express our heartfelt obligation towards the people without whom completion of this Paper have not been possible. We would like to thank our guide Prof. Megha Mudholkar for her immense support and continuous encouragement to our team.

REFERENCES

[1] http://www.coep.org.in/page_assets/341/121003016.pdf
[2] http://www.sciencedirect.com/science/article/pii/S1877050915032780
[3] http://computerresearch.org/index.php/computer/article/viewFile/431/431
[4] https://www.researchgate.net/publication/29681808_Software_Obfuscation_on_a_Theoretical_Basis_and_Its_Implementation
[5] https://www.iacr.org/archive/crypto2001/21390001.pdf
[6] http://www.cs.jhu.edu/~susan/papers/HRSV07.pdf
[7] https://blog.cryptographyengineering.com/2014/02/21/cryptographic-obfuscation-and/
[8] http://searchsoftwarequality.techtarget.com/definition/obfuscation
[9] http://softwareengineering.stackexchange.com/questions/129296/the-case-for-code-obfuscation
[10] http://profs.sci.univr.it/~giaco/download/Watermarking-Obfuscation/SW%20obfuscation%20in%20security%20(survey).pdf
[11] http://www.sciencedirect.com/science/article/pii/S1877050915032780