METHOD OF USING BLOCKCHAIN FOR HARDWARE DEVELOPMENT PROCESS

by

Mudassir Ejaz, BS

THESIS

Presented to the Faculty of Computer Engineering

The University of Houston-Clear Lake

In Partial Fulfillment

Of the Requirements

For the Degree

MASTER OF SCIENCE

in Computer Engineering

THE UNIVERSITY OF HOUSTON-CLEAR LAKE

MAY, 2022

METHOD OF USING BLOCKCHAIN FOR HARDWARE DEVELOPMENT PROCESS

by

Mudassir Ejaz

APPROVED BY

_____

Ishaq Unwala, Ph.D., Chair

_____

Xiokung Yang, Ph.D., Committee Member

_____

Jiang Lu, Ph.D., Committee Member

RECEIVED/APPROVED BY THE COLLEGE OF SCIENCE AND ENGINEERING:

_____

David Garrison, Ph.D., Associate Dean

_____

Miguel A. Gonzalez, Ph.D., Dean

Dedication

I dedicate this thesis to my parents. I am obliged to my parents and siblings, who have always been there for me whenever I needed them. Their unconditional love and support motivates me to set higher targets. May Allah bless all of them. Ameen.

Acknowledgements

All praise is to Almighty Allah who bestowed upon me a minute portion of His boundless knowledge by virtue of which I am able to accomplish this challenging task.

I am greatly indebted to my supervisor Dr. Ishaq Unwala, without his priceless supervision, advice and invaluable guidance, completion of this thesis would have been doubtful. I am deeply thankful to my thesis committee members Dr. Xiokung Yang and Dr. Jiang Lu, for their encouragement and support during this work.

Also, I would like to express my deepest gratitude and blessings to everyone who has helped me in any respect, for the completion of this thesis.

ABSTRACT

METHOD OF USING BLOCKCHAIN FOR HARDWARE DEVELOPMENT PROCESS

Mudassir Ejaz
University of Houston-Clear Lake, 2022

Thesis Chair: Ishaq Unwala, Ph.D.
Committee Member: Xiokung Yang, Ph.D.
Committee Member: Jiang Lu, Ph.D.

Modern microprocessors contain many millions of gates and finding any hidden malicious Hardware Trojan (HT) is difficult. Additionally, HTs may not need any additional external input pins to activate. Multiple solutions have been proposed to find these HTs, but none of these solutions have promising result due to their limitations. Moreover, pre-silicon verification and post-silicon testing don't address the issue of HTs. In this thesis we present a methodology based on blockchain technology to limit the possibility of inserting HTs into the design. Blockchain based technology limits the opportunity for insertion of HT, avoiding them in the design and fabrication process. We proposed a solution by monitoring hardware development process files, and maintaining integrity and trustful relationship using encryption and smart contracts in a blockchain network.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I:

INTRODUCTION

Hardware devices, integrated circuits (IC), including microprocessors, were considered more secure as to subterfuge than software packages, but now this rational has changed with discovery of Hardware Trojans (HT). Many suspicious incidents related to hardware bugs and hardware Trojans have been reported by IEEE Spectrum [2]. The US Defense Science Board [1] is so concerned that a task force about HTs has been created. This thesis defines HT as basically undesired alterations and possibly malicious changes to a hardware design that manifest themselves in the long run. The globalization of hardware design and fabrication process provides an opportunity for an adversary to perform malicious activities and alterations. These types of alterations are essentially designed to either disable, weaken the system, or leak confidential information.

Recently, hardware security has become a significant research topic. Hardware security research mainly comprises of the study regarding threats to hardware devices and components, vulnerabilities, detection and their prevention. As compared to software packages, hardware devices and components were considered more trusted and secured. However, due to the increasing number of failure incidents caused by the hardware components [1][2], finding the root cause becomes more important. One of the reasons for concern is that, most of the hardware fabrication foundries are now global and adversaries can take advantage of the situation and add some malicious circuits during the final design or fabrication process.

In the literature, researchers have proposed and developed many methodologies for detecting Trojans at pre-silicon and post-silicon stages but the results are not promising [3]. They are mainly categorized in two groups; logic testing by Trojan activation and the side channel analysis techniques [4]. The Trojan activation strategies

are more focused on activating the malicious circuitry as quickly as possible using probabilistic approaches. On the other hand, in side channel analysis techniques, real-time power and timing information is collected to detect unwanted circuitry. Also, these techniques require visibility of internal structure of microprocessor.

There are few other approaches for detection of malicious circuitry, one is through comparison of side channel parameters with Golden IC. Such type of comparison could be done in ideal cases where Golden IC is available. However, if multiple foundries and third-party intellectual properties (IPs) are involved in design process, this solution is not applicable [9]. Destructive reverse-engineering schemes are also one solution to check the integrity of manufactured microprocessors. Although, this approach gives relatively promising results but, it's much time consuming and expensive. And in unconventional hardware designs, reverse-engineering could be extremely challenging.

The involvement of third parties in manufacturing process with weak or non-existent integrity and trust mechanisms creates a loophole for adversaries to exploit. These trust and integrity issues in ICs and microchips manufacturing process can be addressed by cryptography and blockchain technology. The concept of blockchain technology was first introduced in 1990 by Stuart Haber and W. Scott Stornetta [5]. They laid the foundation for a cryptographically secured chain what is now known as the blockchain.

Blockchain technology has wide range of applications across multiple fields including banking and financial services, internet of things (IoT), healthcare sector, business and many more. This blockchain technology has the potential to be used in any domain due to its flexible architecture that supports security and integrity and its structure that makes it difficult for any fraudulent record to be created [6]. Fundamentally, the blockchain is the chain of records that are created whenever there is any transaction in the

blockchain network. Additionally, these transactions are temper-proof, meaning that transaction records can't be deleted or erased once they are part of the chain.

Blockchain is a distributed ledger that is stored in peer-to-peer network. Whenever a transaction is performed, a block containing specific information (date, time, block id and hash etc.) is created. The block then propagates to every peer of the network. The basic idea of blockchain is relatively straightforward, it is based on the distributions of records to a large group of recipients. By doing so, there is always a readily available source through which integrity of the records can be verified.

Considering the limitations of proposed schemes in literature for ensuring the logical and functional integrity of ICs, a promising and efficient solution is imperative. Consequently, in this thesis, a preventive approach is proposed that can limit the opportunity to affect the integrity of hardware designs and microprocessors. Our proposition is mainly based on smart contracts and blockchain. We used cryptographic algorithms to secure hardware design process files and monitor the reliability of entire development lifecycle. The main contributions of this study are listed below.

- Actual data files are secured by using encryption and hashing algorithms
- Encrypted files are stored in a distributed database
- Version control system is used to have updated files in the database
- Storing hash in the blockchain
- Audit and verification of integrity of files

The organization of the remaining thesis is as follows: Chapter II presents the related works on HT detection and role of blockchain in data security. In Chapter III, we discuss the problem statement. Chapter IV explains our proposed solution in detail and Chapter V concludes this work.

CHAPTER II:

RELATED WORK

This section discusses existing research regarding detection of HTs and blockchain.

A: Protocols for Detecting Hardware Trojans

Multi-parameter side channel analysis technique for detecting HT is proposed in [3]. As opposed to conventional side channel analysis techniques, the authors have considered multiple parameters. They correlated quiescent current (I_{DDQ}) to the maximum operating frequency (F_{max}) of the circuit for detecting intrinsically leaky ICs. Also, they used power gating techniques to decrease the background power which as a result improves the SNR. The proposed scheme was validated through simulations and hardware testing using a 120nm FPGA chip.

For detecting HT, a new methodology based on thermal maps and inception neural networks (INN) is proposed in [7]. Authors have exploited side-channel leakages using thermal maps. The temperature distribution and voltage noises of Trojan infected (TI) chip are distinguishable compared to those of Trojan free (TF) chip. Additionally, Lagrange multiplier and Karush–Kuhn–Tucker (KKT) conditions are used to optimize the sampled thermal maps. As a result, optimum input data for training INNs is obtained. Also, INNs with assistance of customized filters outperforms conventional convolutional neural network (CNN). Simulation results show that 98.2% accuracy can be achieved for detecting HTs.

A neural network based learning-assisted scheme, named AVATAR, for detecting HTs is proposed in [8]. In this methodology, authors performed side channel analysis without using Golden IC (Trojan Free IC). A neural network is trained at design time using Static Timing Analysis (STA) data and later used as a process tracking watchdog.

Moreover, authors have considered different types of process variations while training neural network in order to increase the probability of HT detection. It doesn't require Trojan to be partially or full activated to be detected as an inactive malicious circuitry can still add delay. The results show that AVATAR was able to detect ~98% of HT over selected benchmarks.

The need for low-cost IC and outsourcing chip production supply chain has created security issues concerning HT, which is deliberate malicious modifications in the original design. Extensive research has been done using side-channel parameter analysis for finding HT. However, a significant part of the research depends on a golden chip which can be used for comparison. Similar to the human brain, Faezi et al., [10] have suggested Hierarchical Temporal Memory (HTM) for HT detection. This approach circumvents the requirement for a golden chip by employing a self-referencing approach. The method was found resistant to natural fluctuations in side-channel measurements, yet being able to properly identify anomalous chip activity when the HT is activated. Additionally, through TrustHub standards, on average observed detection accuracy was 92.20 %.

The globalization of the semiconductor industry has raised the security concerns associated with fabrication and authenticity of ICs. The secrecy of information in communication technologies and electronic systems rely on both software and hardware. There was a lot of work done to locate HT in order to build a secure and reliable piece of hardware. Nevertheless, there are limitations in the methodologies used in combinatorial testing, requiring a unique approach. Recently, an approach has been proposed that solely depends on the combinatorial qualities of the test vector and test results [11]. Authors claim that the proposed scheme gives promising result for finding combinational HTs under some particular assumptions. Furthermore, their method detected HTs that were

integrated in an Advance Encryption Standard (AES) circuit with a key length of 128 bits. Additionally due to its algorithmic specification, it may be used to evaluate various cryptographic circuits.

B: Using Blockchain for Data Integrity

Authors in [6] have proposed an integrity monitoring protocol for the files in the cloud. The architecture of proposed protocol comprises of four entities; client, cloud storage service (CSS), integrity check service (ICS) and blockchain network (BN). Authors further divided the scheme into three phases for simplicity; preparation phase, storage phase and integrity check phase. Moreover, clients can monitor file integrity in such a way that their file content won't be exposed to third parties involved in the process. Authors also proposed a security mechanism for CSS, that is, after verification of rules defined in trust management contract, only then CSS will work.

Stuart Haber and W. Scott Stornetta [5] are the pioneers of digital time-stamping a document which then became the foundation of bitcoin and blockchain. In the author's viewpoint, there is a need of securing the data, not only the medium. So, they proposed a solution of time-stamping a digital document while maintaining the privacy and trust of client and addressing the storage space issue as well. The authors proposed a time-stamping service (TSS), which mainly tackles two issues; first, time-stamping a document and second, making sure that digital signature (date and time stamp) must not be forgeable. For addressing the storage space issue, authors first use the hash algorithm which compress the arbitrary length strings (data) to fixed length string (hash). It is not practical to send and store large size document to TSS.

A data revocable and monitor-able P2P file sharing system based on smart contracts and blockchain is proposed in [10]. In conventional P2P file sharing systems, once the data is shared, the owner has no control over the data. Considering General Data

Protection Regulation (GDPR), authors have proposed a solution that improves privacy and security of P2P file sharing system. Moreover, provides control over revocation of shared files to the owner. By using Inter-Planetary File System (IPFS) synchronization status, the proposed system ensures that every node has updated files.

Secure data sharing in patient monitoring system is very critical. Traditionally, data-sharing protocols mainly rely on third-party systems, which can put the system's integrity at risk. A smart contract based secure data sharing scheme is proposed to address the security and integrity issues in healthcare 5.0 [12]. Access control policy is written in a smart contract and after the verification of rules, transaction takes place on completion of treatment. In order to test the proposed solution, authors have used Solidity programming language for writing smart contracts, Ethereum blockchain and decentralized storage. And achieved data integrity and confidentiality in healthcare 5.0.

With the revolution of software techniques, captured videos can be tampered very easily. In [13], a blockchain based scheme is proposed for surveillance systems to retain the integrity of videos and manage the storage space efficiently. A hash is created from the original captured scene, and stored as the reference along with its frame. The first block in the blockchain contains hash, frame name and timestamp of the reference scene. Then after predefined interval, hash is created from newly captured frame and is compared with the hash of reference frame. If the two hashes are same, that means there is no difference between newly captured frame and the reference frame. So, the newly captured frame is deleted for saving the space and few of its attributes such as hash value, date, time and frame name are stored in CSV file for future use. However, if the two hashes are not identical, that means some new information is captured. Then, its hash, captured time and frame name is stored on the blockchain. The proposed solution provides integrity and efficient storage management for surveillance systems.

CHAPTER III:

PROBLEM STATEMENT

Due to economic reasons, most hardware manufacturers have outsourced their IC fabrication to inexpensive facilities in foreign countries like Taiwan, China and South Korea etc. By doing so, the cost of IC fabrication is reduced considerably but, it may enable the adversaries to compromise the integrity of IC. In Figure 1, a simple crypto device is shown. Inside the device, there is a crypto module and a secret key is stored. So, whenever an input is given to the IC, it is fed to the crypto module along with the key. And then at the output pin, ciphertext is generated for the given input. Moreover, we can say that, as long as the key inside the IC is well concealed, the ciphertext at the output doesn't contains any information regarding the secret key.

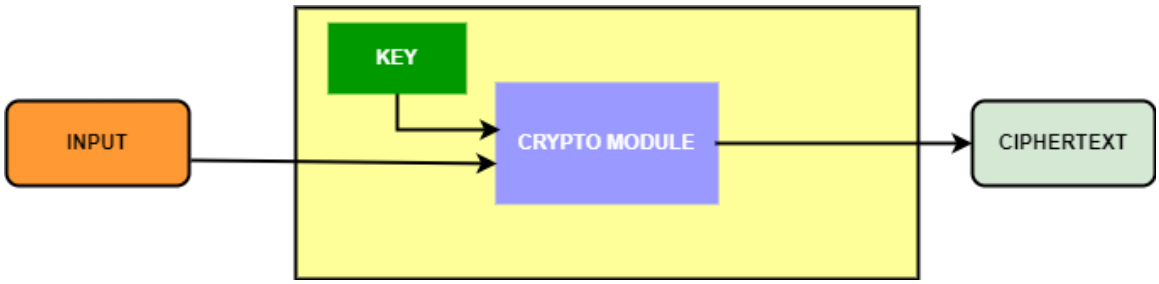


Figure 1

Crypto Device

Now, another crypto device is shown in Figure 2. Let's assume that during the fabrication process, an adversary has added a HT which consists of a trigger and a 2-to-1 multiplexer. Secret key and output from the crypto module is connected to the inputs (In_0, In_1) of mux respectively and a trigger is connected to the select line (S). The functionality of mux is simple, depending on the value of the select line it passes on either of the inputs to the output. In a given scenario, we assume that, In_0 is 1 and In_1 is 0. So, whenever the

trigger feeds 0 to the select line, ciphertext is passed on to the output pin. Similarly, when the select line is 1, it passes on the secret key to the output pin. Hence, the key which is concealed inside the chip is now visible at the output. So, in this example, we have seen that, by injecting this type of small HT in an IC, the secret and sensitive data can be exposed to the output.

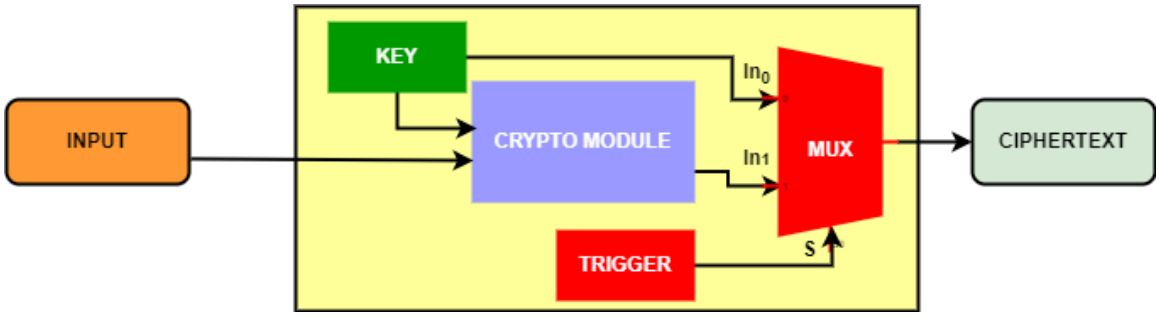


Figure 2

Compromised Crypto Device

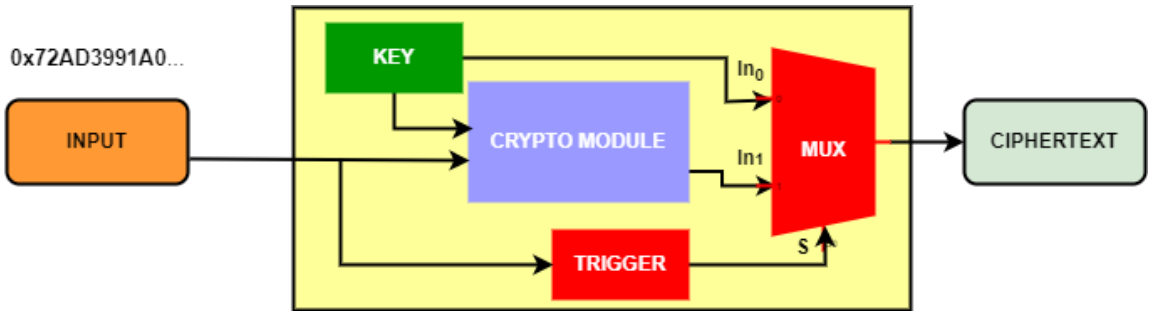


Figure 3

Compromised Crypto Device (Trigger is connected to device input pin)

Let's consider another scenario shown in Figure 3. In this figure, the trigger is listening to the input of IC. This is an example of combinational Trojan which uses a

specific sequence of code to trigger. That means, the trigger remains in the passive mode and waits for the specific input (e.g. 0x72AD3991A0…). When the sequence appears, the trigger generates an output of 1 and as a result, the secret key is passed to the output. It is very difficult to detect these types of combinational Trojans, as the specific code is only known to the attacker.

The HTs are designed in such a way that they are extremely difficult to be detected by functional testing, if not impossible and yet they are capable of causing disastrous damage. So, according to our point of view, if the hardware development supply chain can be secured, the integrity and vulnerability issues can be mitigated.

CHAPTER IV:

PROPOSED SCHEME

In order to maintain the integrity and securing hardware development process, the proposed solution uses cryptography and blockchain technology. As many phases in hardware development process are outsourced due to cost and technical reasons so, the process is vulnerable and prone to security risks. Moreover, the blockchain technology has emerged as a promising solution for security and integrity in many fields like cryptocurrency, banking, IoT and healthcare.

This thesis aims at integrating blockchain and cryptographic techniques with hardware development process to provide

- Data security against HTs
- Transparency
- Integrity and trustful relation with untrustworthy third parties

The objectives of this study are listed below.

- Securing the actual data by encrypting and hashing the files after completion of each phase in hardware development lifecycle
- Storing encrypted files in a distributed database
- Use of version control system to have updated files in the database
- Creating the hash from the original data
- Storing hash in a distributed ledger to form a blockchain
    - Ethereum technology for implementation
    - Ganache personal Ethereum blockchain is used for testing
- Audit and verification of integrity of files and entire development lifecycle

In the hardware development process, many different types of files are generated after the completion of each step. The hardware development process is shown in Figure 4.

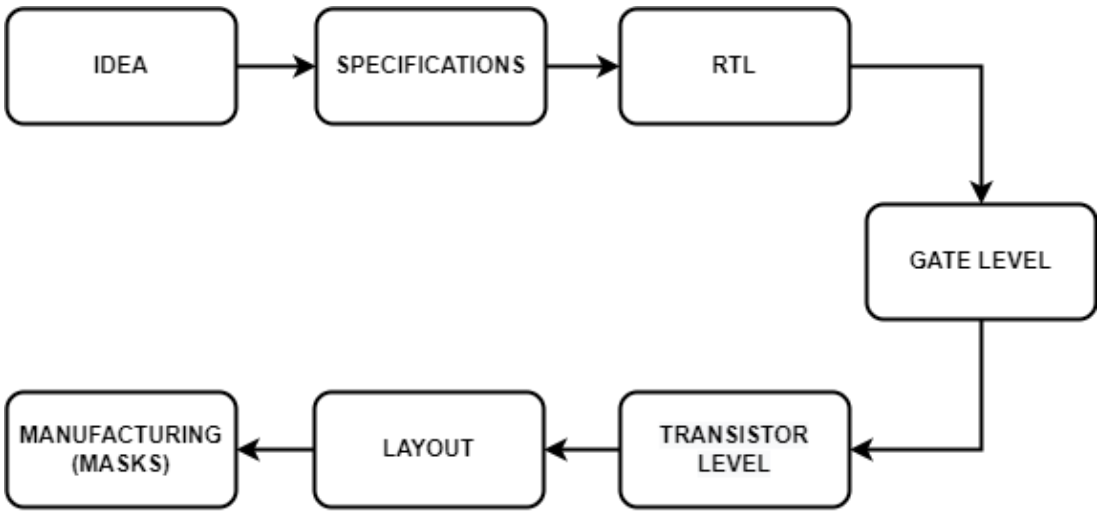


Figure 4

Hardware Development Process

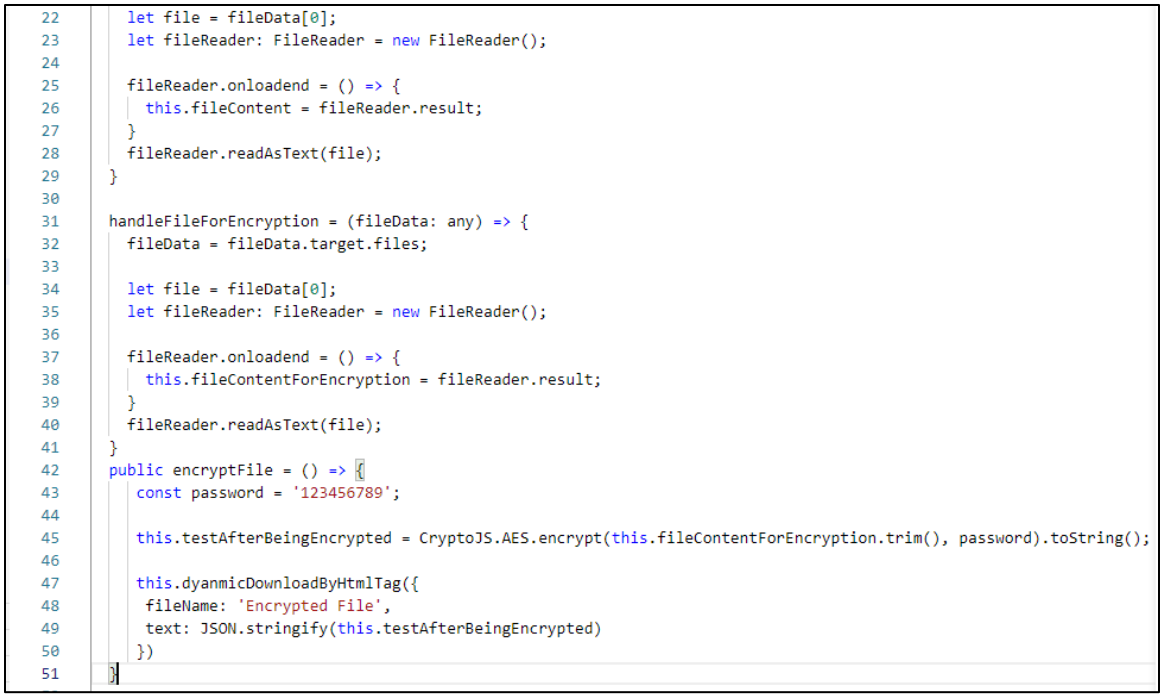
In the proposed solution, all files after a successful completion of a step are encrypted, and then stored on the separate distributed database. To create a hash from original file, a cryptographic SHA-256 hash algorithm is used. This algorithm takes an arbitrary size string as an input and generates a unique fixed 256-bit hash. The algorithm generates different hash even if only one symbol is changed. A small windows application has been developed that allows user to select a file, and then that application generates its hash. A code snippet for the hash function is shown in Figure 5.

```
11  export class AppComponent {
12    title = 'fileEncryption';
13    fileContent: any = '';
14
15    // Read file data ...
16    public handleFile = (fileData: any) => {
17      fileData = fileData.target.files;
18
19      let file = fileData[0];
20      let fileReader: FileReader = new FileReader();
21
22      fileReader.onloadend = () => {
23        this.fileContent = fileReader.result;
24      }
25      fileReader.readAsText(file);
26    }
27
28    // Generate file hash and download the file as text...
29    public generateHash = () => {
30      of('0X'+shajs('sha256').update(this.fileContent).digest('hex')).subscribe((res) => {
31        this.dyanmicDownloadByHtmlTag({
32          fileName: 'File Encryption Hash',
33          text: JSON.stringify(res)
34        });
35      });
36    }
```

Figure 5

Code Snippet (SHA-256)

When a user selects a file, its path is stored in a variable and then the same path is used for storing the hash.



```
    let file = fileData[0];
    let fileReader: FileReader = new FileReader();

    fileReader.onloadend = () => {
      this.fileContent = fileReader.result;
    }
    fileReader.readAsText(file);
  }

  handleFileForEncryption = (fileData: any) => {
    fileData = fileData.target.files;

    let file = fileData[0];
    let fileReader: FileReader = new FileReader();

    fileReader.onloadend = () => {
      this.fileContentForEncryption = fileReader.result;
    }
    fileReader.readAsText(file);
  }
  public encryptFile = () => {
    const password = '123456789';

    this.testAfterBeingEncrypted = CryptoJS.AES.encrypt(this.fileContentForEncryption.trim(), password).toString();

    this.dyanmicDownloadByHtmlTag({
     fileName: 'Encrypted File',
     text: JSON.stringify(this.testAfterBeingEncrypted)
    })
  }
```

Figure 6

Code Snippet (AES Encryption)

Similarly, AES algorithm is used to encrypt the original files and then encrypted files are uploaded to the database. Figure 6 shows the code snippet for AES function. Additionally, the detailed flow diagram of the proposed solution is shown in Figure 7.

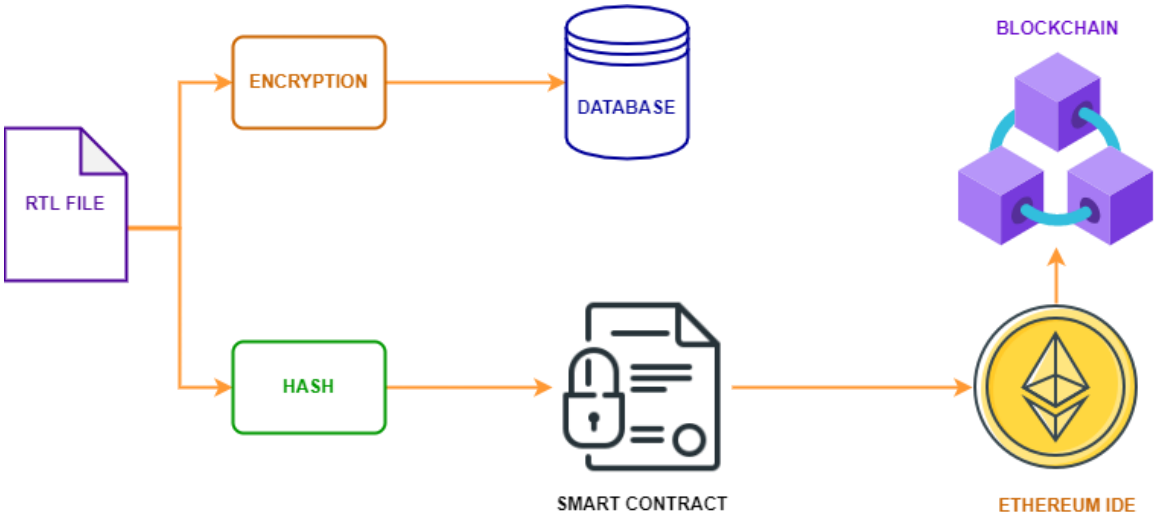


Figure 7

Flow Diagram

Consider Figure 7, for an instance, a hash is generated using SHA-256 algorithm for a single RTL file. Similarly, the original file is encrypted and then stored on a database. For testing the proposed solution, Solidity programming language is used for writing smart contract and personal Ethereum blockchain (Ganache) as a distributed ledger. The already created hash of RTL file is then stored in the smart contract. In order to compile and deploy the smart contract, Remix Ethereum IDE is used.

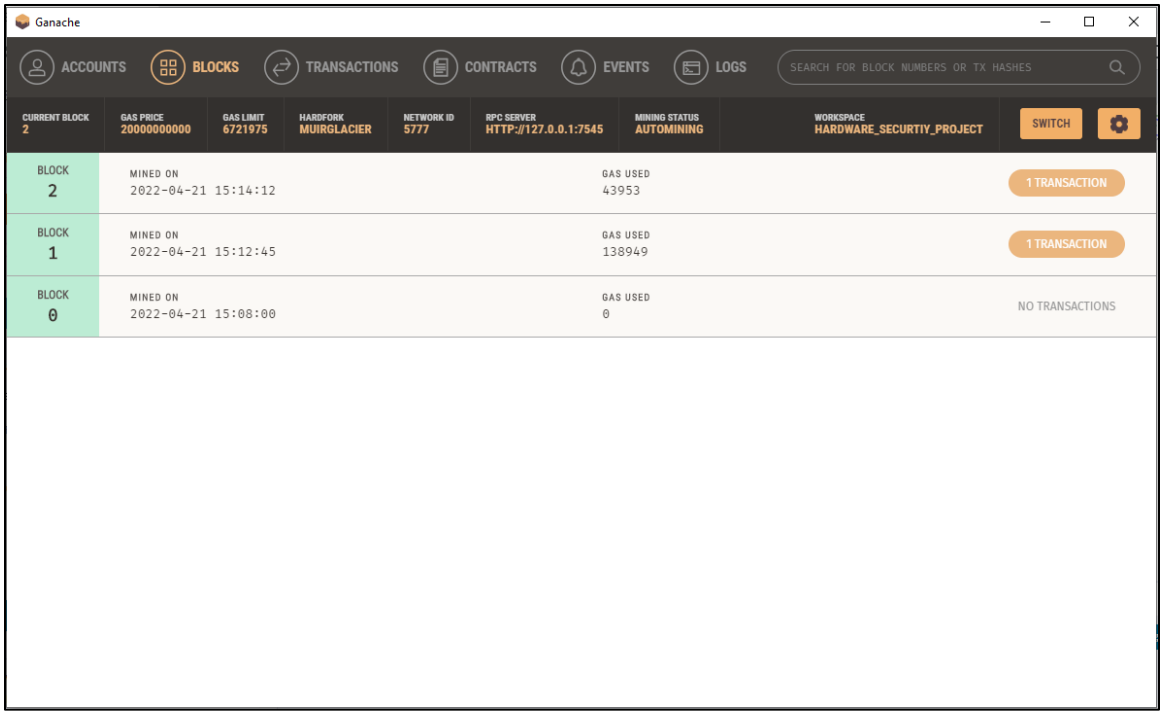


Figure 8

Blockchain

Whenever a hash is stored in the smart contract, a transaction takes place and a new block is added to the blockchain. Figure 8 shows an example of such transaction. Moreover, part of this thesis has been published in IEEE conference [14].

A hardware design usually have hundreds of RTL files, associated with some specific function and different teams work on those files simultaneously. Then multiple RTL files are synthesize to generate gate level. In the proposed solution, hashes of RTL files for every single functionality are stored in separate blockchain. This means that, there will be multiple blockchains for RTL files. Additionally, each stage of the hardware development process has its separate blockchain.

In the blockchain network, transactions can only be performed by verified users (nodes) that are part of the blockchain network. For joining a blockchain network, a user needs a security phrase for authentication purpose. In case of public blockchain, it is

strongly recommended that only known users receive the security phrase. Furthermore, authorized nodes can add other nodes to the blockchain network. Once the information is stored on the blockchain, it can never be erased. The blockchain keeps the record of every single transaction ever made and it's verifiable.

Moreover, actual files will be encrypted first and then stored at the database server. In the proposed scheme, a version control system is used for storing the updated files in the database. Whenever the updated file is pushed to the version control system, whole process of encryption, moving the encrypted files to the database and storing the hash to the digital ledger will be performed again. This process can be easily automated for convenience of the users of the system. If any unauthorized changes are made to the files in database, the hash from the blockchain will differ and the change will be exposed.

Working of Proposed Scheme

In this subsection, the working of proposed solution is explained in detail.

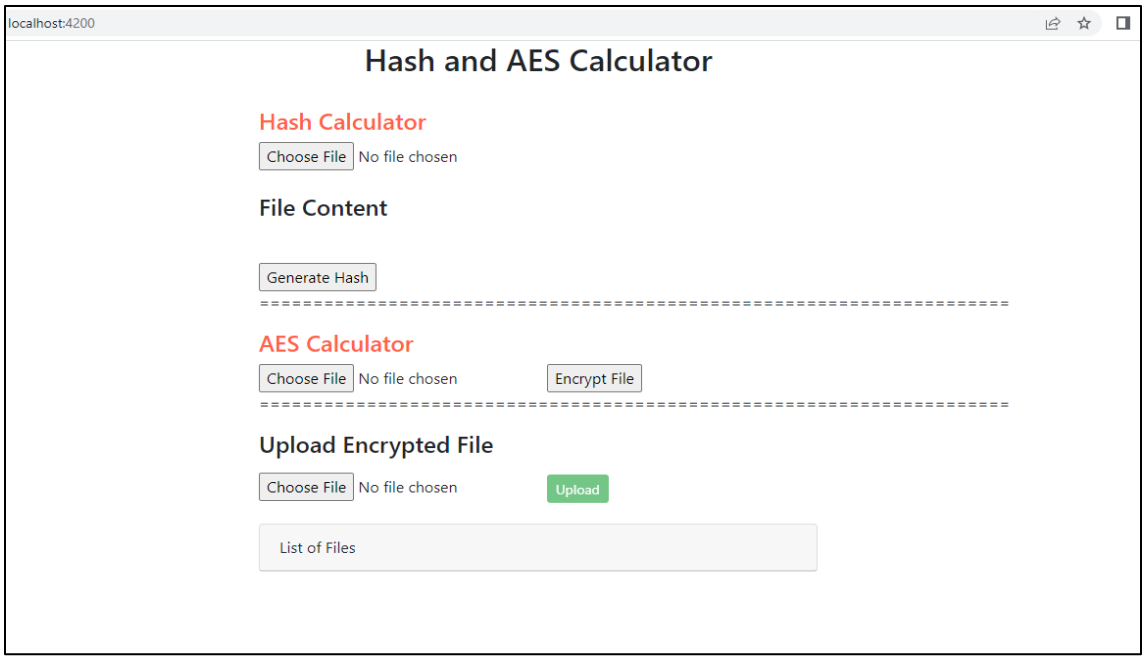


Figure 9

User Interface for Windows Application

Figure 9 shows the User Interface (UI) of windows application, developed for calculating hash and AES. Moreover, an online database is also integrated in this application, with which an encrypted file can be uploaded to the database and a list of uploaded files is also displayed. The application is running on the local machine at port 4200.

The first step after starting the application is to select a file for generating hash. When the user presses *"Choose File"* button, a quick access pop-up window gets open and user can select the file. Figure 10 illustrates this process. After selecting a file, when user presses *"Generate Hash"* button, again a quick access pop-up window gets open to select the destination path for the hash.

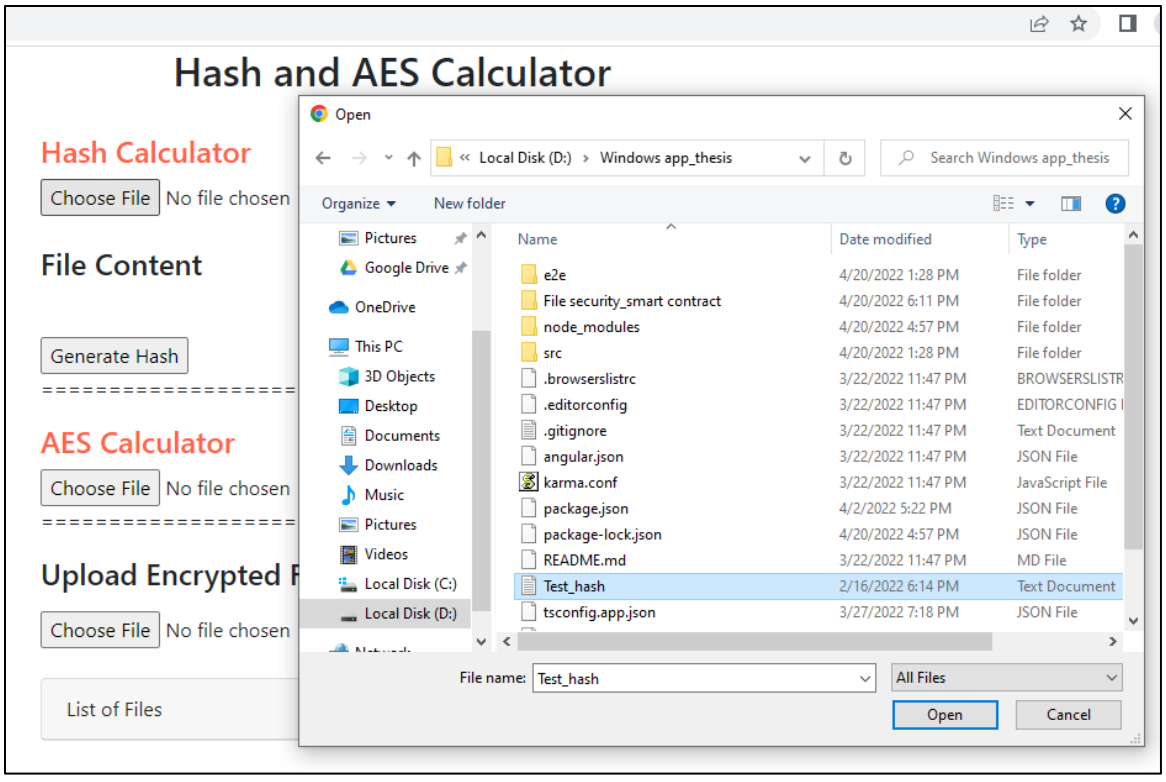


Figure 10

Selecting File for Hash

In the proposed solution, the test environment for Ethereum blockchain is configured on local system which consists of three main entities.

- Ganache (Personal Ethereum blockchain)
- Source code editor
- Integrated Development Environment (IDE)

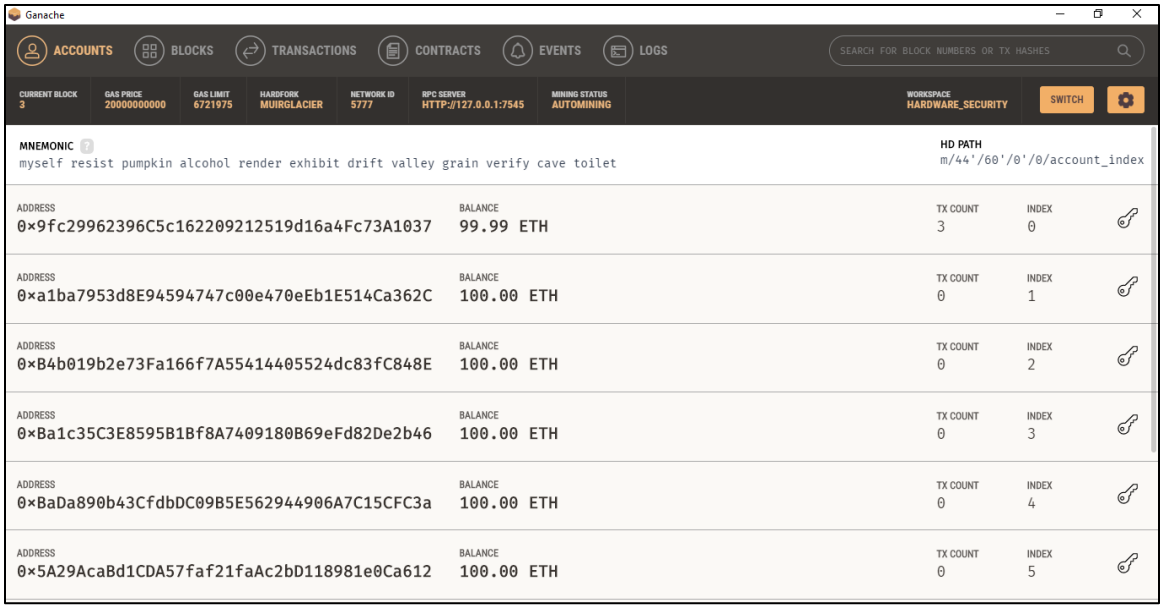


Figure 11

Ganache Admin Panel

First, Ganache, a personal blockchain is configured and an instance is started on the local machine. It is a GUI based blockchain, shown in Figure 11. Then a smart contract is written in Visual Studio Code (VS Code) using Solidity language.

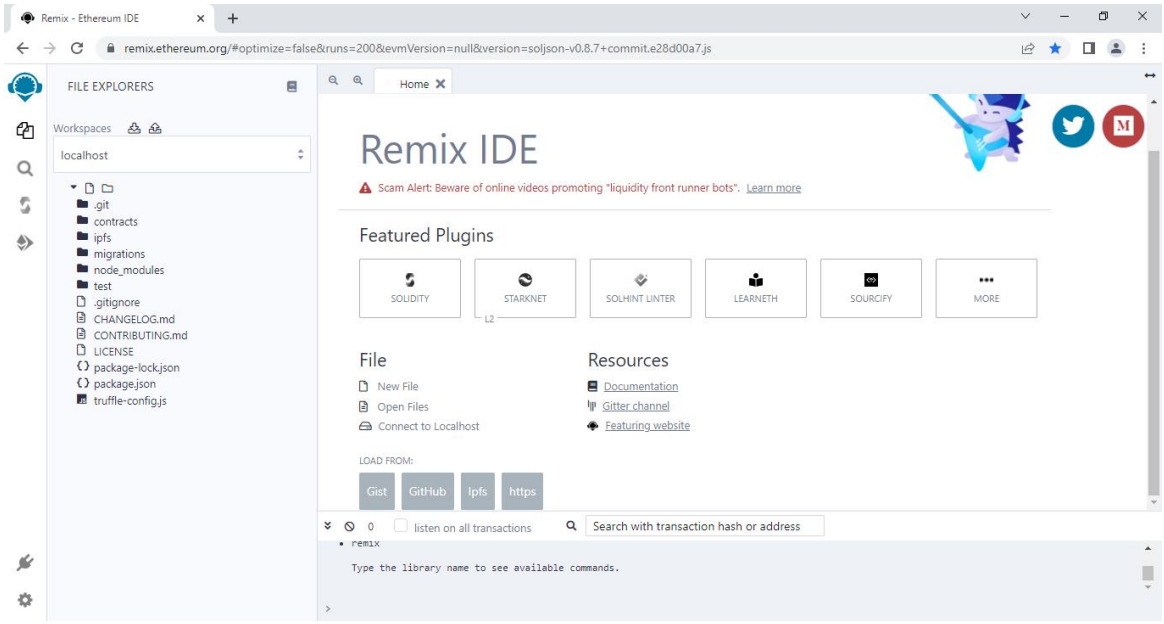


Figure 12

Remix – Ethereum IDE

In order to deploy and interact with smart contract, an open source IDE named Remix Ethereum IDE is used, illustrated in Figure 12. This web based IDE is then connected to the VS code by using Ethereum Remix extension. Once, the smart contract is compiled successfully, it is deployed on the local Ganache blockchain.

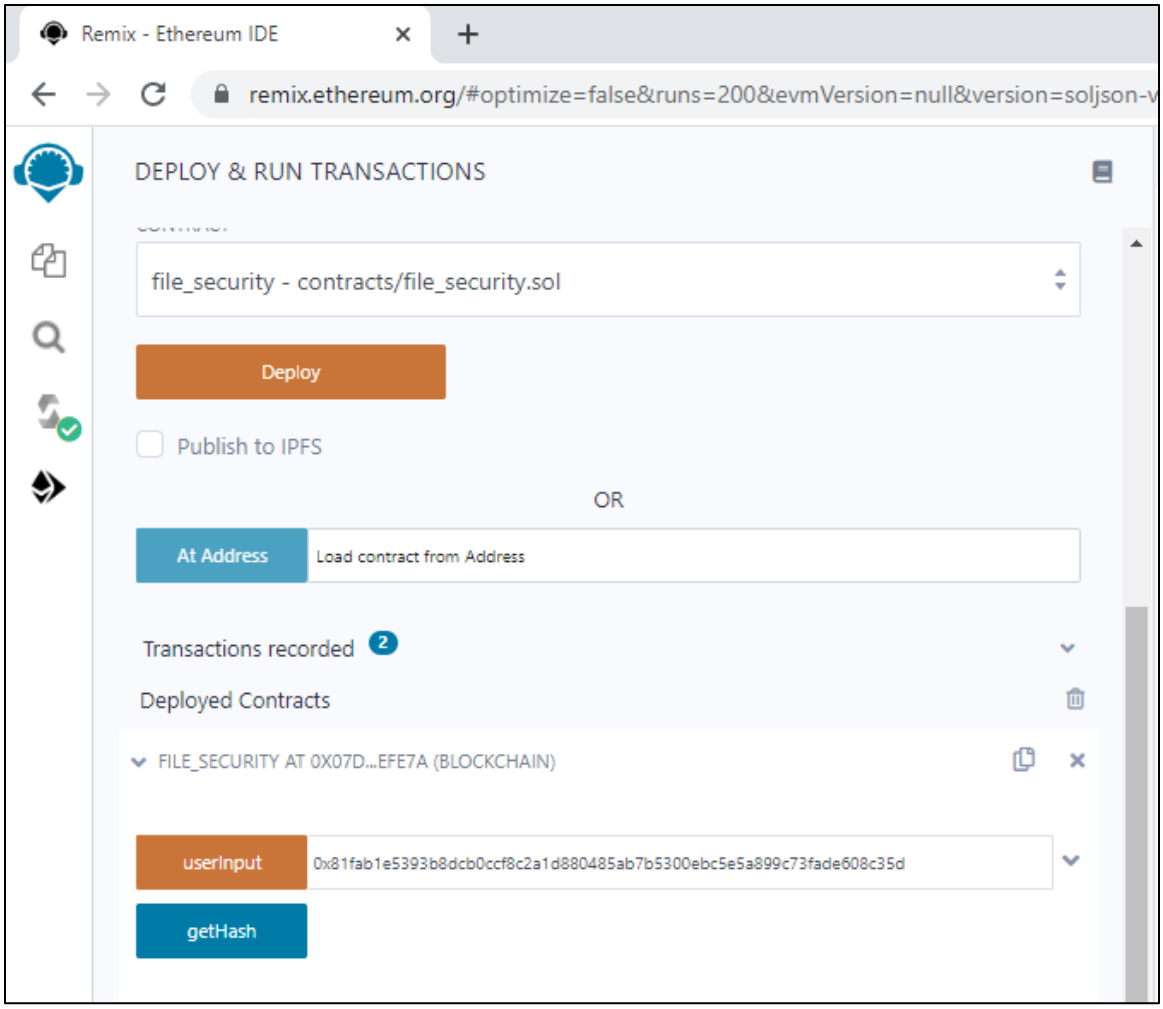


Figure 13

Remix – Smart Contract Transaction

Now, the next step is to store the hash on blockchain. When the smart contract is deployed, a user can interact with it depending on the functions defined in the contract. So, hash of the file which was generated at first step, is entered in the given space (Figure 13). As soon as the hash is entered, a transaction takes place and a new block is created in the blockchain network.

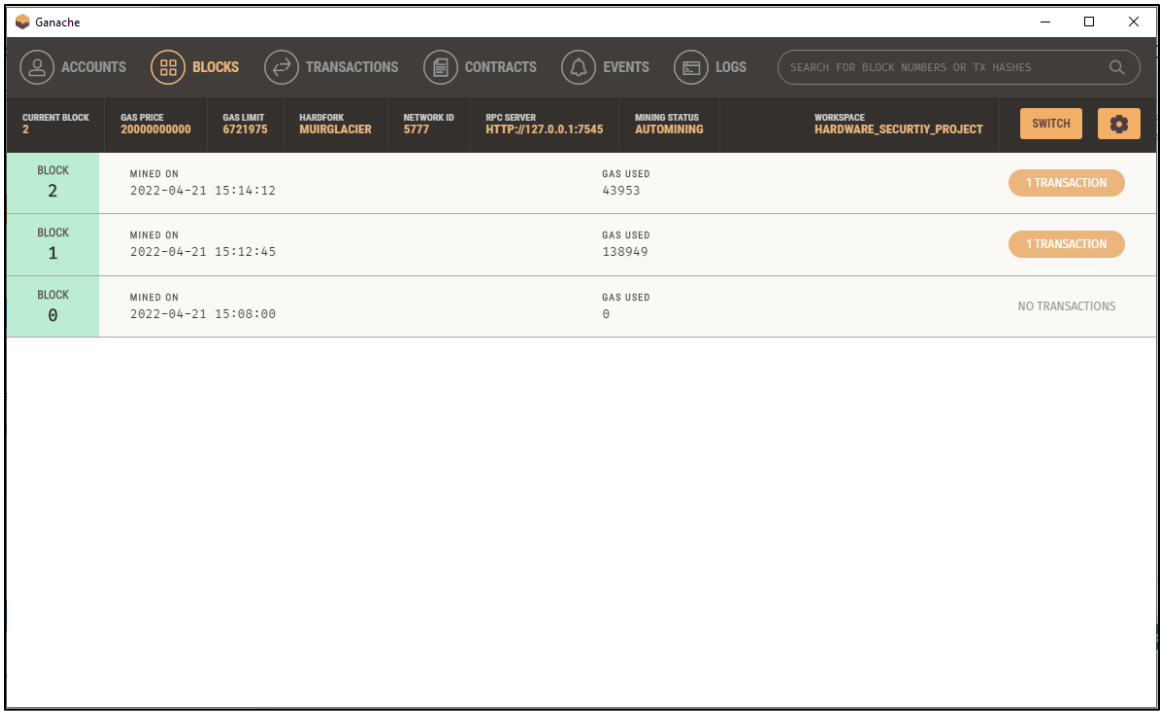


Figure 14

Ganache – Blockchain

A personal blockchain is shown in Figure 14. Currently, there are three blocks in the blockchain. The first block is Block 0, which is created when a blockchain network is formed and this block is also known as Genesis Block. It's the ancestor of all other blocks since every block references the one before it. The next block is Block 1 that is created when the smart contract is deployed. And the third block (Block 2) is created as the result of a transaction performed in the blockchain network.

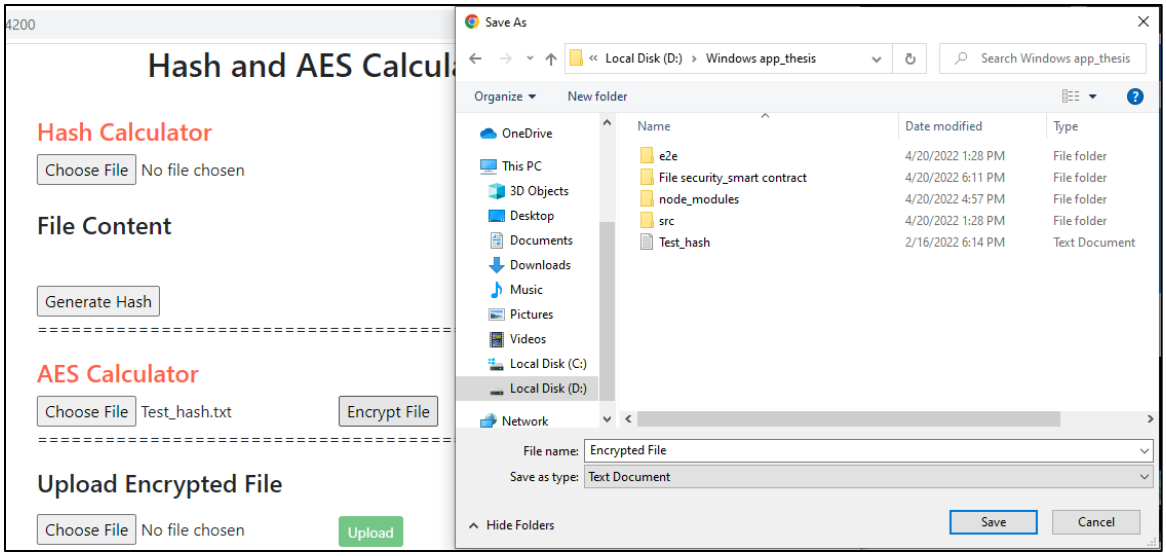


Figure 15

Selecting File for Encryption

The next step in the process is to encrypt the original file and then upload it to the database. Consider Figure 15, a user selects a file from quick access pop-up window and after encryption, user selects the destination path to save the file.

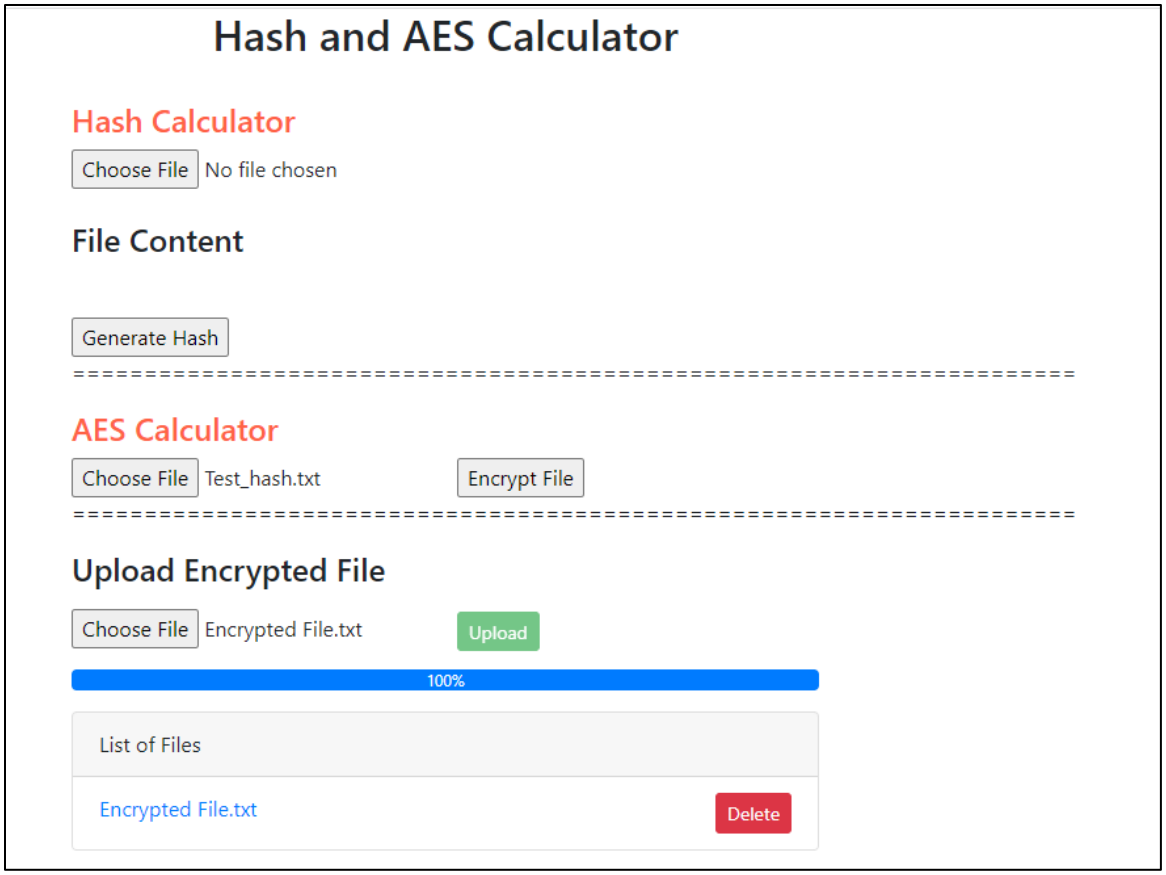


Figure 16

Upload Encrypted File

For uploading the file to the database, the proposed solution uses Firebase, a cloud based database. After uploading the file, a list of uploaded files is displayed in the application that can be seen in Figure 16.

CHAPTER V:

CONCLUSION

Current trends of globalization in design and manufacturing process of ICs and microprocessors create a source of threat and vulnerability from adversaries. Competitors can take advantage of global supply chain to add malicious circuitry at any step of the process. Detection of unwanted or malicious circuitry through conventional methods are not scalable and don't give satisfactory results. This thesis proposes a preventive solution for hardware development process files using encryption and smart contracts in blockchain. Moreover, the proposed scheme provides integrity and verifiable solution for semiconductor design and manufacturing process.

Currently, two separate systems have been developed for testing the proposed solution. One for the cryptographic algorithms and database and second for implementing blockchain. So, for the future work, a single system can be developed in which all the functionalities are integrated in single application. Furthermore, the user interaction with the smart contract can be automated to mitigate the possibility of human error. Also, integration feasibility of proposed solution with current hardware synthesizers and tools can be done.

REFERENCES

[1] "Report of the Defense Science Board Task Force on High Performance Microchip Supply," Defense Science Board, US DoD, Feb. 2005; https://dsb.cto.mil/reports/2000s/ADA435563.pdf.

[2] S. Adee, "The Hunt For The Kill Switch," in IEEE Spectrum, vol. 45, no. 5, pp. 34-39, May 2008, doi: 10.1109/MSPEC.2008.4505310.

[3] Narasimhan, Seetharam, Dongdong Du, Rajat Subhra Chakraborty, Somnath Paul, Francis G. Wolff, Christos A. Papachristou, Kaushik Roy, and Swarup Bhunia. "Hardware Trojan detection by multiple-parameter side-channel analysis." IEEE Transactions on computers 62, no. 11 (2012): 2183-2195.

[4] Tehranipoor, Mohammad, and Farinaz Koushanfar. "A survey of hardware Trojan taxonomy and detection." IEEE design & test of computers 27, no. 1 (2010): 10-25. Haber, Stuart, and W. Scott Stornetta. "How to time-stamp a digital document." In Conference on the Theory and Application of Cryptography, pp. 437-455. Springer, Berlin, Heidelberg, 1990.

[5] Pinheiro, Alexandre, Edna Dias Canedo, Rafael Timóteo De Sousa, and Robson De Oliveira Albuquerque. "Monitoring File Integrity Using Blockchain and Smart Contracts." IEEE Access 8 (2020): 198548-198579.

[6] Wen, Yiming, and Weize Yu. "Combining Thermal Maps with Inception Neural Networks for Hardware Trojan Detection." IEEE Embedded Systems Letters 13, no. 2 (2020): 45-48.

[7] Vakil, Ashkan, Ali Mirzaeian, Houman Homayoun, Naghmeh Karimi, and Avesta Sasan. "AVATAR: NN-Assisted Variation Aware Timing Analysis and Reporting for Hardware Trojan Detection." IEEE Access 9 (2021): 92881-92900.

[8] S. Faezi, R. Yasaei, A. Barua and M. A. A. Faruque, "Brain-Inspired Golden Chip

Free Hardware Trojan Detection," in IEEE Transactions on Information Forensics and Security, vol. 16, pp. 2697-2708, 2021, doi: 10.1109/TIFS.2021.3062989.

[9] L. -Y. Yeh, C. -Y. Shen, W. -C. Huang, W. -H. Hsu and H. -C. Wu, "GDPR-Aware Revocable P2P File-Sharing System Over Consortium Blockchain," in IEEE Systems Journal, doi: 10.1109/JSYST.2021.3139319.

[10] Faezi, Sina, Rozhin Yasaei, Anomadarshi Barua, and Mohammad Abdullah Al Faruque. "Brain-inspired golden chip free hardware trojan detection." IEEE Transactions on Information Forensics and Security 16 (2021): 2697-2708.

[11] Kampel, Ludwig, Paris Kitsos, and Dimitris E. Simos. "Locating Hardware Trojans Using Combinatorial Testing for Cryptographic Circuits." IEEE Access 10 (2022): 18787-18806.

[12] Hathaliya, Jigna J., Rajesh Gupta, Sudeep Tanwar, and Priyanka Sharma. "A Smart contract-based secure data sharing scheme in Healthcare 5.0." In 2021 IEEE Globecom Workshops (GC Wkshps), pp. 1-6. IEEE, 2021.

[13] Kamal, Randa, Ezz El-Din Hemdan, and Nawal El-Fishway. "Video Integrity Verification based on Blockchain." In 2021 International Conference on Electronic Engineering (ICEEM), pp. 1-5. IEEE, 2021.

[14] Ejaz, M, Unwala, I, Yang, X and Lu, J. "Securing hardware development using blockchain" In IEEE Conference on Green Technology 2022.