ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 4, October 2013

Tamper-proofing of Embedded System Software

J Ram Prabu, K Nandini

Abstract*— Though a wide range of methods and techniques have been innovated and practiced to protect software from piracy, illegal copying of softwares still prevails in the field. The anti-piracy techniques fall into three categories namely, technical, ethical, and legal methods. The ethical methods are meant to make piracy an unappealing activity. The legal methods threaten pirates about the legal consequences using copyrights, patents, and registrations. The technical methods including watermarking, obfuscation, encryption, and the like, are used to protect software from copying by means of improving difficulty associated with software duplication. This paper revolves around a technical anti-piracy methodology for Static Random Access Memory (SRAM) based Field Programmable Gate Arrays (FPGAs) that can be implemented using cryptographic authentication. This method ensures an easier and cost efficient security model for embedded system development using SRAM based FPGA.

Index Terms—**SRAM based FPGAs, SHA-1 message digest algorithm, Key-based Authentication, Secure Memory component, Intellectual Property protection.**

I. INTRODUCTION

Field Programmable Gate Arrays are semiconductor devices made up of a number of programmable and re-programmable functional blocks. These blocks can be configures as per user defined functions. This re-programmability and off-the-shelf availability makes FPGAs significant than Application specific Integrated Circuits (ASIC). SRAM based FPGA are easier to design and modify could be instantly prototyped and are economical for low-volume production. Other FPGAs such as anti-fuse and flash-based systems provide relatively secure methods for software protection. This is because, in these devices the configuration data is stored on the FPGA chip and there are mechanisms for preventing the data from being read out whereas in SRAM based FPGAs, once the data is loaded, it is held in SRAM cells that are prone to tampering and snooping [2]. Due to this two-chip solution (FPGA and memory), the configuration data bit-stream is exposed to eavesdropping during the power-up process. Since the FPGA is unaware of whether the bit-stream is genuine or pirated, the Intellectual Property (IP) contained in the configuration data is unprotected. The authentication method proposed in this paper is to identify genuine bit stream for given FPGA such that per device security can be achieved.

II. EXISTING TECHNICAL IP PROTECTION MECHANISMS

A. *Obfuscation*

This method makes the code difficult for reverse engineering [1][7]. This method requires strong technical knowledge on either the data structures or code execution flow in order to make these parameters difficult for analysis.

B. *Fingerprinting and watermarking*

This method integrates a digital proof of ownership to the software by using which the owner of particular software can claim against the software pirate [6][10]. However this method is always a post-system mechanism and the software is still prone to piracy.

C. *Encryption*

Encryption techniques are used for protect software or data from piracy. And also the FPGAs are suitable candidates for implantation of ciphers. This is because they combine the speed of hardware with the flexibility of software [3][4]. The efficiency of the cipher mechanism depends upon the strength of the cryptographic algorithm, key, execution time and memory footprint of the cipher used [4][5]. However these algorithms are associated with execution overhead.

III. AUTHENTICATION MECHANISM

Due to processing overhead presented in implementation of cryptographic algorithms the portions, which are to be encrypted and decrypted, can be reduced. One of such minimized encryption methodologies is called partial encryption where the part of a code or data that is more crucial is alone accessed by the cipher unit [3]. One of such mechanism is used in the proposed system. In this method the cryptography is used only for finding out whether the FPGA and the memory are authenticated. That is, to identify whether one of aforementioned entity is either a friend or foe to the other. If the entities are identified to be friends, the configuration data is transferred and the system is executed. On the other hand, if either of the components is not genuine or authenticated, the configuration data is not transferred across the system entities.

IV. CIPHER USED

Message Digest Algorithm SHA-1 (Secure Hash Algorithm) is used for the system implementation. SHA-1 is one of the candidates from U.S Institute of Standards and Technology (NIST) [8]. This is a hash function that takes as input a message of arbitrary length and returns a digest or Message Authentication Code (MAC).

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 4, October 2013

A. *System workflow*

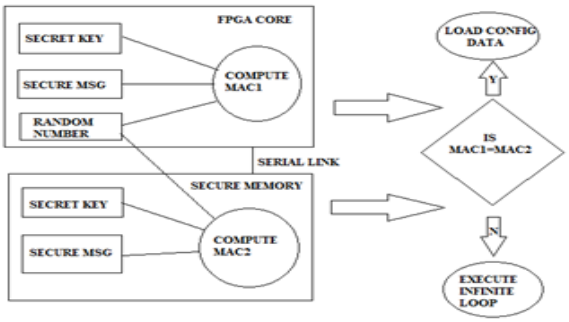

Fig. 1 System workflow diagram

Fig. 1 shows the overall system execution process.The FPGA core and a secure memory such as an EEPROM chip are implemented along with the configuration memory of the system. The term secure memory is used to indicate the significance of strong protection implemented for the EPROM memory. This is where the cryptographic key is to be stored. The same key is also stored in FPGA core during fabrication. A random number is generated in FPGA and it is transmitted to secure memory. Likewise a secure message stored in the secure memory is transferred to FPGA prior to authentication process. The same MDA BLAKE is implemented in both the aforementioned system components. On power up of the system, the MDA implanted on the FPGA and the secure memory executes using the key, message and the random number. FPGA generates a MAC say, MAC1 whereas the secure memory generates its own MAC say, MAC2. MAC1 and MAC2 are now given as input to the authentication core implanted in FPGA. If both the codes match, the environment is found to be free from tampering and the configuration data is loaded on to FPGA and the application is executed. On the other hand, when the authentication codes do not match, then the system identifies the FPGA to be unauthenticated. The significance of using random number in the authentication core is to protect system from replay attacks.

B. *System Architecture*

The system consists of the FPGA, configuration memory and the secure memory. These are interconnected via serial links. The authentication core is the component where the MACs are compared for the proof of identity [Fig.2].

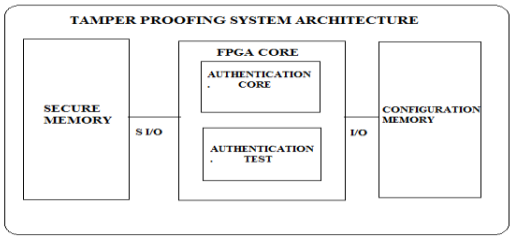

Fig. 2 System structure and components

V. IMPLEMENTATION OF SHA-1

The Secure Hash Standard (SHS) describes the following algorithms: SHA-1, SHA-256, SHA-385, SHA-512, where the number indicates the length of the MAC or hash. These algorithms are normally used in public-key cryptography, especially in message authentication schemes. SHA-1 is used for computing a condensed representation of a message which is called the hash of the specific message. This algorithm provides an exceedingly high level of security since it is computationally infeasible to arrive at a message that corresponds to a given message digest. The slightest change to a message will result in a different message digest, causing signature verification to fail. SHA-1 is an iterative hash function using a compression function as its building block, where the input consists of 160 bit chaining variable and a 512 bit message block. The output is an update of the chaining variable. The compression function consists of 5 rounds, each made up of 20 steps. Finally, a feed forward adds the initial value of the chaining variable to the updated value. Every round uses a particular non-linear function [Table.1], and every step modifies one word of the chaining variable and possible rotates another.

The algorithm step function is implemented as follows

$$A = A + B << 5 + f\ (C, D, E) + x1 + k1\ ;\ C = C <<< 30$$

Where <<< denotes left rotate, A, B, C, D, and E are the words of the chaining variable, k1 and k2 are constants, f () represents any of the functions described in Table.1

TABLE I: SHA 1 IMPLEMENTATION OF NON-LINEAR FUNCTIONS

Step	Non-linear function used
1	(x & y) | ((~ x) & z)
2	x ^ y ^ z
3	(x & y) | (x & z) | (y & z)
4	x ^ y ^ z

Table.1 represents various round functions used in the implementation of SHA-1 algorithm where x, y, z denote the bitwise operands from the secret key, secure message, and the random number, "^" denotes inversion and "~" denotes XOR operation [Fig.1]

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 4, October 2013

VI. CONCLUSION

In this paper, a security key based authentication mechanism is proposed for protection of Intellectual Property of FPGA based embedded systems. This key based model is easier to implement and is of relatively lesser process overhead when compared to other cryptographic methods. The secure EEPROM chip with built in cryptographic engines are available commercial off-the-shelf. Hence rapid prototyping of such a model can be made possible.

REFERENCES

[1] Gareth Cronin, A Taxonomy of methods for Software Piracy Prevention Colberg, C. S & The Saar Drimer, "Volatile FPGA design security- a survey" http://www.cl.cam.ac.uk/~sd410, April 17, 2008

[2] Kun- Wah Yip, Tung-Sang Ng, "Partial Encryption Technique for Intellectual Property protection of FPGA based products," in IEEE transactions on Consumer Electronics, Vol.46.No.1, Febrauary 2000.

[3] Jun Yang, Lan Gao and Youtao Zhang, "Improving Memory Encryption Performance in Secure Processors" IEEE transactions on Computers, Vol.153, No.5, May 2005.

[4] Michael A. Nenashev, "Cryptography Based Tamper-Resistant Software Design Mechanism," U.S Patent 6 976 167 B2, Dec.13, 2005

[5] Brian T. Witten, Reston, VA(US), "Method and Apparatus for Identifying Invariants to Detect Software Tampering," U.S Patent 8 108 931 B1, Jan 31 2012.

[6] "A Survey of Recent Results in FPGA Security and Intellectual Property protection", UCL Crypto Group, Switzerland

[7] "Hash algorithms for Cryptographic Protocols: FPGA Implementations," TELFOR'2002, Nov 26-28 2002

[8] J. Padhye, V. Firoiu and D. Towsley, "A Survey on Protection of FPGA Based IP Designs," ISSN (Print): 2278-8948, Volume -2, Issue 2, 2013

AUTHOR'S PROFILE

Prof. J. Ramprabu is working in the department of EEE of Kumaraguru College of Technology, Coimbatore, India for the past six years. He is a Life Member in Indian Society for Technical Education (India), System Society of India. His research areas include Renewable Energy, Solar Energy and Embedded systems.

K. Nandini is pursuing Master of Engineering (Embedded Systems) in Kumaraguru College of Technology, Coimbatore, India. Her area of interest includes embedded systems, Cipher security.