Online Software Copyright Protection Using Trust Platform Module and Database Watermarking

Abdullah A. Al-Kushari[1] Ammar Zahary [2] Mohammed M. Alkhawlani [3]

ABSTRACT

As the digital information can be copied and transmitted easily, new and novel methods for the investment safeguarding are highly important. Many software companies developed several ways to prevent their software from piracy but hackers are still working. Other companies have their own techniques, but they keep them for confidential. This paper utilizes the digital watermarking and the Trust Platform Module (TPM) to enhance software piracy protection and decrease the efforts costs to protect software copyrights. The paper conducts a new approach called Trust Platform Module and Database Watermarking (TPMDWM) which is developed to enhance the software copyright protection. The proposed approach combines the new technologies of Trust Platform Module with database watermarking, hash function, and advanced encryption standard to enhance software copyright protection and decrease the cost and loose of software revenues. As shown by results, TPMDWM can effectively prevent illegal copying, getting the software at installation or updating license. The results also show that TPMDWM approach does not affect the software code and control. In addition, TPMDWM decreases dongle cost and it can also be used for client identification.

KEYWORDS: Trust Platform Module, Database Watermarking, Hash Function, Public Endorsement key, Advanced Encryption Standard.

[1] - The Arab Academy for Financial and Banking Sciences Sana'a Branch, Yemen Email: abdullah.alkushari@itexsolutions.com
2 - Faculty of Engineering & IT Taiz University Taiz, Yemen Email: aalzahary@gmail.com
3 - Faculty of Engineering University of Science and Technology Sana'a, Yemen Email: m.alshadadi@ust.edu.ye

1. Introduction

During the last two decades, the computer software industry has grown rapidly. Today, the Internet has become a major distribution channel for digital information. This presents both opportunities and challenges to software vendors. As the digital information can be copied and transmitted at great easiness, methods for safeguarding the investment are highly important. Illegal software copying and sharing causes high revenue loss for software development companies. For every legitimate copy of software that is sold, it is estimated that three or four illicit copies are made. This trend has driven software vendors to implement copy protection mechanisms to protect their applications and revenue and ensure that their applications are used legally. Many methods are available now for protecting the computer software from misuse. The protection methods cover a variety of techniques ranging from legal protection by copyright and patent to technical methods. The technical methods include both hardware- based and software-based approaches. They have very different features. Hardware-based approaches mainly provide preventive measure while software-based approaches provide both preventive and detective measures [1,2].

This paper studies the problem of software piracy that may cause losing a lot of money for companies that producing software and permit hackers to use software illegally. The paper presents a brief literature review for the current techniques for piracy protection. Then, it designs a model that helps the companies to enhance their piracy protection techniques. The model utilizes several new technologies such as Trust Platform Module (TPM), Database Watermarking (DWM), Hash Function, and Advanced Encryption Standard (AES) to enhance software copyright protection and to decrease the cost and lose of software revenues. Our proposed approach is called Trust Platform Module and Database Watermarking Model

(TPMDWM). It constructs the machine identification using TPM that is used to identify the customer machine by using Public Endorsement Key (PEK) that can decrease the cost of using dongle as customer identification. The database watermarking is used to hide the information of customer and his machine identification by using AES, which encrypt the data in the database. The hash function is used to create a signature for the customer and to protect the data in database from attackers. TPMDWM has been implemented and evaluated and it gives good results for the ratio of software execution time, the size cost of database and it enhances security and robustness to resist attack's modification.

The next section of this paper outlines the literature related to the paper subject. The third section presents our developed approach and methodology and its main features and benefits. The implementation steps of TPMDWM are explained in the fourth section. Intentional attacks that are possible on a database are illustrated on the fifth section. TPMDWM testing and result study are carried out in the sixth section. Finally, the paper is concluded in the last section.

2. Related Work

Many literatures that are related to the subject of software piracy protection, software watermarking, database watermarking and the ability to use TPM for software piracy protection have been found out. This section presents the related work to this paper.

The authors in [1], analyse some existing methods of piracy prevention and suggest a complete solution for secure electronic anti-piracy software installation and distribution. Their methodology try to deter the software pirates while maintaining the software's ease of use. The authors in [2] present a TPM based framework to improve the security of spread

spectrum watermarking scheme. C. Collberg et al. [3] identify three attacks types on the intellectual property contained in software and three corresponding technical defences. They briefly survey the available technologies for each type of defence. In [4], the authors explain a trusted computing platform to improve the computer security and build a trusted computing environment for both PC system and distributed computing system. They analyze the key and credential mechanisms, which are the two main aspects in the cryptology application of trusted computing. They also give an example application to illustrate that the TPM enabled protection can improve the security of distributed computer system.

In [5], the author introduces software watermarking and obfuscation to protect software from unauthorized access, modification, and tampering. Firstly, he formalizes two important concepts in software watermarking: extraction and recognition and he uses a concrete software watermarking algorithm to illustrate several issues in these two concepts. In addition, a technique called the homomorphism functions through residue numbers to obfuscate variables and data structures in software programs is developed. X. Zhang et al. [6] proposes a hash function based dynamic software watermarking algorithm. The proposed algorithm constructs appropriate hash function with embedded watermark piece. Given the satisfied parameters, the hash function calculates out the corresponding watermark. Paper[7] presents and explores a real scenario for data encryption utilizing trusted computing technology and its core is TPM. The processes from activation of TPM to the encryption of data are investigated and the advantages and limitations of the proposed protection have been addressed.

3. TPMDWM Approach

TPMDWM uses several new technologies such as Trust Platform Module (TPM), Database Watermarking (DWM), Hash Function, and Advanced Encryption Standard (AES) to enhance software copyright protection and decrease the cost and lose of software revenues. Our approach consists of four steps that could be mentioned as follows:

First step: collecting the fingerprint information of a new customer by using batch file online or offline on the main machine that contains the software product. This step collects the information that constructs the customer machine signature to differentiate him/her from other customers.

Second step: sending and registering the fingerprint information of the customer (machine id, customer name, customer_id, email, address, country, purchase order, purchase order date, activate date, serial_no key, secret key and status) in the provider database. Some information comes from interactive form when batch file is executed.

Third step: constructing the one-record customer signature table using database watermarking with hash function and encryption using AES.

Forth step: embedding the customer signature table into the software database and then constructing a library validation for software copyrights with obfuscation. After that, the software will be sent to the customer to install. Also the signature is used when updating the software license.

There are many special features of TPMDWM that could be summarized in the following points.

- TPMDWM takes features of dongle idea by considering the machine as a dongle.
- TPMDWM uses new techniques such as TPM and DWM to enhance software piracy protection.

- TPMDWM traces distribution of software products by provider database.
- TPMDWM can create a fingerprint or a signature of customer before installation process, which can protect software product before distribution.
- There are many benefits of TPMDWM for software providers. These benefits are significant in different aspects such as the ease of use, construction, and maintenance, the wide range of software copyrights protection, and decreasing cost and time.

4. TPMDWM Implementation

In this section, the implementation steps of TPMDWM are explained. These steps are illustrated by Figure 1.

4.1 Collecting Fingerprint Information

The fingerprint information for a new customer is collected by using batch file concept. In this step, batch file can be executed online from the company web site by customer account id and password or offline by sending the batch file to the customer after signing the contract. This batch file collects the main fingerprint information of the customer. This step will be repeated if there is any change in the customer fingerprint information. The collected information includes machine identification, customer name, email, address and country. The machine identification is the main part of this step to get the PEK. PEK in TPM is used as the machine identification in this step.

4.2 Sending and Registering in the Database

In this step, the customer fingerprint information (machine id, customer name, customer_id, email, address, country, purchase order, purchase order

date, activate date, serial_no key, secret key, nodes, status, and customer signature) is sent and registered in the provider database. This information helps the provider to trace and check the customer signature.

The information contains “machine id” field which is based on TPM identification. The “customer_id” field is a unique identifier for the customer. “activate date” is the first activation date on the Internet. “Serial no. Key” is a key that is used" as a first level protection during the installation process. Serial no. key can be public or private. The “Secret key” field is used in AES algorithm to hide the information in the signature table and to process transitions between the customer and software provider. “Customer signature” filed is created in the third step as customer watermark and it is used for comparison when the customer needs to activate software via internet or make sure if he has copyrights for product.

4.3 Constructing and Encrypting the Customer Signature Table

This step describes watermark embedding and algorithm encoding in TPMDWM approach:

Watermark Embedding

Construct the customer signature table that includes TPM as a machine identification using PEK, customer_id, and serial no key. Customer signature is a table with one record that use database watermarking with hash function and AES encryption.

Encoding Algorithm

Construct customer signature field according to the following steps:

1. Concatenate machine identification PEK from customer information and last encrypted information in customer signature table PEK, customer_id, and serial no key.
2. Run hash function for last concatenation and insert the result in customer signature field in both tables; customer signature table and customer

information table. The hash function of attributes is used to get a tamper proof receipt for any update in customer signature table. Secure Hash Algorithm SHA-1 is used to digist the attributes with customer machine id to have a unique signature per customer. We can consider the customer signature field in both tables (customers table and customer signature table) as a watermark which validates the right customer that has software copyrights.

4.4 Embedding the Customer Signature Table into Database

In this step, the customer signature table is embedded into the database as a user fingerprint information. The validation library for software copyrights with obfuscation is constructed. The library validation contains many algorithms as follows: a) watermark detection algorithm which is the main algorithm for checking and constructing customer signature, b) machine_id detection algorithm which detects PEK of machine and other customer information, and c) hash function algorithm which is used for construct customer signature.

4.5 Watermark Detection Algorithm

The watermark detection algorithm is working as follows:

1. Recalculate the customer signature for customer signature record (based on all attribute except customer signature) using the same calculation steps done during embedding stage. At this time, concatenation uses the machine identification PEK that is read from TPM in customer machine.
2. compare the calculated customer signature in the step 1 with the customer signature attribute in the customer signature table reside in the customer side or with the customer signature reside in the customers table at the provider side via internet. If they are the same

signature then, certify the database that it has been received without any tampering, otherwise reject, stop installation, and lock database

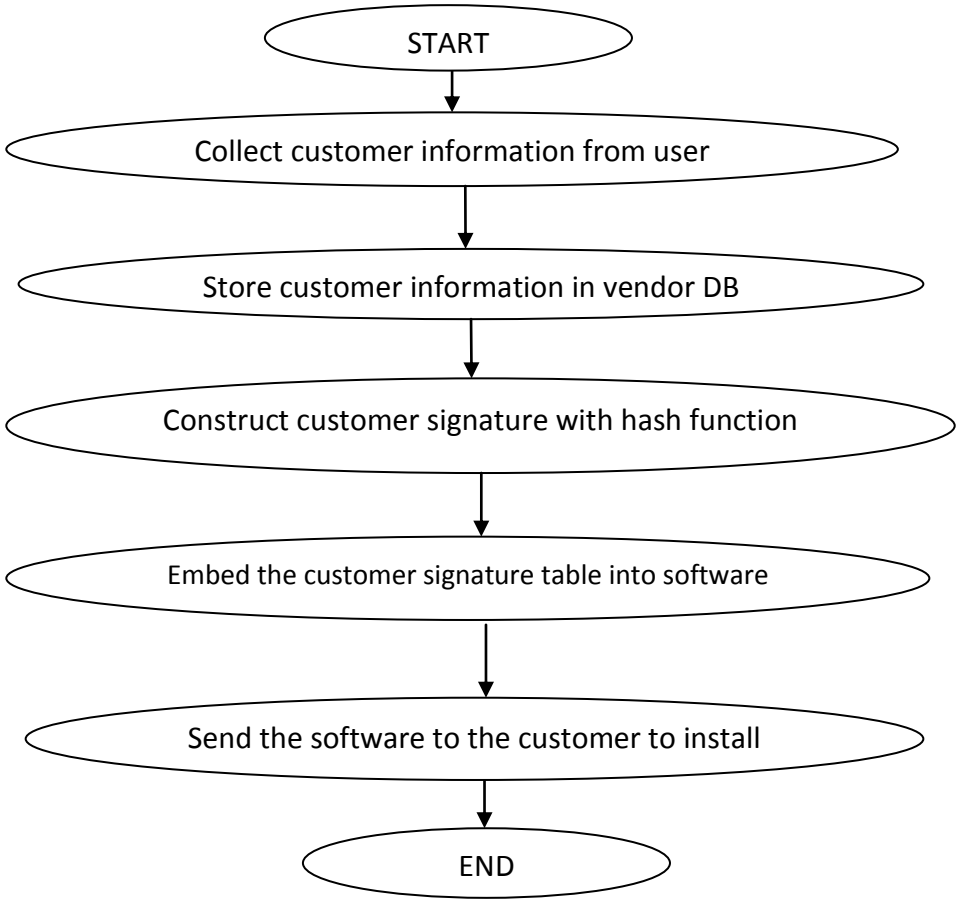


Figure (1): Activity diagram of TPMDWM implementation steps

5. TPMDWM Attacks Analysis

Intentional attacks that are possible on a database could be one of the following: a) inserting a new tuple into the customer signature table, b) altering attributes values, c) deleting some tuples from the customer signature table, and d) reverse engineer attacks.

5.1Insertion Attack

If the attacker inserts an extra tuple into the existing customer signature table, the validation library will issue an exception and locks the database and installation because this attack is happened. For success insertion, the attacker has to have a secret key which is owned by the provider only. So, the attacker couldn't have the same customer signature which is located in the customers information table so the application and database will be locked.

5.2 Alter Attack

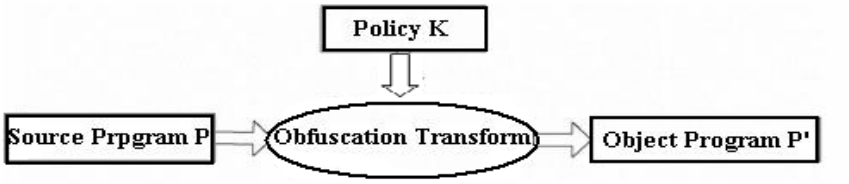
If the attacker wants to alter a column value, he/she should have the secret key. If any alteration attempt happened, the comparison between the calculated customer signature and the customer signature already registered in the customers information table will fail. This is because the calculated customer signature column is based on the hash function that constructs a unique signature and machine identification PEK that is read form current TPM.

5.3 Deletion Attack

If the customer record is deleted from the customer signature table, the validation library will lock the database and send some tracing information to the provider. If the owner customer has problem in his server, he should coordinate with the provider to change his signature depending on new TPM.

5.4 Reverse Engineer Attack

These attackers try to do reverse engineering for your software to change the validation information of the software owner. Hence, the validation library may use the obfuscation to prevent and decrease the risk of this attack. Figure 2 illustrates the obfuscation for a software code.



Figure(2): The model of code obfuscation

6. TPMDWM Testing and Results Study

The aim of this section is to execute TPMDWM testing after finishing the implementation stage in the previous section. A brief discussion about the performance matrix of TPMDWM has been applied in terms of the execution time, the size cost of database, and the robustness to resist attack's modification and security. To test TPMDWM, we should have hardware that supports TPM technology and operating system and package that can be used to read PEK from TPM chip. The code has been emulated to create the PEK that satisfy its attributes which are "Be Unique" and "2048 bits". We have developed the application that fulfills our model and it has been tested in a network that contains 10 clients. The testing proves that the machine signature construction can contribute to prevent software copyrights, decrease numbers of attackers and increase attacker's effort to obsession on software. Two steps have been done to achieve the testing of TPMDWM: machine signature construction and machine signature checking.

6.1Machine Signature Construction

In this step, the machine signature is constructed. This step started by creating an emulation for PEK, however this step will not be needed if the hardware and software supports TPM. After that, PEK is read with the

customer information by registration or execution the batch file. These information is stored in the customer's information table at the database of the software vendor. By using the encryption methodology AES, we construct the customer signature table that contains the encrypted customer information and this table is added to the software application database. Finally, the customer signature field is constructed by using the hash function and it is added to the database of the software vendor (customers information table) database. It is worthy to mention that all these steps are done at the software vendor site only. Figure 3 explains these steps.

6.2Machine Signature Checking

In this step, the machine signature is checked at the customer machine. This step starts by creating PEK such as the one created in the machine signature construction step. After creating PEK, TPMDWM reads it and encrypts customer information (in the application database) to construct a new customer signature by using hash function. Then, TPMDWM compares between the new customer signature and the customer signature field which already stored in the application database or in the license batch file. The last step depends on the vendor strategy. The purpose of this step is to validate the correct owner of software and to protect others from getting the software for free or hacking it by accept or reject features and freezing the usage of the software. It is worthy to mention that, these steps are usually done at customer site only at installation or during updating license. Figure 4 explains these steps.

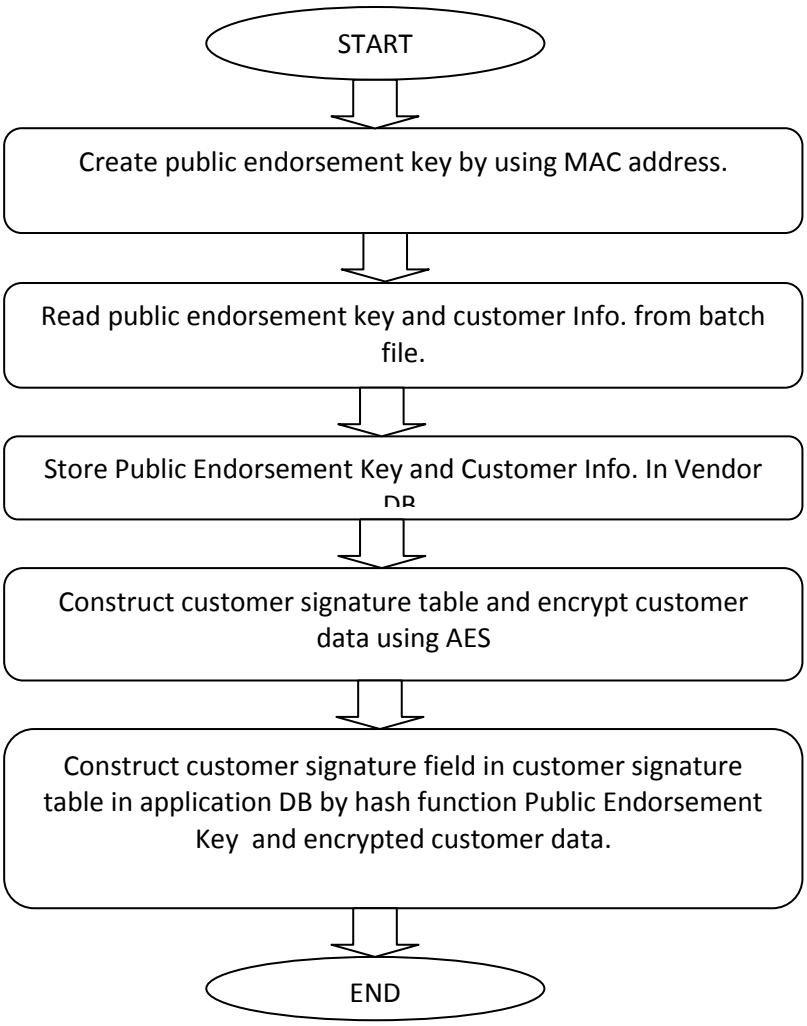


Figure (3): Machine Signature Construction

6.3 Performance Matrix of Our Approach

Our approach has been evaluated against four performance metrics. These metrics are the period of execution time, the size cost of database, the robustness to resist attack's modification and the security. The next paragraphs explain all these metrics and their effect on our approach.

The execution time

The ratio of the execution time of the watermarked program to that of the original program are calculated. In TPMDWM, the performance is not

affected because in one side the signature construction is done in the software vendor which doesn't affect in the performance of application. In the other side, the step of signature checking is performed during the installation or updating the license so it is also does not affect the total performance.

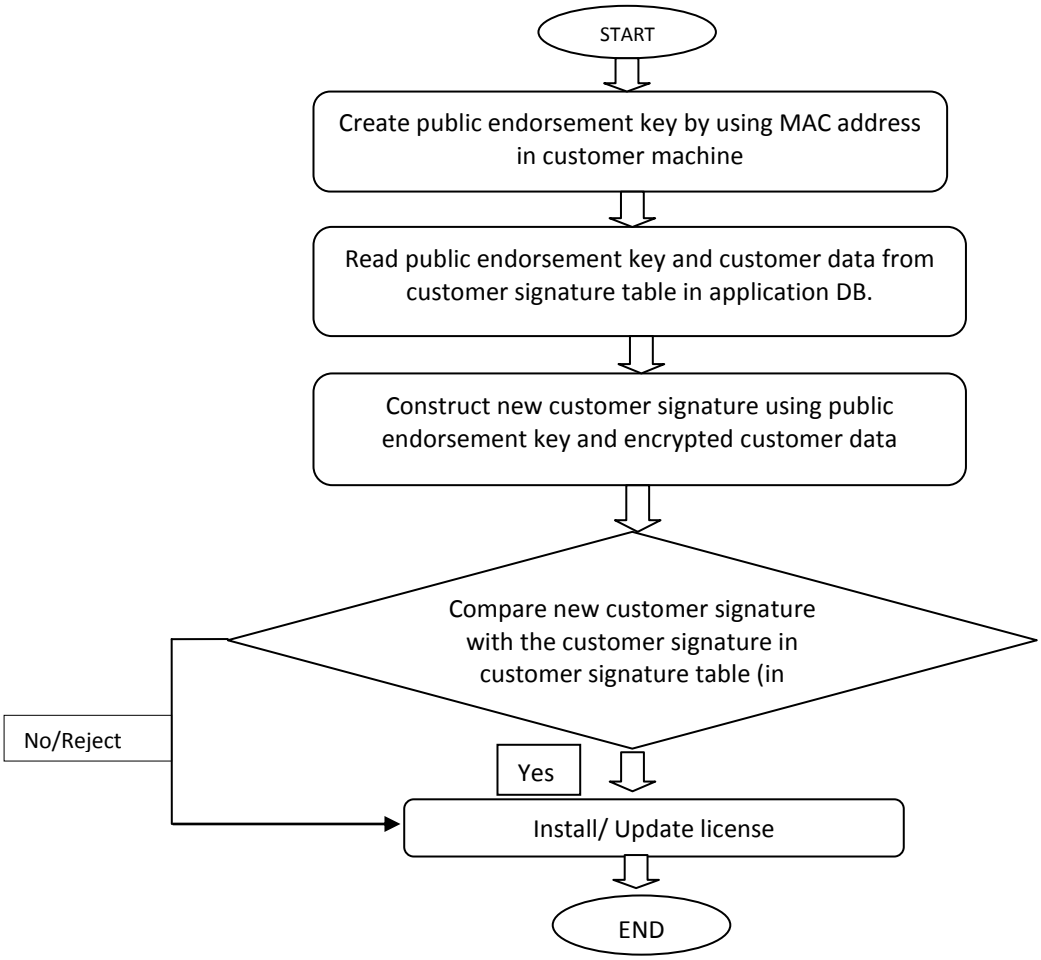


Figure (4): Machine Signature Checking

The testing has been done in five computers with different specifications to compute the average Extra Time (ET) that added to the execution time of software. Table 1 explains the extra time that is added to the software execution time as follows:

T1: the execution time of software with TPMDWM.

T2: the execution time of software without TPMDWM.

ET: the extra time that is the difference between T1 and T2.

AVG: the average of extra time which is added to our software.

Table (1): execution time with/without our approach

PCS	T1	T2	ET	AVG
PC1	2	1	1	1
PC2	3	1.5	1.5	1.5
PC3	2.5	2	0.5	0.5
PC4	2.5	2	0.5	0.5
PC5	2.5	2	0.5	0.5
TOTAL	12.5	8.5	4	0.8

The size cost of database

The size cost of database is expressed as the amount of information that could be stored in a watermark or database [8]. In TPMDWM, a small extra size for machine signature field (software watermark) in vendor database is required. In the application database, only one record is added in the table for customer signature so it doesn't affect the size of database. The size of database is computed as shown in table 2:

S1: is the size of software and database files with TPMDWM.

S2: is the size of software and database files without TPMDWM.

ES: is the extra size of software and database files that is the difference between S1 and S2.

Table(2): Size Cost with /without TPMDWM

PC	S1	S2	ES
Size (KB)	65.8kb	53kb	12.8kb

Robustness to resist modification attack

This paragraph studies the ability of TPMDWM to resist attacker modification in database tables where attackers can modify the content of customer signature table. Any modification can causes that the application doesn't work because any change in any input character causes a different output so the customer signature doesn't match the same signature in the database because of the hash function. There are two cases where the application or software can be run by another customer as follows: Case 1: if the customer has the same PEK to construct the same customer signature which is impossible. Case 2: if the attackers have experience in reverse engineering to stop and change the code of software used for checking customer signature. This case could by overlapped using code obfuscation. If the customer isn't the right owner, any online license updating can give us information about the true customer signature checking. Results show the effect of attacker modifications specially who has experience in software reverse engineering. As shown by results, the obfuscation of code can help software provider to resist these types of attackers.

Security

A watermark should be secured so, it will be hard for unauthorized party to detect or remove the watermark or knowing the algorithms used for embedding and extracting the hidden information [9]. TPMDWM depends on the security of TPM, AES and Hashing Function. All of these techniques cooperate to increase our TPMDWM security.

7. Conclusion and Future Work

In this paper, we have conducted a new software piracy protection approach called Trust Platform Module and Database Watermarking Model (TPMDWM). The approach utilizes the benefits of TPM such as computer

identification by non-migratable keys. It also utilizes the benefits of the public endorsement key as a unique key that has been used for software copyrights protection while decreasing the cost of using any other hardware such as dongle. In addition, database watermarking, advance encryption standards and hash function have been used to hide and encrypt data in the database without noticeable overhead in the database transaction and database size. TPMDWM approach also shows that it does not have impact on the software code and control.

As shown by the achieved results, TPM can improve security and software piracy protection, decrease dongle cost and it can be used for clients identification. Also, registering customer information at database of software provider such as (computer identification and other information...etc) enhances tracing the usage of the legal and illegal software. Results also show that TPMDWM can effectively prevent illegal copying or getting the software at installation and updating license. On the other hand, it is concluded that using database watermarking and other software watermarking is not only to prove the right owner by watermark but also to expand the usage of watermarking to prevent illegal usage of software. Finally, it is concluded that software providers can use AES and hash function to protect data transfer and all information between customer and software provider with the best security.

As a future work, there are several issues in software piracy protection that could be addressed. For example, how to use the best obfuscation method to protect the code from reverse engineer attackers. Secondly, a researcher can study the possibility to use TPM for software watermarking to prove software copyrights protection. Finally, a researcher can study the possibility of using TPM benefits to do hash functionality and database watermarking by TPM functions to produce machine signature and encrypt data for software copyrights protection.

8. References

[1] S. Mumtaz, S. Iqbal and E. I. Hameed, "Development of a methodology for piracy protection of software installations", 9th IEEE International Multitopic Conference (INMIC 2005), 2005, pp 1-7.

[2] H. Yanjun, Z. Huanguo, W. Lina, () "Secure spread-spectrum watermark detection based on extended TPM", First IEEE International Symposium on Data, Privacy, and E-Commerce (ISDPE 2007), 2007, pp 350-352.

[3] C.S. Collberg and C. Thomborson, "Watermarking, tamper-proofing, and obfuscation - tools for software protection", IEEE Transactions on Software Engineering, 2002, Vol. 28, No. 8, pp 735–746.

[4] Z. Shen and X. Wu, "The protection for private keys in distributed computing system enabled by trusted computing platform", International Conference on Computer Design And Applications (ICCDA), 2010, pp 6-12.

[5] W. Feng, "Concepts and Techniques in Software Watermarking and Obfuscation", PhD Thesis, University of Auckland, New Zealand, 2007

[6] X. Zhang, F. He, W. Zuo, "Hash function based software watermarking", IEEE Conference on Advanced Software Engineering and Its Applications (ASEA 2008), 2008, pp 95–98

[7] E. Vila, P. Borovska, "Data protection utilizing trusted platform module", International Conference on Computer Systems and Technologies (CompSysTech'08), 2008, pp 72–79

[8] R. Tu, F. Wang, J. Zhao, and A. El Saddik, "Copyright protection of Web applications through watermarking", IEEE First International Conference on Innovative Computing, Information and Control (ICICIC '06), 2006, pp 78-82

[9] M. Al-Mualla and H. Al-Ahmad, "Information hiding: steganography and watermarking", Multimedia Communication and Signal Processing (MCSP) Research Group, Etisalat College of Engineering, Sharjah, UAE, 2007.