Microcontroller based embedded approach to prevent software theft

RAJANI R V
PG Student,EEE Department
PSN College of Engineering and Technology
Tirunelveli, Tamil Nadu, India
rajanerv@gmail.com

RAJESH T
Assistant professor, CSE Department
PSN College of Engineering and Technology
Tirunelveli, Tamil Nadu, India
tnanjilrajesh@gmail.com

Abstract:-To avoid piracy and to build secured products the embedded industry is looking for a right solution like a Piracy Preventer. This project employs a hardware interface to prevent software from getting pirated. The proposed approach is based on challenge response in which server initiates the authentication process and identifies the genuine software copy. Server starts the process by sending the authentication challenge to client then client will response to it and server will verify the response. After getting expected reply from client, server will send status message to client by which client machine decides what should be done next. The Hardware interface is a plug and play device connected to the PC port and the license code is embedded using watermarking technique. The interface consists of an integrated encryption engine to aid to security policies. In order to discourage the program theft a digital watermark is embedded in to the software. Finally, a Real-time embedded finger-vein recognition system is used for the purpose of implementing more security. The system is implemented on an embedded platform and equipped with a novel finger-vein recognition algorithm.

Keywords: Digital Watermark, Embedded Systems, Finger-vein Recognition, Software Watermarking, Side Channel Attack.

1. INTRODUCTION

Detecting Software Theft project based on Embedded System. Embedded System can be defined as the combination of both software and hardware. An embedded system has specific requirements and performs pre-defined tasks unlike a general purpose personal computer. An embedded system is a combination of computer hardware and software and perhaps additional mechanical or other parts, designed to perform a dedicated function. Program memory protection mechanisms that prevent unauthorized read access to the program memory are frequently used in today's microcontrollers[2].

Hence, to gain access to the program code of an embedded system these protection mechanisms would have to be defeated rest. This makes testing embedded devices towards software plagiarism very difficult, especially if it needs to be done in an automated way. Furthermore, the person who is illegally using unlicensed code might apply code-transformation techniques[3] to hide the fact that he is using someone else's code. In this case, detecting the software theft is hard even if the program code is known. Software watermarks enable a verifier to test the program whether it was copied or not.

But in embedded applications, many programs are written in C or assembly. Furthermore, many previously proposed software watermarks have shown to be not very robust to code-transformations. In this paper, we introduce new and very efficient methods to detect software theft in embedded systems. To determine whether or not an embedded system is using our code, we use side-channel in formation that the device leaks out while the suspected code is running on the device. In this way, we avoid the need to gain access to the program or data memory[5]. The basic idea is to measure the power consumption of the device under test and uses these measurements to decide whether or not a piece.

As embedded devices are increasingly integrated into personal and commercial infrastructures, security becomes a paramount issue. For example, if a patient is wearing a heart-monitoring device that sends data wirelessly to a doctor, the embedded system must Keep this information confidential and deliver it uncorrupted to the doctor. An embedded network sensor monitoring water quality to prevent bioterrorism must have multiple methods to detect tampering in both hardware and software, lest an attacker bypass security measures and corrupt the water supply.

The design of security for embedded systems differs from traditional security design because these systems are resource-constrained in their capacities (and consequently in their defenses) and easily accessible to adversaries at the physical layer. Embedded security can't be solved at a single security abstraction layer, but rather is a system problem spanning multiple abstraction levels. We use an embedded biometric authentication device to demonstrate the necessity of addressing all levels of the security pyramid to ensure a fully robust and secure embedded system. Software watermarking involves embedding a unique identifier within a piece of software, to discourage software piracy. Watermarking does not prevent copying but instead discourages software thieves by providing a means to identify the owner of a piece of software and/or the origin of the stolen software [7]. The hidden watermark can be recognized or extracted, at a later date, by the use of a recognizer or extractor to prove ownership of stolen software [86]. It is also possible to embed a unique customer identifier in each copy of the software distributed which allows the software company to identify the individual that pirated the software.

For implementing better security biometric authentication scheme is used. There is a long list of available biometric patterns, and many such systems have been developed and implemented, including those for the face, iris, fingerprint, palm print, hand shape, voice, signature, and gait. Notwithstanding this great and increasing variety of biometrics patterns, no biometric has yet been developed that is perfectly reliable or secure. For example, fingerprints and palm prints are usually frayed; voice, signatures, hand shapes and iris images are easily forged; face recognition can be made difficult by occlusions or face-lifts [13]; and biometrics, such as fingerprints and iris and face recognition, are susceptible to spoofing attacks, that is, the biometric identifiers can be copied and used to create artifacts that can deceive many currently available biometric devices.

The finger-vein is a promising biometric pattern for personal identification in terms of its security and convenience . Compared with other biometric traits, the finger-vein has the following advantages [7]: (1) The vein is hidden inside the body and is mostly invisible to human eyes, so it is difficult to forge or steal. (2) The non-invasive and contactless capture of finger-veins ensures both convenience and hygiene for the user, and is thus more acceptable. (3) The finger-vein pattern can only be taken from a live body. Therefore, it is a natural and convincing proof that the subject whose finger-vein is successfully captured is alive.

We designed a special device for acquiring high quality finger-vein images and propose a DSP based embedded platform to implement the finger-vein recognition system in the present study to achieve better recognition performance and reduce computational cost.

The rest of the paper is organized as follows.In Section 2, we review the related works. In Section 3, Our approach for detecting software theft is explained.Section4 deals with the implemented technique for preventing software theft.In Section 5, we conclude the paper.

2. RELATED WORKS

2.1 PLAGARISM DETECTION WITH STRING MATCHING ALGORITHM

An efficient method that makes use of the fact that the power consumption of the microcontroller is related to the Hamming weight of the perfected opcode. This dependency allows us to map the power consumption of every clock cycle, which may consist of thousands of sampling points, to only one Hamming weight. The result is a high dimensionality eduction of the power traces. For subsequent analyses, string matching algorithms are applied for comparisons between original and suspicious software executions.

A closer look at the power consumption reveals that the prefetching mechanism is leaking the Hamming weight of the opcode[2] .Fig. 1 shows the power consumption of random instructions with different Hamming weights of the prefetched opcode recorded at a clock rate of 1 MHz. Right before the rising to the second peak, the Hamming weights of the prefetched opcodes are clearly distinguishable from each other.

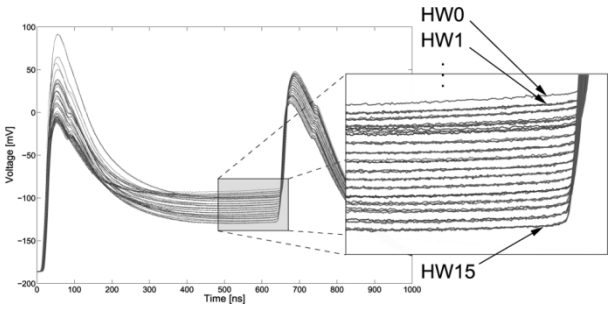


Fig. 1 Power consumption of instructions with different Hamming weights.

2.2 . Software Watermarking

Software watermarking involves embedding a unique identifier within a piece of software, to discourage software theft. Watermarking does not prevent theft but instead discourages software thieves by providing a means to identify the owner of a piece of software or the origin of the stolen software. It can then be extracted by an extractor or verified by a recogniser to prove ownership of software. The former extracts the original watermark, while the latter merely confirms the presence of a watermark.[9] A watermark recognition or extraction algorithm may also been classified as blind, where the original program and watermark is unavailable, or informed, where the original program and/or watermark is available.

It is also possible to embed a unique customer identifier in each copy of the software distributed which allows the software company to identify the individual that pirated the software. It is necessary that the watermark is hidden so that it cannot be detected and removed.

2.3. Side Channel Analysis

Side-channel analysis is a type of attack on a cryptographic system that utilizes the information unintentionally leaked from the real-world implementations of the cryptographic hardware via side-channels. These unintended side channels can include the instantaneous power consumption of the hardware, radiated electromagnetic fields or timing information leading to what are aptly named power analysis, electromagnetic analysis, and timing analysis, respectively[10]. Sometimes secrets such as plaintext can be discovered directly, but often the goal of the attacker is to determine the secret keys used to protect the data.

In one of the simplest cases, Simple Power Analysis (SPA), the bits of an important key might be seen directly in the power consumption of an integrated circuit using that key to perform an encryption or decryption operation.If the attacker can successfully detect the watermark using a side-channel analysis, the attacker also

gains the knowledge of the exact clock cycles where watermark instructions (e.g., the leakage generators) are executed.In this case, the attacker only needs to remove or alter these instructions to make the watermark detection impossible. Therefore, the watermark should only be detectable by the legitimate verifier who possesses the watermark secret.

3. PREVENTING SOFTWARE THEFT

3.1.Embedding Licence code

Software watermarking involves embedding a unique identifier within a piece of software, to discourage software piracy. Watermarking does not prevent copying but instead discourages software thieves by providing a means to identify the owner of a piece of software and/or the origin of the stolen software. Watermarking source code can be used to determine ownership of code and help prevent tampering of source code. The process of watermarking involves embedding secret messages into a cover message. In this case, embedding structures or data into an executable. The actual contents of a watermark differ, but are usually some identifier or copyright information. Examples could be a vendor ID number, company or author name, an number that is a product of primes.

The Software Watermarking problem can be described as follows: Embed a structure W into a program P such that: W can be reliably located and extracted from P even after P has been subjected to code transformations such as translation, optimization and obfuscation; W is stealthy, i.e. hard to tamper with. W also has a high data rate and embedding W into P does not adversely affect the performance of P. W has a mathematical property that allows us to argue that its presence in P is the result of deliberate actions.

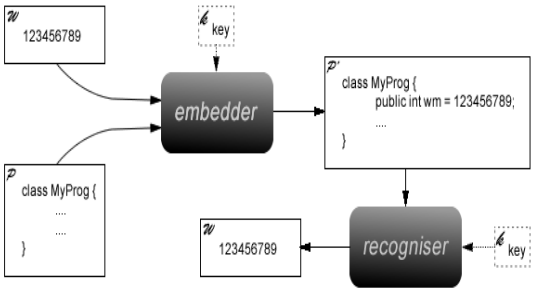


Fig:2 Embedding Watermark

The hidden watermark can be recognized or extracted, at a later date, by the use of a recognizer or extractor to prove ownership of stolen software. It is also possible to embed a unique customer identifier in each copy of the software distributed which allows the software company to identify the individual that pirated the software.

3.2. Sms Gateway:

An **SMS gateway** is a telecommunications network facility for sending or receiving Short Message Service (SMS) transmissions to or from a telecommunications network that supports SMS. Most messages are eventually routed into the mobile phone networks. Many SMS gateways support media conversion from email and other formats.

The proposed approach is based on challenge response in which server initiates the authentication process and identifies the genuine software copy. Server starts the process by sending the authentication challenge to client then client will response to it and server will verify the response. After getting expected reply from client, server will send status message to client by which client machine decides what should be done next.

On the software installation phase the user details are analysed and registered.The user details involve users mobile number and there personal details.Each time the user activates his software a password is generated and send to his mobile number.Through the login phase,the user details are verified and only the authenticated user is allowed to access the software.

3.3.Robustness and security analysis of software watermark

we have introduced our side-channel watermarks and showed that we are able to reliably detect the watermarks. However, so far we have not talked about the security of the watermark. Traditionally, the security of watermarks towards different attacks is called robustness. To the best of our knowledge, there does not exist a completely robust software watermark that can withstand all known attacks.

For software watermarks, and especially side-channel watermarks, it is very difficult to quantify the robustness of the watermark. We do not claim that our watermark is "completely robust" or secure—given sufficient effort, the side-channel software watermark can be removed. In the following, we will introduce our security model and describe some possible attacks against the system. We will provide arguments why these attacks can be nontrivial in practice. Hence, we will show that the watermark—although not impossible to remove—still represents a significant obstacle for attackers.

In the security model of the software watermark, three parties are involved: the owner of the watermark who inserted the watermark, the verifier who locates the watermark in a suspected device, and an attacker who tries to remove the watermark from a software code. The attacker has only access to the assembler code of the watermarked program. The attacker does not know the design of the combination function as well as what part of the assembler code implements this combination function and which internal states or constants are being used in this combination function.

This knowledge is considered the watermark secret. The verifier needs to be a trusted third party who shares the watermark secret with the owner of the watermark. A successful attack is defined as follows:A transformation of the watermarked software code that 1) will make it impossible for the verifier to locate the watermark with means of side-channel analysis and 2) does not change the functionality of the software program. Hence, an attacker was unsuccessful if either the verifier is still able to detect

the software watermark or the resulting software code does not fulfill the intended purpose of the program any longer. We will discuss three different attack approaches to remove the watermark from the assembler code.

Reverse-engineering attack: In a reverse-engineering attack,the attacker tries to locate the assembler instructions that implement the watermark using reverse-engineering techniques so that he can remove or alter these instructions.

Code-transformation attacks: In a code-transformation attack, the attacker uses automated code-transformations to change the original assembler code in a way that the resulting code is still functioning correct but the watermark detection is impossible.

Side-channel attacks: In a side-channel attack, the attacker tries to use side-channel techniques to locate the side-channel signal in the power consumption. This gives the attacker the knowledge of the location of some of the watermark instructions (e.g., the leakage generator).

4. Finger Vein Recognition

The finger-vein is a promising biometric pattern for personal identification in terms of its security and convenience.Compared with other biometric traits, the finger-vein has the following advantages: (1) The vein is hidden inside the body and is mostly invisible to human eyes, so it is difficult to forge or steal. (2) The non-invasive and contactless capture of finger-veins ensures both convenience and hygiene for the user, and is thus more acceptable. (3) The finger-vein pattern can only be taken from a live body. Therefore, it is a natural and convincing proof that the subject whose finger-vein is successfully captured is alive.

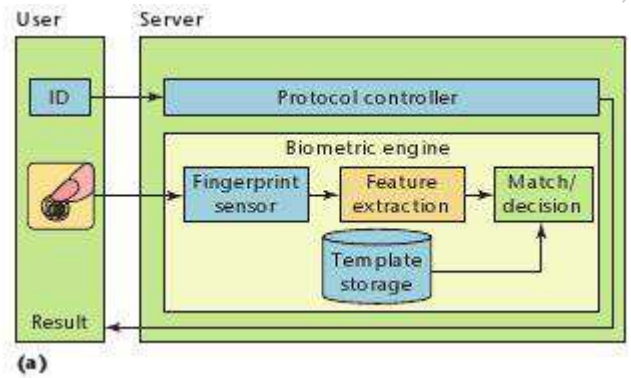


Fig 3: Finger Vein Recognition system working

The proposed finger-vein recognition algorithm contains two stages: the enrollment stage and the verification stage. Both stages start with finger-vein image pre-processing, which includes detection of the region of interest (ROI), image segmentation, alignment, and enhancement. For the enrollment stage, after the pre-processing and the feature extraction step, the finger-vein template database is built. For the verification stage, the input finger-vein image is matched with the corresponding template after its features are extracted. Fig. 3b shows the flow chart of the proposed algorithm. Some different methods may have been proposed for finger-vein matching.

To obtain high quality near-infrared (NIR) images, a special device was developed for acquiring the images of the fingervein without being affected by ambient temperature. Generally, finger-vein patterns can be imaged based on the principles of light reflection or light transmission. We developed a finger-vein imaging device based on light transmission for more distinct imaging.

Finger vein recognition is a recently proposed method in which finger vein patterns are analyzed. Based on the physical characteristics of the vein patterns, finger vein is unique and individual.Principal Component Analysis is a commonly used method for pattern extraction.In order to enhance the performance of the PCA Kernel Principal component analysis algorithm is designed.

5. CONCLUSION

In this paper, through implementation of watermarking algorithm in the software code for preventing the software theft and also hardware lock for protecting the software is not well efficient. They can be rewritable. To avoid such a situation for hacking the embedded software we have to provide security to the embedded system. Security can be implemented to the system in many ways but we have to ensure that it must not be hack able.

This paper focus on the security implementation through the cryptographic algorithm. AES which is one of the symmetric key algorithms which is used for higher security due to its computation complexity it is very difficult to decode. Also another method of providing security from pirating the software is to encode or watermark the secret license code inside the software so that if the code is hawked means we can easily detect it through correlation factor. Finger vein biometric authentication is used for further security of the embedded software program

REFERENCE

[1]. Alessandro Piva and Alessia De Rosa , "Correlation power analysis with a leakage model," in Proc. Cryptographic Hardware and Embedded Systems (CHES), pp. 16–29,Mar 2010.

[2]. Christian S. Collberg, and Clark Thomborson, " Watermarking, Tamper-Proofing, and Obfuscation Tools for Software Protection", IEEE Transactions on software engineering, Vol 8, No. 8,Nov 2002.

.

[3]. Ginger Myles and Christian Collberg , " Software watermarking through register allocation: Implementation, analysis, and attacks", In International Conference on Information Security and Cryptology,2003.

[4]. Collberg.C.S, Huntwork.A, Carter.E, Townsend.G, and Stepp.M, "More on graph theoretic software watermarks: Implementation, analysis, and attacks," Inf. Softw. Technol., vol. 51, no. 1, pp. 56–67,Jan 2004

[5]. Corcoran P. and Cucos A., "Techniques for securing multimedia content in consumer electronic appliances using biometric signatures,"IEEE Transactions on Consumer Electronics, vol 51, no. 2, pp. 545-551, May 2005

[5]. David D.Hwang, Patrick Schaumont, " A top-down, multi abstraction layer approach for embedded security", In International Conference on Information Security and Cryptology,Dec 2006.

[6].. Hamilton.J and Danicic.S , "An evaluation of static Java bytecode watermarking," in Lecture Notes in Engineering and Computer Science:Proc. World Congress on Engineering and Computer Science ,San Francisco, CA, pp. 1–8,May 2004

[7] Jain A. K., Pankanti S., Prabhakar S., H. Lin, and A. Ross, "Biometrics: a grand challenge", Proceedings of the 17th International Conference on Pattern Recognition (ICPR), vol. 2, pp. 935-942, 2004

[8]. Maria Chroni and Stavros D. Nikolopoulos, "Power Analysis Attacks Revealing the Secrets of Smart Cards", New York: Springer,2010.

[9]. Palsberg.J. Sowmya.K. and Minseok.k, "Experience with Software Watermarking", Inf. Softw. Technol., vol. 24, no. 1, pp. 556–567, May 2000.

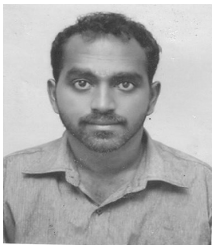
[10] Strobel D. and Paar C., "An efficient method for eliminating random delays in power traces of embedded software," in *Proc. Int. Conf. Information Security and Cryptololgy (ICISC)*, Seoul, Korea, 2011.

[11] Tarhio J. and E. Ukkonen, "Approximate Boyer-Moore string matching," *SIAM J. Comput.*, vol. 22, pp. 243–260, Apr.1999

[12].Ying Zeng, Fenlin Liu, Xiangyang Luo, and Chunfang Yang, "Side-channelbased watermarks for integrated circuits," in Proc. IEEE Int. Symp. Hardware-Oriented Security and Trust (HOST), pp. 30–35,Oct 2010.

ABOUT THE AUTHORS

Ms. R.V.Rajani, is a PG Student of Embedded System Technology in PSN College of Engineering and Technology, Tirunelveli. She is working in the area of Securing Embedded system under the guidance of Mr.T.Rajesh. She received her Bachelor's degree from Anna University, Chennai. Her research interests include Networking and Embedded System. (e-mail: rajanerv@gmail.com)



Mr.T.Rajesh, is a is working as Associate Professor in the Department of Computer Science and Engineering at PSN College of Engineering and Technology, Tirunelveli District, TamilNadu with more years of teaching experience. His research is centered on the Digital Image processing.(e-mail: tnanjilrajesh@gmail.com)