Mobile Information Systems 6 (2010) 85–105
DOI 10.3233/MIS-2010-0094
IOS Press

The design and implementation of tamper resistance for mobile game service

Hang Bae Chang[a,*], Hyuk Jun Kwon[a] and Jong Gu Kang[b]
[a]*Daejin University, San 11-1, Sundan-Dong, Gyeonggi-Do, 487-711, Korea*
[b]*Yonsei University, New Millenium Hall, 262 Seongsanno, Seodaemun-Gu, Seoul, 120-749, Korea*

Abstract. The commensurate number of the attacks and infringement targeting a vulnerability of the game service has been increasing constantly, due to the dramatic growth and expansion of the impact of the game industry. However, there exist no subsequent researches for the differentiated technology, which is to prevent the reverse function of the game service. Therefore, in this study, we examined the current status of infringement toward online game services which are provided in the market currently and designed the proper technical measures ('Software Tamper Resistance') for a manipulation of the game service which is the most vulnerable part. In detail, we have encrypted an execution file and decrypted it in real time process. After that we implemented antibebugging, disassemble, and antidump technology.

Keywords: Mobile game security, software tamper resistance, debugging, disassemble, memory dump

1. Research background

According to related data, size of Korean game service industry is estimated as the approximately 7.2 billion dollars in 2006 and it seems to increase at the rate of 10% annually. Such a rapidly increasing game service industry features few characteristics such as below. Firstly, game service possesses cultural characteristic. Game service includes 4 elements such as language, illustration, sound, and movement, and the technology, which is integrated with these elements can be considered comprehensive culture. Secondly, game service reflects an industrial characteristic [8]. Game service industry has grown rapidly and is highly value added, featuring the very high spirit of adventure. Therefore, this industry is more suited to an advanced country where effort to develop quality of life and solve economic problems at the same time exists than a developing country where capability of country is concentrated on solving economic problems. Thirdly, game service reflects a social characteristic. Game service has been showing a strong appetence accompanying addiction, because it possesses a strong power of attracting service users to be eager to want. Lastly, game service has learning characteristic. Game service can cultivate sensitivity, which is necessary for work and learning concerning game and contribute to improving individual ability and imagination by furnishing a chance to expand one's creativity.

However, considering influence of game service on economic-social-cultural issues and a speed of advancement of this industry, there exists lack of systemic and differentiated research on game service, which may mitigate reverse function of game service, in comparison with research on information security technology regarding general information system [16]. Therefore the number of game eservice

*Corresponding author. Tel.: +82 31 539 1752; Fax: +82 31 539 1750; E-mail: hbchang@daejin.ac.kr.

infringement accidents has been gradually increasing and those accidents may be enlarged to various damages such service break, leakage of game service source and etc [13]. In this study, we plan to design technical countermeasures to develop information security technology featured for game service via defining infringement types and analysis of infringement cause for each type by investigating current status of mobile game infringement.

2. Game service infringement survey

Due to an increase of the number of game service infringement accidents, a type of infringement tends to be various and extended to social crimes. In this study, we have investigated specific infringement cases (duplication of game items, double log in, increase of attack and movement speed, modification of attack/defense score) of Korean mobile game service and state of establishment of countermeasure. To carry out investigating state of game service infringement, we have designed questionnaires based on proceeding studies on investigation of information security status and level. We have asked professionals who are related with this field to review questionnaire and executed pre-test survey to maximize validity of survey. The investigation into the actual condition of reverse function of information, which has been conducted by Korea Information Security Agency annually, was carried out to accumulate objective data for the purpose of establishing policy alternatives to prevent reverse function of information. This survey have investigated users' experience of infringement, attitude and perception, prevention and countermeasure for 6 different categories (infringement of personal information and privacy, spam mail, harmful information distribution, computer virus, illegal access to information system and sabotage/hacking of information system, and wireless internet security). The investigation into the actual condition of reverse function of SMEs' information, which has been conducted by Korea Technology and Information Protection Agency for SMEs investigated status of information security workforce and organization, establishment and compliance of guideline, status of investment on information security and management, experience of security incident and countermeasure, problems caused by information security promotion and demand for support from authority.

These preliminary researches have limitation of understanding cause of reverse function of information and organizing infringement type as to information service. Furthermore those researches have targeted only service users not service providers and have possessed disadvantage that questionnaire had too many items to be answered to collect many kinds of data at once.

The status investigation categories for this study was designed basically in accordance with proceeding researches and to investigate the analysis of game service infringement cause separately. The questionnaire has consisted of less than 20 survey questions for game service users, developer, and administrator. The survey questions for administrator and developer of game service was to ask state of establishment of countermeasure for game service infringement and the survey questions for service users was to ask actual state of game service infringement that users actually feel about.

The research objects have been selected as 80 games service providers that have a number of game service users more than certain level and 350 of their users who are older than university students based on the contents of 2005 White Paper on Korean Games.

An indirect survey (telephone, email, fax, and et.) and interview was used for survey method for 30 days (1st Aug 2006 $\sim$ 30th Aug 2006) and survey analysis was conducted with valid questionnaires collected. Finally the number of collected valid questionnaires was 50 answered by developer and administrator of game service, 200 answered by game service users. The kinds of investigated game

Table 1
Status of infringement for respective type of game service

	Role playing game	Casual game	Board game	Shooting game
Server	70.0%	72.0%	62.5%	0.0%
Network	20.0%	20.0%	25.5%	33.3%
Client	10.0%	8.0%	12.5%	66.7%

services were roll playing game, casual game, web board game, shooting game, and etc. To validate the collected questionnaires, we measured reliability by utilizing statistical method.

The reliability indicates how often similar results are shown under similar condition of test, which is being repeated. In this study, we analyzed the reliability by running SPSS 12.0 with survey results, applying internal consistency examination method, which deduces reliability through Cronbach Alpha.

The reliability level is evaluated by Cronbach Alpha with internal consistency examination method and according to preliminary studies, it is explained as enough if Cronbach Alpha is higher than 0.6 (higher than 0.8 for basic science field and higher than 0.9 for important decision making). Prior to reliability analysis, we organized analysis direction after reviewing a general distribution pattern of collected data. Generally if the analysis results of collected data show pattern of normal distribution, it will possess more accurate result than data with other distribution patterns. Hence, the result of examination of normal distribution for collected data, we have obtained the result that is close to normal distribution. The Cronbach Alpha value for the questionnaire items asking developer (administrator) about status of countermeasure (perception of importance of countermeasure for infringement accident, construction of proper environment) for infringement accident was calculated as 0.7223 and this value indicated that there was not any statistical problem with analyzing survey results.

The survey results indicated that 70% (35 people) of developer (administrator) of game service and 38.5% (77 people) of users of game service have experienced infringements of game service. The reason the number of respondents game service users who have experienced infringement accidents is smaller than that of developer (administrator) of game service might be that users couldn't have even recognized a certain experience is infringement accident due to lack of professional background knowledge. As the result of survey also showed that client area was the place where infringement cases had occurred most frequently amongst infringement areas of game service and this fact means that a general development direction for information security technology that is focused on network and server is opposed to the current status of infringement accident for game service. The reason infringement cases occur frequently in client area is due to characteristics of game service users' who utilize infringement tool to use game service easier, while infringements against general information system are to obtain certain information or disturb an operation of the system by attack the system.

The status of construction of information security system and status of establishment of countermeasure are relatively low in comparison with status of recognition of importance of information security for game service infringement. The reasons that construction of environment for information security is insufficient are due to the absence of task team and information security management system, which can block game service infringement occurring present.

3. Game service infringement classification

To conduct an investigation into actual infringement cases of game service based on the investigation of status of game service infringement, the types of game service infringement should be classified

in advance [6]. In this study, we would like to define types of game service infringement, which are appropriate for game service field by analyzing characteristics of infringement against game service, considering classification system of infringement of general information security.

The initial researches on the classification system of types of information security infringement have been conducted by Bisbey (1978) and Brain Matick (1990), yet those researches were little indistinct and arranged infringement types focusing on vulnerability of information assets rather than external attacks. Bishop's study (1995) has classified happenable vulnerabilities under Unix environment in accordance with 6 attributes axe (Nature, Time of Introduction, Exploitation Domain, Effect Domain, Minimum Number, Source), breaking from tree-like taxonomy that were previously used. Howard (1997) defined each steps (Attackers, Tools, Access, Results, Objectives) of external attackers process for information assets and arranged types of infringement according to characteristics of each step. Lough (2002) classified types of infringement according to characteristics of external attack (Improper validation, Improper exposure, Improper randomness, Improper deallocation). Ray Hunt (2005) analyzed types of external attack and divided types of infringement by 4 dimensions (Attack Vector, Attack Target, Vulnerabilities and Exploits, Attacks Payloads or Effects).

In this study, we would like to classify types of game service infringement with first and second classification criteria (Attack Vector, Attack Target) among 4 dimensions, because researches on analysis of specific cause of game service infringement just have stayed at initial stage. The classification method of infringement type that conforms to the first criteria is carried out according to infringement method. For instance, it is divided into virus, worm, denial of service attack, password attack, and etc. in the first step and it is specifically divided into general file infection, core information of system infection, macro infection, and etc. Denial of service attack is divided into host/network-based. The classification method of infringement type that conforms to the second criteria is carried out according to infringement object. For example, infringement object is divided into hardware and software in the first step. Hardware is divided into computer system and Software is divided into operation system, application software, and network in the second step. In the third step, computer system is specifically divided into hard disk, network equipment, peripheral device, and etc. and operation system is divided into Windows, Unix, Mac OS, and application program is divided into server, user, and network protocol, and etc. Likewise methodology of Ray Hunt (2005) carries out classification of infringement type continuously by stepwise refinement based on determined criteria. In this study, we have conducted classification of infringement cases according to infringement method and then classified cases based on infringement object. We have classified cases as vulnerability, which is caused by game service's own vulnerability, attack, which is caused by external attack, and social engineering, which is cause by people or external environment.

To design specific system of game service infringement type, we organized collected cases and preliminary studies and finally arranged infringement types using Delphi Methodology with professional group. Delphi method is that the repetitive process of taking advice for statistical analysis from professionals. This method provides professionals with chances to modify their opinions and to share others' opinion. Currently this method is prevailed in the field of technology forecasting research. It also gives a chance to guarantee reliability by participation of professional group.

We have conducted surveys 3 times, the first one was carried out from 11th Oct 2006 to 13th Oct 2006, the second one was executed from 18th Sep 2006 to 20th Sep 2006, and the last one was conducted 1st Oct 2006 to 10th Oct 2006. The infringement types were designed based on previously explained methodology and questionnaires were sent via email and fax. The reviewers who belong to professional group were 3 professors who are studying information security and 2 managers who are supervising works related to information security. The return rate of survey was completely 100%, because participants of

Table 2
Classification of game service infringement type based on method and object

Infringement method	Infringement object		
	Server	Network	Client
Attack against vulnerability of information asset	– Creation of experience(MOB) by initializing server – Improper access via backdoor – IIS/ASP attack by utilizing vulnerability of uni code – Modification of database via game community web site – Attack against vulnerability of buffer overflow – Denial of service attack	– Obtaining information via port scanning – Wiretapping through packet sniffing – Camouflage via spoofing – Attack against vulnerability of communication protocol – Denial of service attack – Distributed denial of service attack	– Software forgery and alteration – Key log capture – Using map hack – Duplication of game item – Memory forgery and alteration
Utilizing malicious program by outsider	– Distribution of virus – Execution of worm program – Execution of bot program – Execution of unauthorized activity by Trojan – Denial of service attack	– Execution of worm program – Execution of phising program	– Using auto mouse – Using macro – Using speed hack
Social engineering	– Leakage of game information by insider – Theft of game information by outsider – Impersonation of administrator	– Physical disturbance of game operation by outsider	– Fraud – Leakage of ID & password

the survey were only 5 people. The gap between survey respondents was quite big at the initial stage, yet as the number of survey increased, difference between respondents has been stabilizing. Table 2 shows the final design of infringement method and game service infringement type according to infringement object.

There appear more specific infringement cases from game service client area, while similar patterns of infringement tend to be found from game service network or server area in comparison with patterns of infringement from general program. The specific infringement cases from game service client area are as below

- Software forgery and alteration: It alters game service execution program, therefore it allow game service to execute under the condition where not every execution environment is prepared [16].
- Key Log theft: Stores key log or event message as text file.
- Use of Map Hack: Visualizes the position of others or a piece of others' card which are set up as hidden.
- Duplication of game item: Duplicates the items used in the game service by utilizing a bug or vulnerability of service.
- Memory forgery & alteration: Unauthorized process approaches memory to manipulate process memory.
- Use of auto mouse input and macro: Automatically repeats inputs which are necessary for game service utilization
- Use of Speed hack: Modification to increase users' moving and attack speed in the game service by allowing users' devices to be faster irregularly.

– Fraud: The way to obtain game items by deceiving other users without proper transaction procedure.
– ID and password leakage: To obtain other users' access authority, seize ID or password with an immoral way.

Specific patterns of infringement which are detected in the area of game service are similar to general patterns of information infringement.

– Modification of database via game community web site: Outflow internal information stored by utilizing illegal questions to database.
– Inappropriate access through back door: System approach without log record and formal authentication procedure.

With classification standard of game service infringement as explained above, we analyzed the survey which was collected from developer or administrator of game service and users of game service for 10 days (10th Oct 2006 $\sim$ 20th Oct 2006). Amongst collected valid questionnaires (20 from developer or administrator of game service, 100 from users of game service), more than 68.9% of respondents have experienced game service infringement and survey results showed that the most frequent infringement cases are auto program (28.3%), software forgery and alteration (28.0%), and use of speed hack (20.0%) in the client area. The major infringement cases of game service for network area (20.9%) were packet sniffing and for server areas (9.3%) were back door and database attack. The reason infringement cases occur frequently in client area is due to characteristics of game service users' who utilize infringement tool to use game service easier, while infringements against general information system are to obtain certain information or disturb an operation of the system by attack the system.

Actual countermeasures to prevent those game service infringements have been established by no more than 50% of game service providers and 42% of those providers with countermeasures only have progressing an implementation of mere information security system. The reason why there exists a limitation of applying researches on information security for general application software to information security for game service, because information security for game service needs accumulated practical experience and understanding of characteristics of game service.

4. Case study on game service infringement

In this study, we have conducted an analysis of specific infringement cases (auto program, software forgery and modification, speed hack, and infringement of keyboard input information) that have been the most frequently occurring in client area of the game services, which are top ranked game services in Korea.

As cases of using auto program, we can see a repetition of the certain character/word/sentence, which is input to program in advance during game service and see a game character moving or hunting monsters in the game service automatically without users' command made by keyboard or mouse. Table 4 indicates functions of auto programs, which were collected from survey.

By utilizing software forgery and alteration program, ones can execute the certain game with randomly generated product key and terminate an operation of protection program which is designed to ban illegal use of game as shown in Fig. 1.

Once speed hack program is installed at game service, it makes it possible to shorten a travel time of 10 seconds, which is needed for game character to move to circled point vertically to less than 1 second. It also shorten a travel time of 20 seconds, which is needed for game character to move from right bottom

Table 3
Distribution of analyzed game service infringement cases

Criteria	Type of game service	Distribution	Ratio (%)
Operation environment	Mobile game service	27	90.0
	PC game service	3	10.0
Genre	Roll playing	19	63.3
	Casual	8	26.7
	Shooting	2	6.7
	Strategic simulation	1	3.3
Service	Trial service	10	33.3
Stage	Free service	12	40.0
	Charged service	8	26.7
	Total	30	100

Research Objects: Mobile Game Services provided in Korea (point of time, 1st Nov 2006).

Table 4
Auto programs for game services

Extracted mobile game service	Auto program	Method of activation
G*	G-Chat_v1	Input same character repeatedly and periodically
R****	sinroo_jansa	
S***	Macro G	Automated application of item by automatic input of certain key
K***	playKAL	
T**	G Macro	Automatically repeating certain mouse coordinates or keyboard input value
N*****, C**	ZleGEMc	Automatically conduct mouse click and input certain words periodically
D**	QMouse	
K***	AutoCloverSetup	Automated attack activity and application of item
H**	AutoClickProject	Automated cast and collection of fishing rod

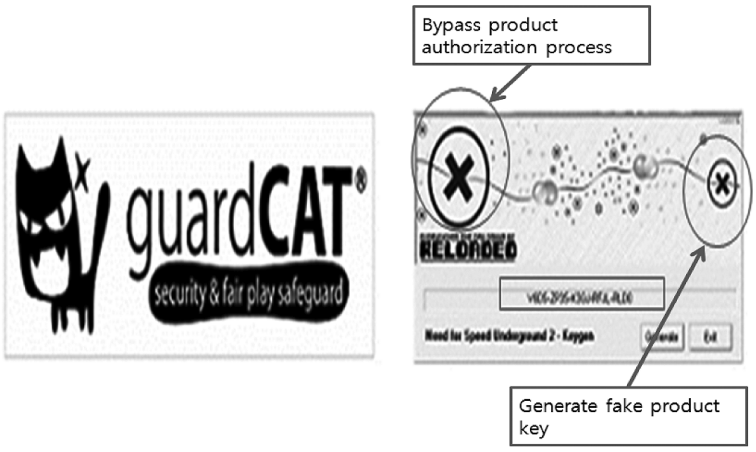


Fig. 1. Software forgery and alteration case.

of the screen to left bottom of the screen horizontally to less than 10 seconds as described in Fig. 2. Furthermore speed hack can increase an attack speed. Table 6 shows functions of speed hack programs which were extracted from game services.

Once key logger program is installed, which make it possible to see keyboard input information, someone's keyboard input information is extracted and displayed in word pad as Fig. 3 shows.

Table 5
Software forgery and alteration program for game service

Extracted mobile game service	Software forgery and alteration programs	Method of activation
N* * ***	Keyzen	Execution of game with illegal authorization by generating product key
Z**	zweipet	Execution of game without activating CD ROM
D*****	!FullScreen	Switch full screen mode to window mode
M***	gcdown	Blocking security software of M*** (Guard CAT)
S******	loader	Connect to Battle net without process of authorization
Y***	Ting∼Yock	Blocking jargon-free policy

Table 6
Speed hack programs for game service

Extracted mobile game service	Speed hack programs	Method of activation
D****	SPSF	Increasing/Decreasing game speed by multiple
D**, P****, K**, T**	Speed Gear	Increasing/Decreasing game speed of certain point of time by multiple
D** *, B****, C*****, K***, C***, K****	Speeder	

Fig. 2. Speed hack case.

5. Literature review

It is urgent to develop countermeasures for mobile game service infringement, because the most frequent infringement cases arise from client area amongst game service infringement areas as previous survey result. While countermeasures for infringement have been developed as below, the current technologies developed cannot cope with tamper resistance software [16]. Hence, we would like to develop the element technology, which completely blocks malicious users' illegal gains via software reverse analysis and alteration.

(1) API Hooking (Preventing Auto Program and Speed Hack)

Once hacking program is executed, call for API function, which causes a virtual input in the program, will take place [9]. Call for API function by hacking program calls import address table that is in the

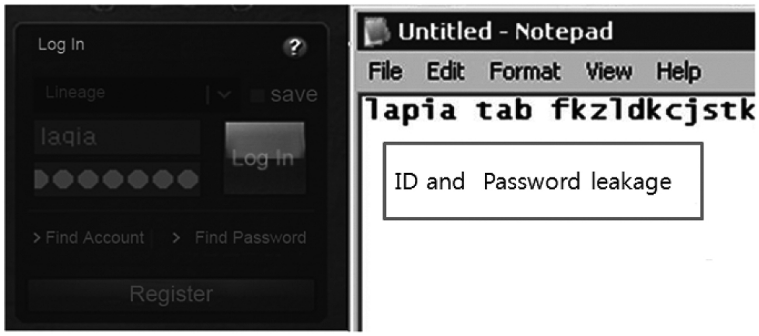


Fig. 3. Key log infringement case.

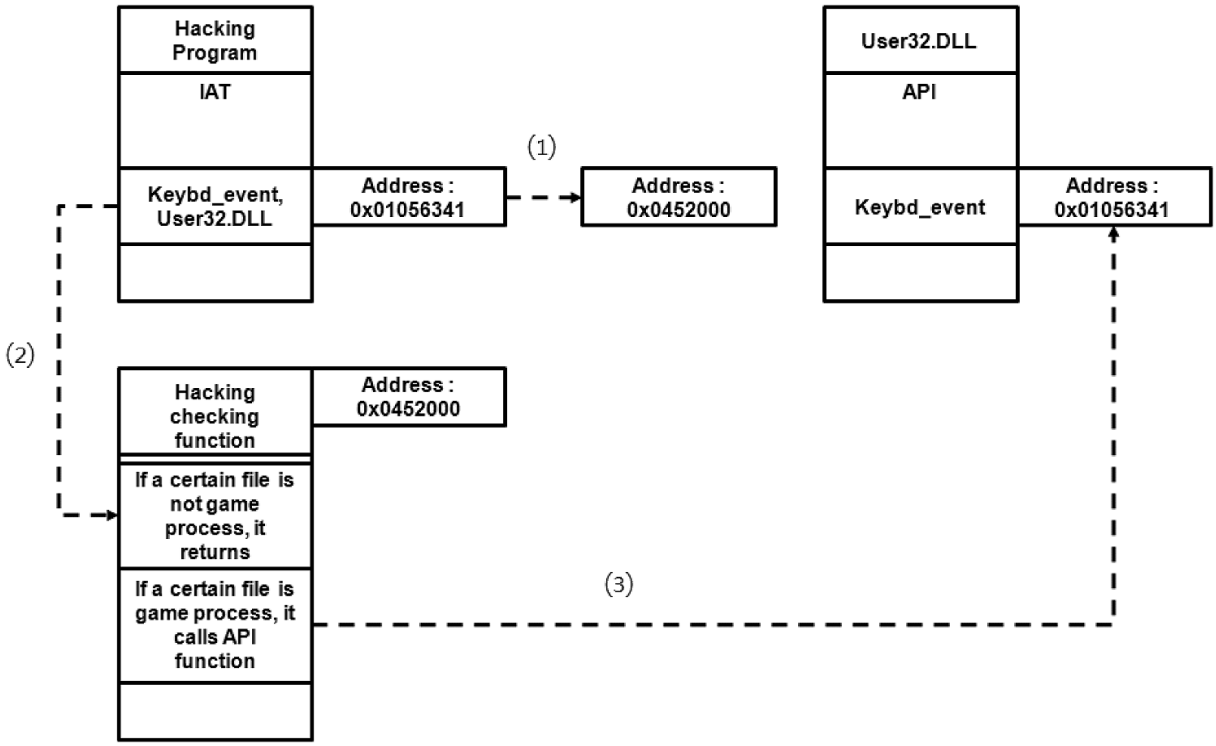


Fig. 4. Structure of API hooking.

program. At this stage hacking may be prevented by judging hacking with replacing that address with hacking checking function.

- 1st Step: There exists import address table which retains a list of API function address for activation of infringement program in the program. The infringement program brings the actual address from the certain function of 'USER32.DLL file (i.e. keybd event, and etc.), which is loaded to memory to execute game infringement [2]. When the program tries to call a certain function to use through function address retained in a infringement program, it changes a relevant address to address that hacking checking function has.
- 2nd Step: A countermeasure program for infringement checks every programs using illegally called

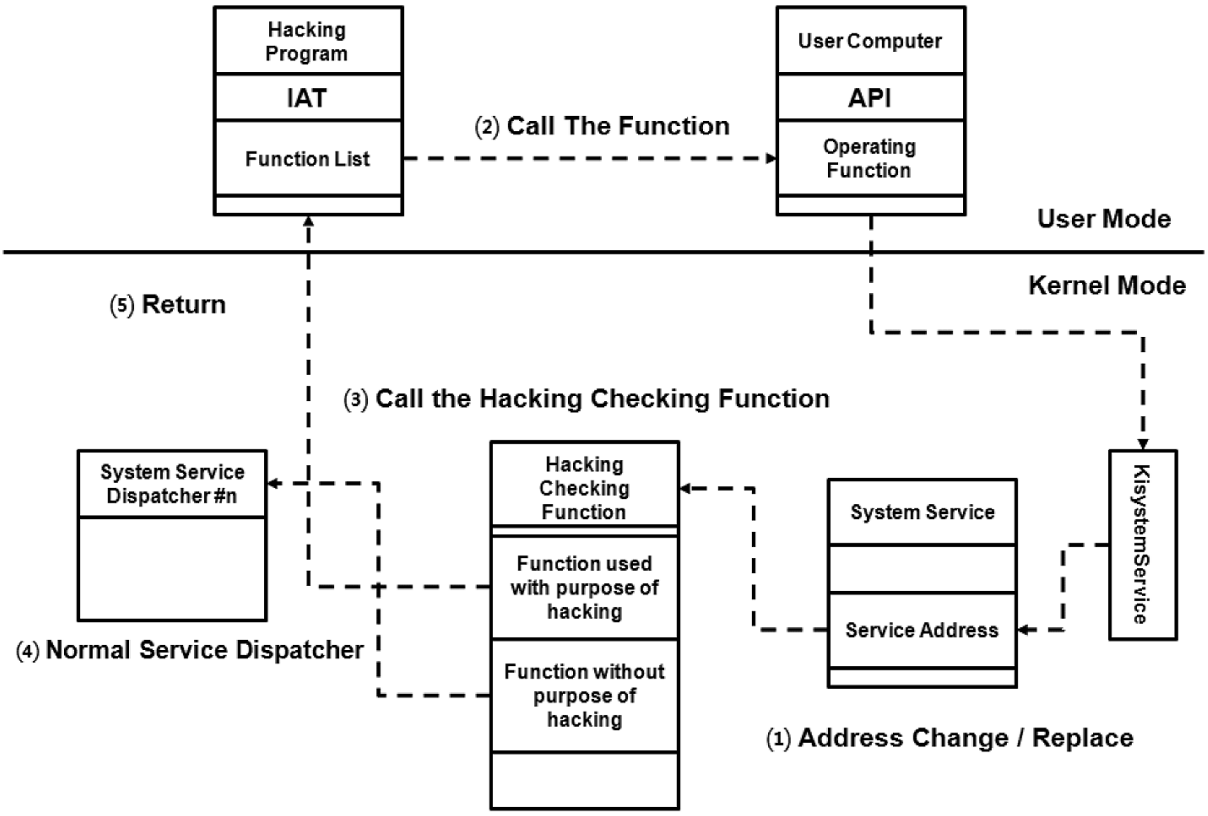


Fig. 5. Structure of SST hooking.

function (i.e. keybd_event, and etc.) for illegal access by applying hacking checking function to every program.

– 3rd Step: Hacking checking function judges whether program using a certain function is game process or not. In case that process is judged as unrelated, it would conduct a return instruction after classifying process as unrelated, which may harm the game service. If program using a certain function is game process, it will conduct an actual activity instructed after moving to an actual address.

(2) SST Hooking

If infringement program is executed, it calls relevant file by referring to API function retained in the import address table in the program to call API, which includes function (instruction) causing a virtual input in the program. An instruction of API function, which is being called, uses a virtual input service address that is in 'KiSystemService' of Kernel mode to carry out an actual activity [3].

– 1st Step: To activate an infringement activity, program calls API function. To use a certain function (i.e. keybd event), it seek the address of certain function of 'USER32.DLL file and call the address within file. Activity about a certain function is conducted by the called address.
– 2nd Step: The certain function of 'USER32.DLL and commands for virtual input exist within address. These commands approach the service, which is related to virtual input under kernel mode by generating software interrupt.

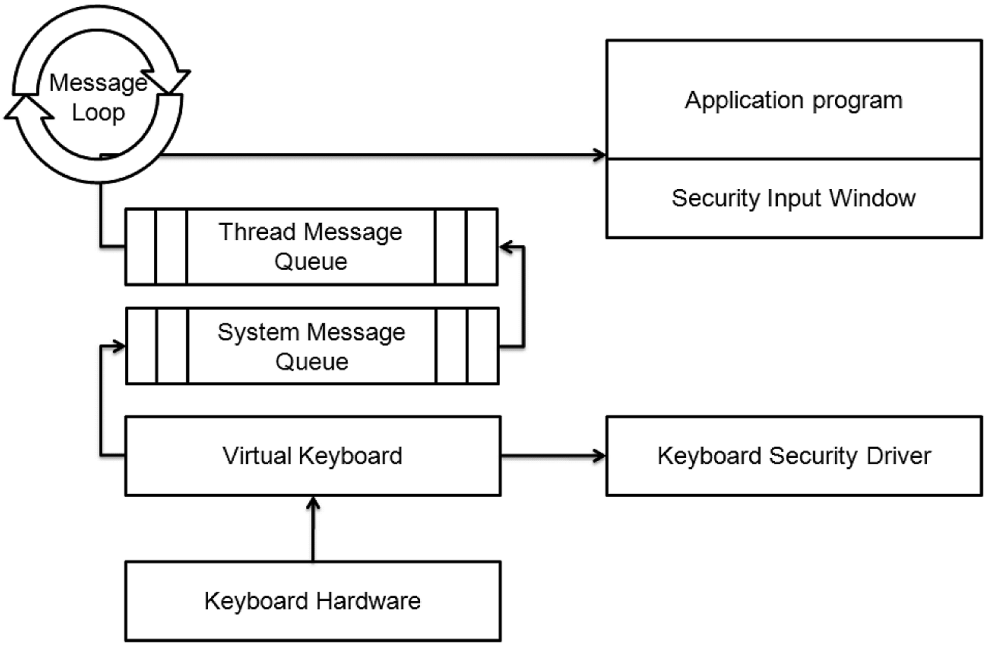


Fig. 6. Protection technology for key board security.

– 3rd Step: When commands of the certain functions approach service of kernel mode to execute activities, the used service is 'KiSystemService and this service approaches virtual input service address via System Service Table then execute an actual activity. At this stage command, which directs to move to hacking checking function is added to part of the virtual input address.
– 4th Step: Hacking checking function judges whether program using a certain function is game process or not. In case that process is judged as unrelated, it would conduct a return instruction after classifying process as unrelated, which may harm the game service. If program using a certain function is game process, it will conduct an actual activity instructed after moving to the part of virtual input service address.

(3) Key Board Security

The online-based key logger prevention technology grasps channels of keyboard input information leakage and vulnerability of them, and then integrates necessary technologies for prevention of those vulnerabilities. It prevents keyboard input information leakage by using keyboard security driver and security input window, which transmits keyboard input information to application program directly without general process [10].

– 1st Step: In case of occurrence of interruption caused by keyboard input, virtual keyboard driver loads the values of keyboard input information that is stored at the input/output port.
– 2nd Step: Keyboard security driver stops the flow of existing keyboard input information and directly sends keyboard input information retained by web page or application program to security input window.

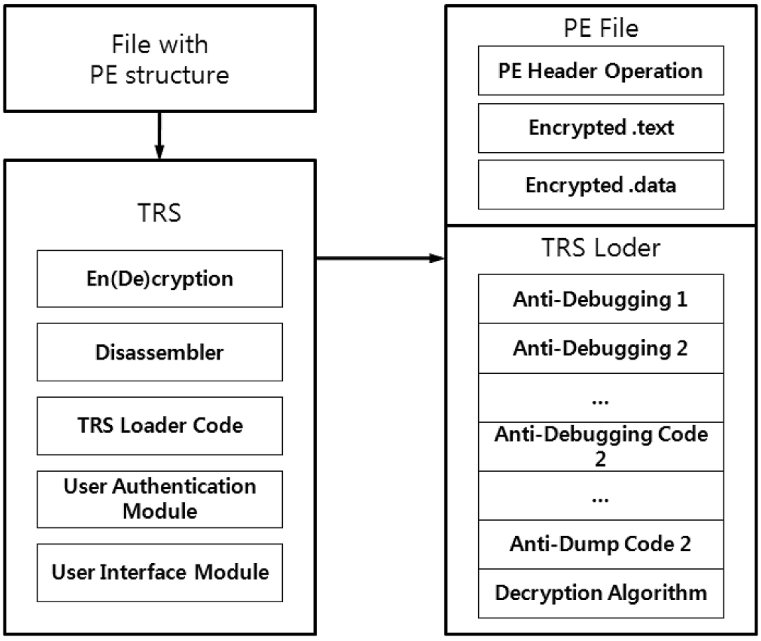


Fig. 7. TRS structure and operation flow.

6. Tamper resistance software for mobile game service

6.1. Proposed tamper resistance software architecture

Tamper Resistance Software is the technology to prevent a reverse analysis and alteration of software which is required for various application areas where software has to be protected from an illegal modification or deletion for e-commerce, industry related to digital contents, mobile game, etc. [11]. The mobile game service that Tamper Resistance Software for mobile game is applied precludes users from realizing an illegal benefit via a reverse analysis and alteration of mobile game service. Hence, it may protect service provider and user from damages.

Tamper Resistance Software for Mobile game is conducted by inserting specific functions into every file (.EXE, .DLL, .OCX, .SCR, etc.) which features PE (Portable Executable) structure. TRS consists of en(de)cryption part, disassemble for an analysis of machine language bite code, TRS loader code for action code of PE(software forgery & alteration prevention technology is applied), TRS user's authentication part, and user's interface part. Once general file which possess PE (Portable Executable) structure gets through the process for each function, PE (Portable Executable) file handles header of file and encrypts text and data section. Finally it generates a new file by adding TRS loader [17].

6.2. Proposed tamper resistance software components

6.2.1. Program executable file

There exist various kinds of file format supplied by Microsoft. Amongst those file formats, certain files related to execution (.EXE, .DLL, .OCX, .SCR) consist of standard format of PE (Portable Executable) structure [12]. The place where PE (Portable Executable) structure is applied is text section or data

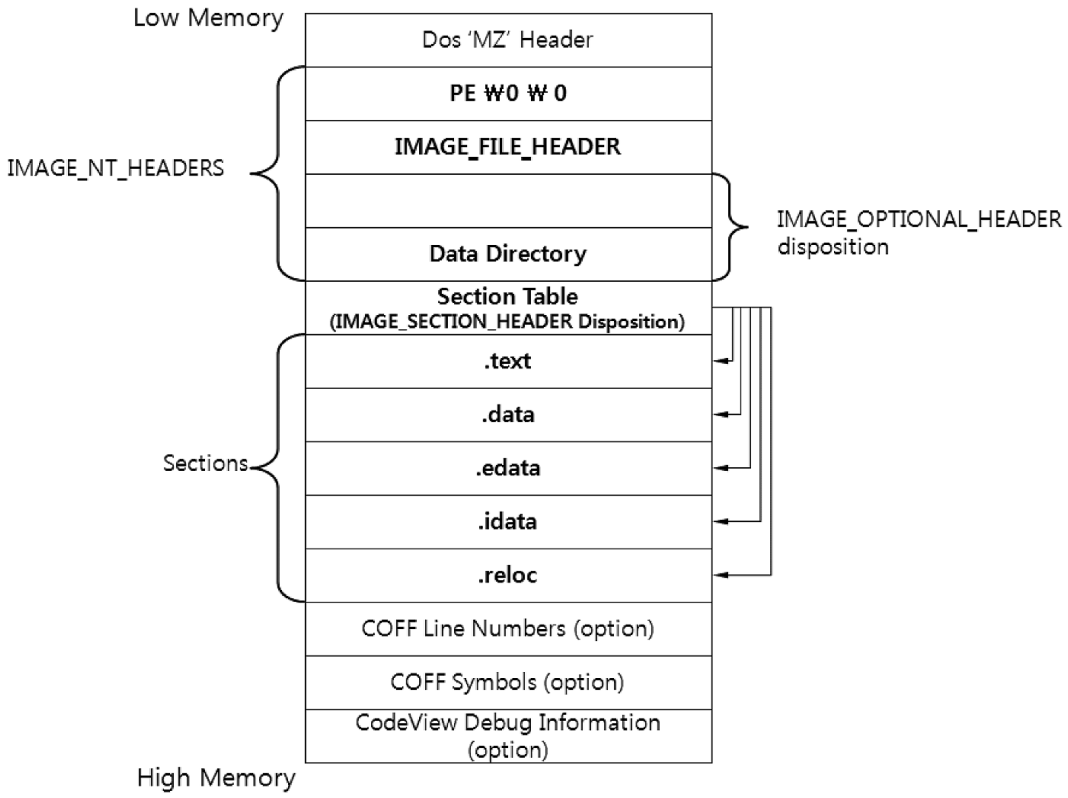


Fig. 8. PE structure related to execution.

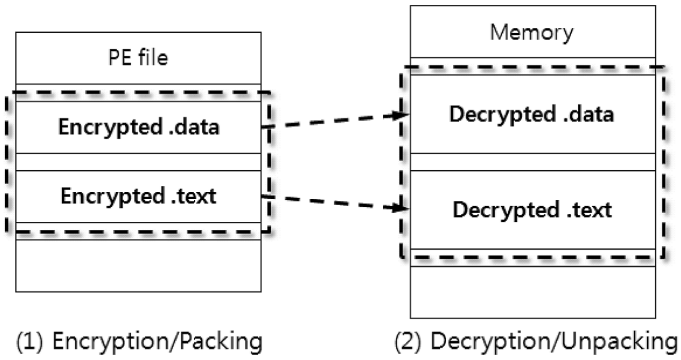


Fig. 9. PE structure mounted to memory.

section and software execution code is in text section and data which is necessary for execution is stored in data section. Encryption or packing of code and data area makes disassembly of software impossible.

When the file possessing PE (Portable Executable) structure is mounted to memory, it is encrypted or unpacked and the software mounted to memory is not protected [7].

6.2.2. Loader section trap

– Anti Debug, Anti Soft Ice: In case of discovery of debugger, it closes or ramifies into arbitrary

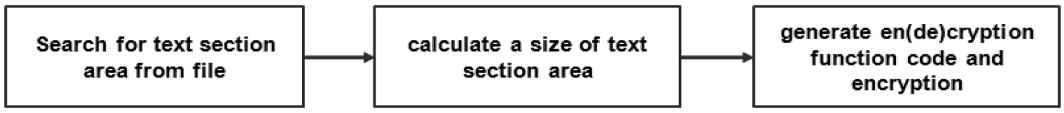


Fig. 10. Encryption process.

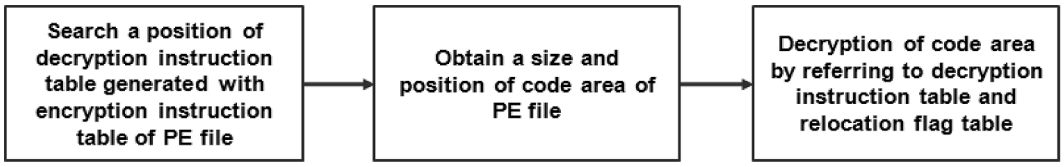


Fig. 11. Decryption process.

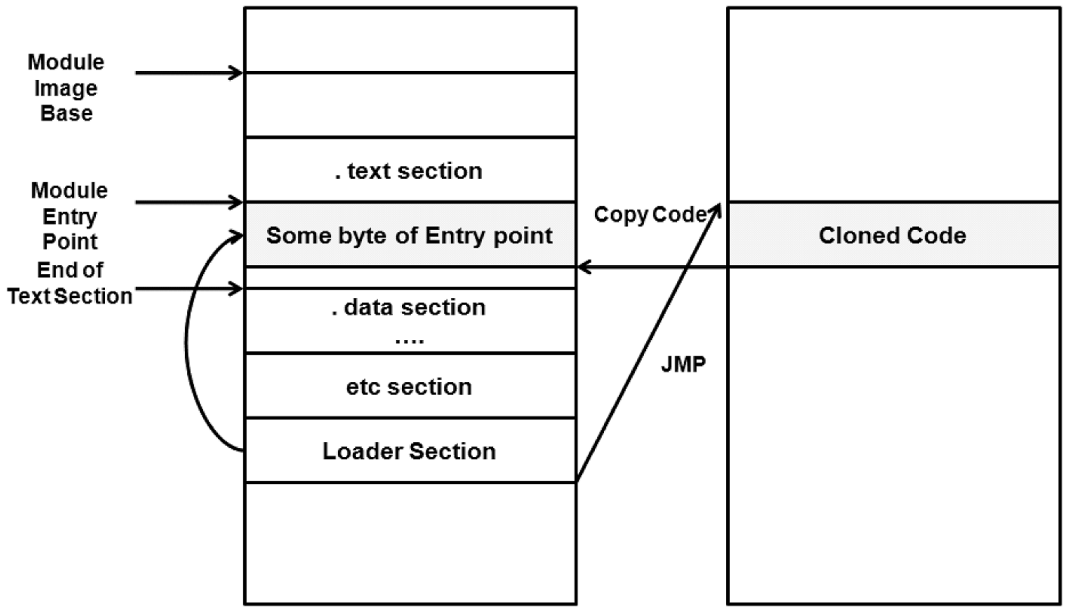


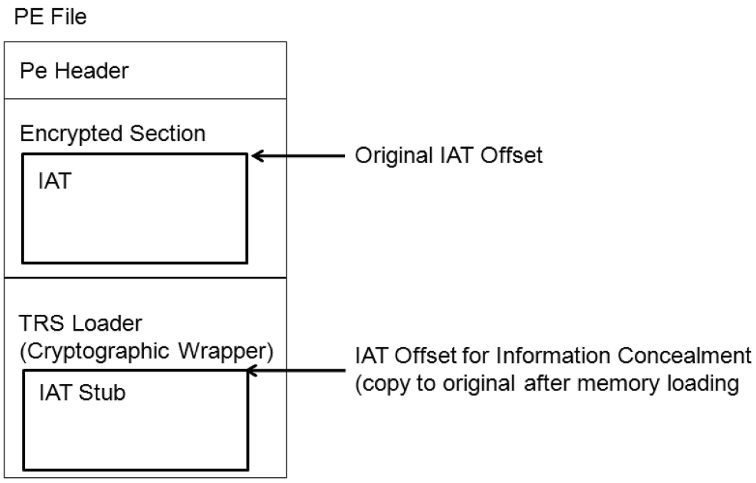
Fig. 12. Anti debugging process.

address through code which can identifies whether debugger work or not.

- Partial Checksum: Once debugger set up a break point, it mediates to execute a next instruction by utilizing change in memory code when a checksum of certain part of loader is correct.
- XOR Trap: The process of obtaining a regular option from encrypted option to get an option used in a loader.
- Transposed loader decryption code: It transposes a loader decryption code to prevent loader section from referring to loader decryption code table by using debugger or disassemble.

6.2.3. Loader section and text section en(de)cryption: Cryptographic wrapper

The loader en(de)cryption processes are same except for reference of relocation flag table when using section encryption process [4]. In addition, the decryption is operating same as encryption process via decryption instruction code.



※ IAT: Import Address Table

Fig. 13. Check sum of PE file right after decryption.

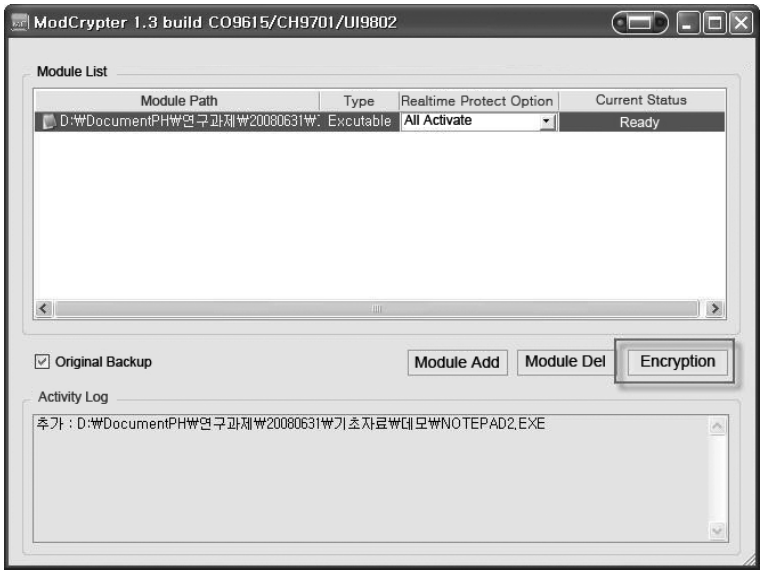


Fig. 14. Encryption of execution file.

6.2.4. Data section encryption: Cryptographic wrapper

In case of data section en(de)cryption, it is same as encryption process of .text section are, yet there exists difference [14]. When key materials do not match by utilizing hash value (Vender Key) of hardware information collected from user's mobile device, it does not operate properly.

6.2.5. Self integrity check

It connotes check sum value of ordinary file cryptographically and hides it, then one more inspection of check sum value is conducted for integrity when it is mounted to memory.

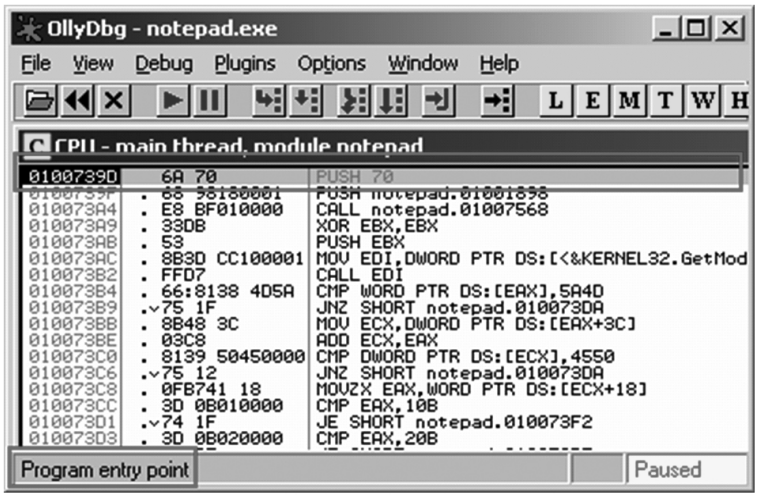


Fig. 15. Debugging result of the unencrypted EXE file.

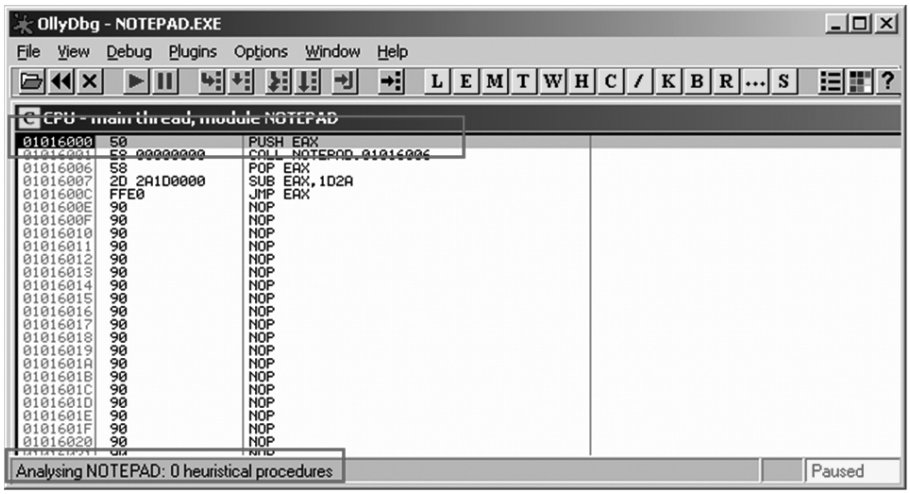


Fig. 16. Debugging prevention result of the encrypted EXE file.

6.3. Proposed tamper resistance software procedure

6.3.1. Anti debugging procedure

The basis of reverse analysis is debugging PE file (process) that is executing or executing PE file via debugger. The malicious user analyzes operation of process through debugging and inserts, deletes, and modifies code to induct unintended operation.

Hence to prevent these processes debugging, we can utilize technologies such as Structured Exception Handler, Interrupt Handler Operation, and Process Environment Block Modification to detect an operation of debugger [17]. The debugging prevention procedure by hiding entry point is explained as below.

– The loader decrypts .text Section and .data Section then grant a Import Address Table value.
– Import Address Table: It retains win32 system which is necessary for execution or information

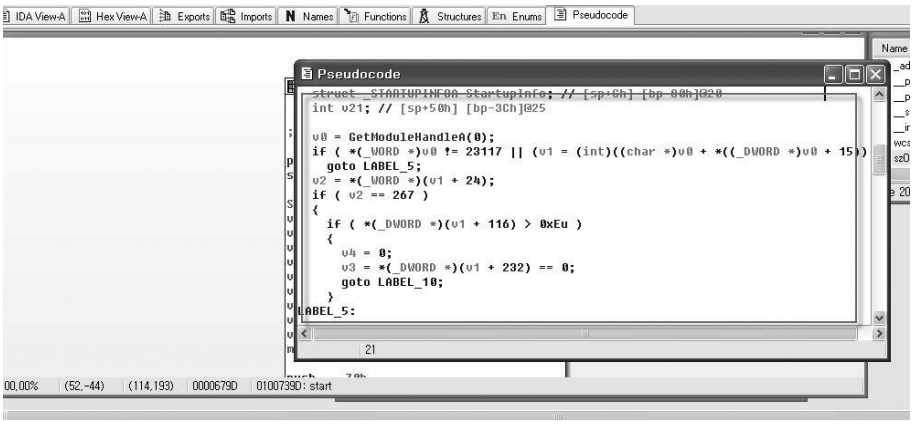


Fig. 17. Disassembling result of unencrypted file.

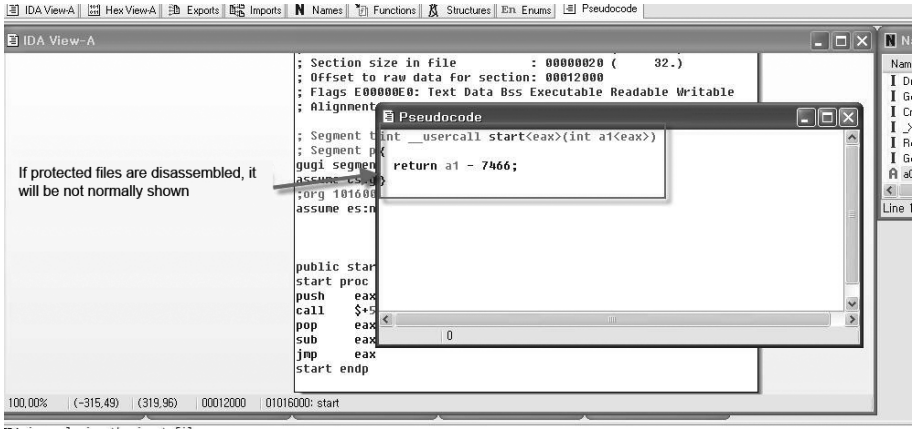


Fig. 18. Disassembling prevention result of encrypted file.

concerning user defined DLL. It also has direct information that a certain process calls which Win32 API.

- Before the loader jump to Original Entry Point, loader duplicates assembly code value of arbitrary length at the entry point of .text section to buffer.
- The size of code being copied is already computed as the length of well formed byte.
- Delete the byte as the amount of copied from text section.
- Loader moves to cloned code and executes the same amount of copied assembly code. Finally it moves to the place that right after deleted code.

6.3.2. Anti disassembly procedure

As previously explained, .text section and .data section of PE (Portable Executable) file cannot be analyzed by disassemble due to encryption by Cryptograph Wrapper.

6.3.3. Dependency hiding procedure

The malicious user can cause unintended operation in the program by setting up breakpoint of debugger or utilizing technology such as API hooking with gained information from Import Address table [15].

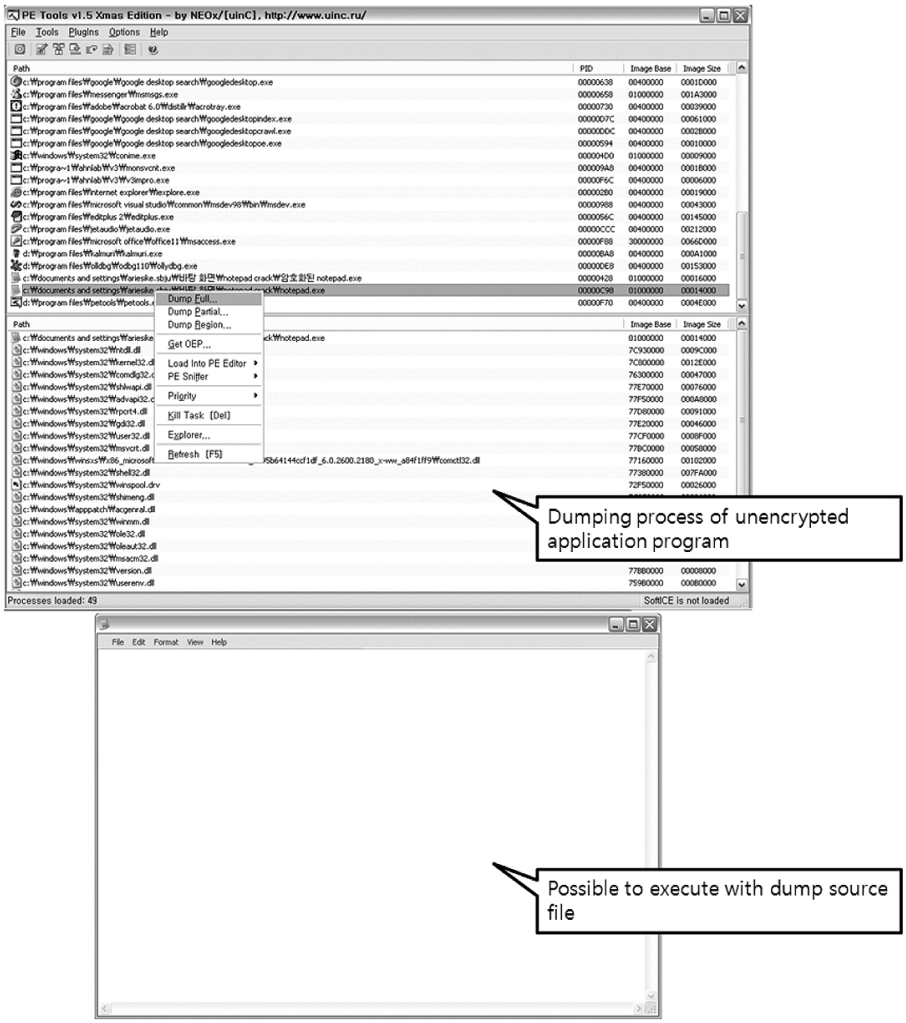


Fig. 19. Dumping result of unencrypted file.

Therefore in this study, we modified the information of PE (Portable Executable) file and hide the direct information via Import Address Table. Consequently modified information of PE (Portable Executable) file is restored when it is mounted to memory.

6.3.4. Anti memory dump procedure

Once PE (portable executable) file encrypted by TRS (Tamper Resistance Software) is mounted to memory, it is impossible to obtain an original binary after dumping process mounted to memory. Because encrypted PE (portable executable) file is decrypted right after mounting to memory.

7. Verification for proposed tamper resistance software

To examine the probability of application of mobile game service forgery & alteration prevention technology, we encrypted EXE file as below.

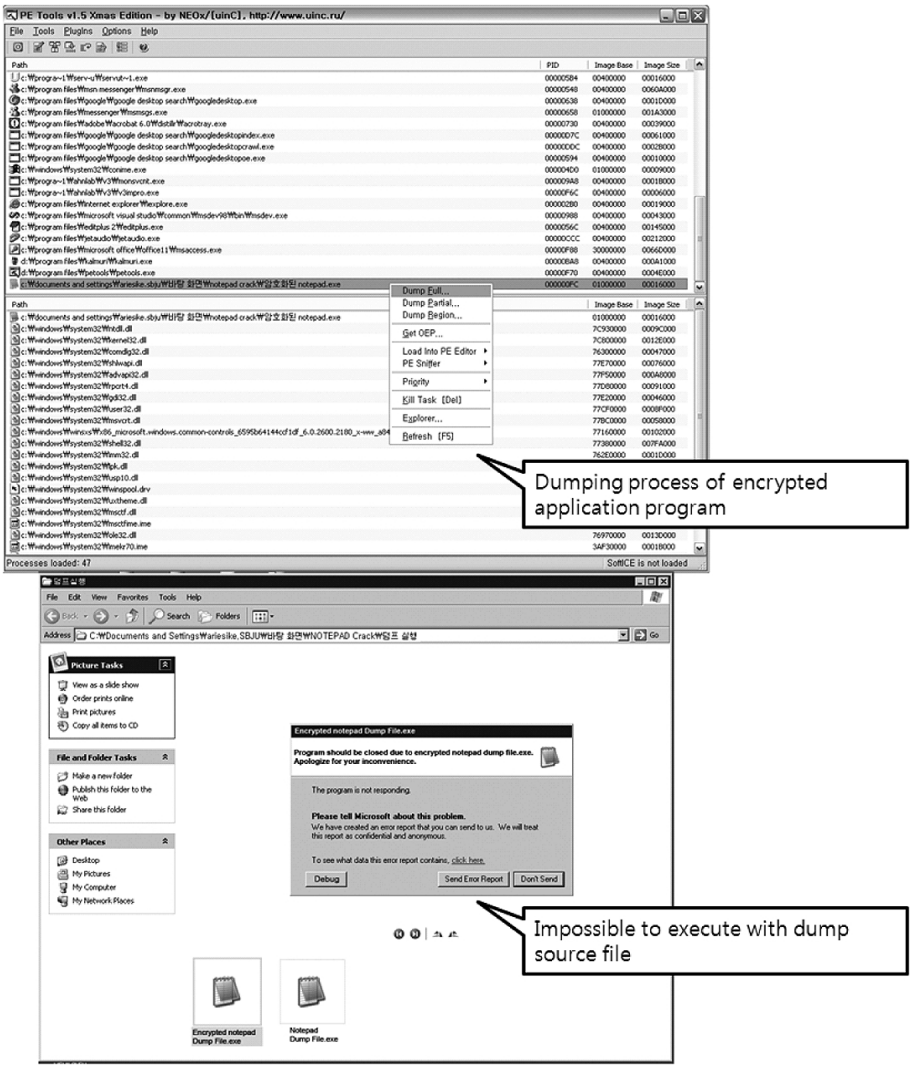


Fig. 20. Dumping result of encrypted file.

To conduct a confidentiality experiment concerning Anti Debugging, we verified that exe file encrypted by this study cannot be debugged with 'OllyDbg program(execution file debugging program)'. It is possible to debug by using unencrypted text program as below.

Yet for the encrypted execution files, we could prevent debugging as below.

To conduct a confidentiality experiment concerning Anti Disassembling, we verified that DLL file encrypted by this study cannot be disassembled with 'IDA Pro program (Disassembler of DLL file)'. Unencrypted msgsndr.dll file can be converted to assembly language by IDA Pro program as below.

In case of encrypted DLL file, it can be prevented from disassembling as below.

We have verified that the file that this study's technology is applied to is not executed by 'M Dump program', which dumps execution files. If one uses unencrypted application program, it will be executed as Fig. 19.

In case of encrypted .EXE file, it was possible to prevent dumping as below.

8. Conclusion

We have experienced lack of the differentiated information security technology research for preventing reverse functions when comparing with its social impact and a tendency of industry growing at high speed [1]. Therefore in this study, we designed technical countermeasures for mobile game service forgery & alteration which are the biggest security vulnerabilities deduced from the investigation of status of mobile game service infringement. Tamper Resistance Software for mobile game is a technology to prevent a reverse analysis and alteration of mobile game service and it is consist of Anti Debugging, Anti Disassembling, Anti Memory Dump, etc.

In this study, we have improved the preliminary study which needs independent compiler and hardware to prevent mobile game service forgery & alteration. We also have developed the technology that the execution file can prevent its reverse analysis and alteration by itself.

The specific method proposed by this study is that we have installed a loader twice which can separate data section and code section from execution file and en(de)crypt separated section. For anti debugging, we made mobile game service to be shut down when external debugger operation is detected. For anti disassembling, we changed the header of execution file that disassembling program is referring to for preventing operation of debugger. Consequently, for anti memory dump, we solve the problem by copying data which needs to be protected to temporary allocation memory. In this study, we have investigated the patterns of mobile game service infringement. The study results indicate that it is possible to preserve infringements against internal logic of mobile game service by actualizing mobile game planning and development with prudent consideration of adequate countermeasures and this means that it can reduce potential damage cost of country or enterprise, and even ripple effect. It may also guarantees the competitiveness of country or enterprise by protecting mobile game contents safely and it is expected that this could bring a revitalization of mobile game contents market. In further study, we are going to construct the integrated mobile game security environment for client area by progressing a study on keyboard input information security technology which obstruct a normal operation of mobile game service by outflow of user's ID or password

References

[1] A. Durresi and M. Denko, Advances in mobile communications and computing, *Mobile Information Systems* **5**(2) (2009), 101–103.
[2] A. Baker and J. Lozano, *The Windows 2000 Device Driver Book: A Guide for Programmers*, Prentice Hall, 2001.
[3] L. Bernard and R. Solms, A Formalized to the Effective Selection and Evaluation of Information Security Controls, *Computer & Security* **19**(2) (2000).
[4] E. Bott, C. Siechert and C. Stinson, Microsoft Windows XP Inside Out, *Microsoft Press*, 2001.
[5] D. Venugopal and G. Hu, Efficient signature based malware detection on mobile devices, *Mobile Information Systems* **4**(1) (2008), 33–49.
[6] E.N. Deker and J.M. Newcomer, *Developing Windows NT Device Drivers: A Programmer's Handbook*, Addison-Wesley, 1999.
[7] J. Eloff and M. Eloff, Information Security Management – A New Paradigm, Proceedings of SAICSIT, 2003.
[8] Inca Internet, *Method to cut off an Illegal Process Access and Manipulation for the Security of Online Game Client by Real Time*, Korean Patent 10-0483700, 2005.
[9] J. Daemen and V. Rijmen, The Design of RijndaeL: AES – *The Advanced Encryption Standard*, Springer-Verlag, 2002.
[10] B. Knittel, *Windows XP Under the Hood*, QUE, 2003.
[11] L. Barolli, H.-H. Hsu and Y. Shibata, Mobile systems and applications, *Mobile Information Systems* **4**(2) (2008), 77–79.
[12] S.D. Liming, *Windows NT Embedded Step-By-Step*, Annabooks, 2000.
[13] M. Safar, H. Sawwan, M. Taha and T. Al-Fadhli, Virtual social networks online and mobile systems, *Mobile Information Systems* **5**(3) (2009), 233–253.

[14] P. Norton and J.P. Mueller, *Complete Guide to Microsoft Windows XP*, SAMS, 2002.
[15] K. Otwell and B. Aldridge, The Role of Vulnerability in Risk Management, IEEE Proceedings of the 5th Annual Computer Security Applicant Conference, 1989.
[16] T. Peltier, *Information Security Risk Analysis*, Auerbach, 2001.
[17] N. Rajeev, Windows NT File System Internals: *A Developer's Guide*, O'Reilly & Associates, 1997.

Hang Bae Chang is a Professor of Business Administration at Daejin University, Korea. He received his Ph.D. in Information System Management from Graduate School of Information at Yonsei University, Korea. He has published many research papers in international journals and conferences. He has been served as chairs, program committee or organizing committee for many international conferences and workshops; Chair of IPC '07, MUE '07, UASS' 08, UNESST '08, TRUST'08 and so on. His works have been published in journals such as Journal of Super Computing, Computing and Informatics, Journal of Computational Information Systems, and Lecture Notes Computer Science. His research interests include issues related to Industrial Security Management and System.

Hyuk Jun Kwon is a student of Graduate School of Information at Yonsei University, Korea. He has received his master degree in Business Administration from Graduate School of Yonesei University, Korea. He has published many research papers in international journals and conferences. He has been serve as session chairs and program committee for few international conferences and workshop; UNESST' 08, CSA' 08, ISA' 09, and so on. His works have been published in Journals such as Journal of Computational Information Systems, and Computing and Informatics. His areas of concern are Information Security Management, and Virtual Reality.

Jong Gu Kang is a student of Business Administration at Dajin University, Korea. He has published a few research papers in international journal and conference. His areas of concern are Industrial Security Management for SMBs, and Behavioral Aspects of Industrial Technology Theft.