Bulletin of Networking, Computing, Systems, and Software – www.bncss.org, ISSN 2186-5140
Volume 7, Number 1, pages 50–52, January 2018

Program Obfuscation Method using Entropy

Takuto Omine, Chikatoshi Yamada, Kei Miyagi
National Institute of Technology, Okinawa College
905 Henoko, Nago-shi, Okinawa
{ac164602@edu., cyamada@, k.miyagi@}okinawa-ct.ac.jp

Shuichi Ichikawa
Toyohashi University of Technology
1-1 Hibarigaoka, Tempaku-cho, Toyohashi
ichikawa@tut.jp

***Abstract*—In this research, we aim to improve a program obfuscation method. Protection of program obfuscation is decreased by using various hacking tools such as disassemble, decompilation and hex editor, etc. A previous research provided obfuscation of control structure by dummy blocks and sentence cross-over in the structure. We reveal a method of previous research using measurement of entropy and Kolmogorov. Therefore, we control dummy blocks by pre-calculation of the entropy. As a result, we consider protection enhancement of our proposed method.**

I. Introduction

It is used which software package of electronic trading by utilizing the internet. However, we need using Software Protection. the reason is because we need it to counter Program analysis program and software illegal copy[1]. The following can be given as Software protection, obfuscation, encryption, divergence, digital watermark and tamper resistance, etc. However, the following exist as countermeasure, disassemble, de-compilation, hex editor, memory dump, monitoring tool. Therefore, Improving the functioning of Software protection is require[2][3]. In this research, we aim to improve a program obfuscation method. It is applied if statement and while statement and dummy block which Mr. Tubouti's research [4] using obfuscation of control structure. Proposal method aim at improving entropy by operating based on prior probability to dummy block of generating method.

II. Related works

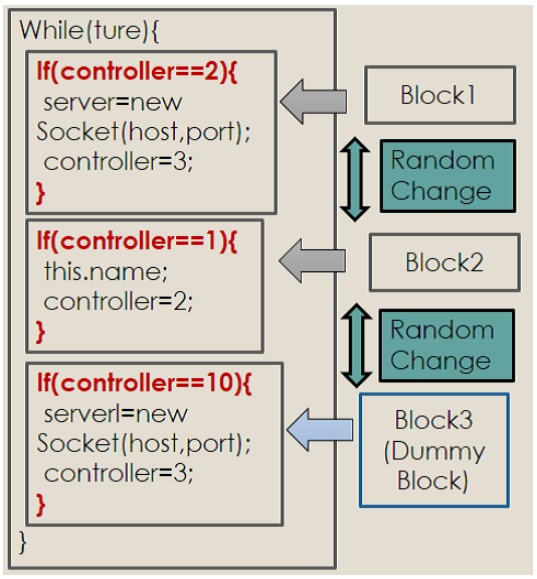
A. Obfuscation method of control structure

Prior method has obfuscation and digital watermark by packaging method using dummy block and if statement to part of program[4]. The following indicate prior method of detail.

First, prior method separate program to a part called "block" and use it. Moreover, the program is executed by controllable parameter called controller. The example of separating program is shown in Fig. 1.

Next, Controller vary hash value, And block shift other blocks in a random order. In this way, it is possible to obtain the obfuscation hardly decrypting the whole of data, because serializability is collapse. In addition, if malicious attacker does 冲 know true key, he should use all combinations.

Finally, it is added dummy block in Aggregate of Block. Dummy block is copy from Aggregate of Block. Then, dummy block changes a little, and can't execute if program is run by true key. The example is shown in Fig. 1.



```
While(ture){
 If(controller==2){
  server=new
 Socket(host,port);
  controller=3;
 }
 If(controller==1){
  this.name;
  controller=2;
 }
 If(controller==10){
  serverl=new
 Socket(host,port);
  controller=3;
 }
}
```

Fig. 1. Obfuscation method of control structure

This figure complicate analysis by adding pattern to round robin analysis and sequential execution analysis which process observe by running program frequently halt on virtual computer.

B. Evaluation system using noise

In this paper, control structure obfuscation quantify to be index by which evaluation system use it for obfuscation by Futamura[5]. Noise is large in program, program trait is small from deviation of order. From that point of view, Evaluation system adopt concept of noise and define it as tow numerical expressions of information theory. Evaluation use assemble program. At this time, It is indecipherable status that malicious attacker shows noise. And, evaluation is executed. It is state that program is noise as that each order randomly emerge in program. Then, this state can signify frequency of appearance of each order and regularity of line. Therefore, it betoken two numerical expressions, as shown bellow.

1) Entropy: Expression of frequency of appearance of each order indicates equation1.

$$H(S) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{1}$$

p_1 is information source and signify frequency of each order. At this time, The base of a logarithm is 2, because frequency of each order define half .

2) Kolmogorov complexity: Regularity of line indicates equation2.

$$k(p) = \frac{K(p)}{l(p)} \qquad (2)$$

$K(p)$ is size of program after compression. $l(p)$ is size of program before compression.

III. Proposed Method

We innovate Entropy and Kolmogorov complexity from chapter II-B for which we quantify program of noise. In this paper, we expressly focus entropy of maximization.

A. Voluntary generation of dummy block

probability distribution of order be equal by Murayama that program quantify small order and implicate structure[6]. However, We think It is subject to result by structure of program. Then, It isn't impaired execution speed of obfuscation of control structure using IF statement and While statement, because this method only execute necessary block. Thereat, certain entropy aim improvement at proposal method by that dummy block generation is option. Summary are shown bellow.

1) Summary of proposed method: Base of program separate into block, as shown bellow. Frequency of each blocks is two-third of substitution and one-third of addition.

At this time, previous method insert block of substitution or addition to program. On the other hand, proposal method insert block of addition, as shown bellow. Therefore, because frequency of each blocks be similar it, entropy rise.

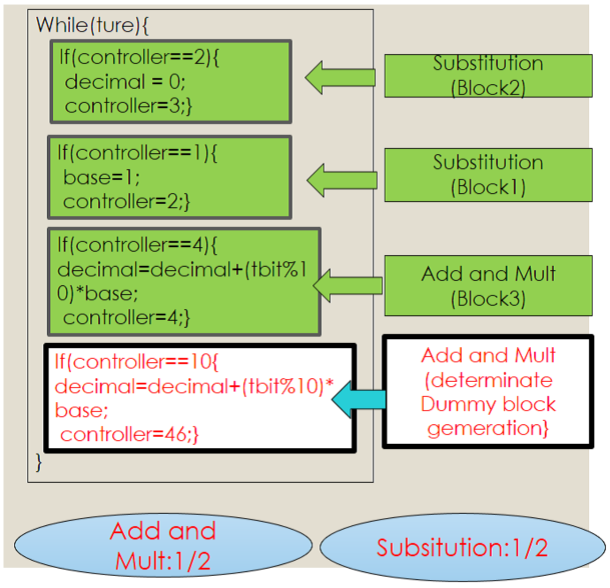


Fig. 2. A Proposed method

B. Proposed method of process

It is process from chapter III-A1 that proposal method insert into program, as shown bellow.

1) control structure is made by while statement and if statement.
2) control argument controller will be hash value.
3) probability of order is cast before dummy block insertion.
4) Dummy block is generated in accordance with 3) result. for that small block fill up and probability of order make uniform.
5) Blocks are changed.

C. Measurement of proposed method against prior method

In this paper, we measure proposal method against prior method by Simply simulation using C programming language. table.1, 2, and 3 shown bellow for result. hyou

TABLE I
SIMULATION ENVIRONMENT

OS	windows 7 Enterprise
Using PC	Dynabook R730/E730/E26BR
System type	32bit operating system
Memory	4GB
Processor	Intel(R) Core(TM) i3 CPU M380 2.53GHz
Program language	C
Compile tool	MinGW32(gcc version 5.3.0)

TABLE II
VALUATION REQUIREMENT

program	100,000
order type	100
dummy block generation	50,000
dummy block type	50

TABLE III
RESULT OF SIMULATION

	entropy H(S)
program	4.889
prior method	5.588
proposal method	5.933
maximal value	6.643

It is observed 0.345 improvement to measure proposal method against prior method.

IV. Conclusion

In this study, we aimed to improve a program obfuscation method. It was applied if statement and while statement and dummy block which Mr.Tubouti's research using obfuscation of control structure. Proposal method aimed at improving entropy by operating based on prior probability to dummy block of generating method. As a result, we could observe 0.345 improvement by simply simulation.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP17K01063.

References

[1] Christian S. Colberg, Member, IEEE Computer Society, and Clark Thomborson, Senior Member, IEEE, " Watermarking, Tmper-Proofing, and Obfuscation-Tools for Software Protection ", IEEE Transactions on Software Engineering, Vol.28, No.8, August 2002

[2] Aikito Monden, Clark Thomborson, " Recent Software Protection Techniques - Software-only Tamper Prevention - ", IPSJ Magazine Vol.46 No.4 Apr. 2005

[3] Aikito Monden, Clark Thomborson, " Recent Software Protection Techniques - Hardware-assisted Tamper Prevention - ", IPSJ Magazine Vol.46 No.5 May. 2005

[4] Masayuki Tsubouchi, Yasuo Okabe, " Fingerprinting in Executable Files for Tracking Illegal Uploaders ", IEICE Technical Report SITE2013-74 IA2013-99(2014-02)

[5] Ami Futamura, Ami Futamura, Akito Monden, Haruki Tamada, Yuichiro Kanzaki, Masahide Nakamura, Kenichi Matsumoto, " Evalution of Software Unserstandability Based on the Randomness of Instructions ", FOSE2012

[6] Takanori Murayama, Masahiro Mambo, Eiji Okamoto, Tomohiko Uyematsu, " How to make a software program hard to understand ", SCIS96-8D, January 1996