Shield Key: A Deep Learning Process for the Generation of Public & Private Keys using Hybrid Architecture in Blockchain

by

Tanvir Ahmed Joy
17101039
Sultan Gias Uddin Ahmed
16301151
Mir Abrar Ahmed
17101197
Sazzad UL Islam
17101457
Menuka Tasnim Nima
17101414

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
June 2021

Declaration

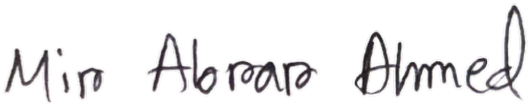
It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Tanvir Ahmed Joy
17101039

Sultan Gias Uddin Ahmed
16301151



Mir Abrar Ahmed
17101197

Sazzad Ul Islam
17101457

Menuka Tasnim Nima
17101414

Approval

The thesis/project titled "Shield Key: A Deep Learning Process for the Generation of Public & Private Keys using Hybrid Architecture in Blockchain" submitted by

1. Tanvir Ahmed Joy (17101039)
2. Sultan Gias Uddin Ahmed (16301151)
3. Mir Abrar Ahmed (17101197)
4. Sazzad Ul Islam (17101457)
5. Menuka Tasnim Nima (17101414)

Of Spring, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering.

Examining Committee:

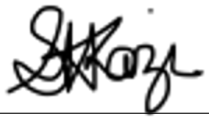
Supervisor:
(Member)

Moin Mostakim
Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr.Md.Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)



Sadia Hamid Kazi
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement

The thesis is carried out in complete compliance with research ethics norms, and the codes and practices set by BRAC University. In our thesis we use the data from primary sources. We are ensuring we use references and in text citations properly. We the five co authors take full responsibility for the thesis code violations. For solving problems we read different websites, youtube tutorials, and Questionnaire Free tools. We also took help from our university faculty members. Finally, we declare that we give credit to every person from whom we took help. We did not make any fraudulent means for completing the thesis. Our work is in compliance with the ethics standard set by BRAC university.

Abstract

The fast-growing cryptocurrency system is changing the financial world with its ensured blockchain technology. It's specific use in expounding cryptocurrencies is highly acknowledged in the modern world, although the acceptance is growing in several other domains such as the Internet of Things (IoT), Artificial Intelligence, Fintech, and many other fields. Proof of Work (POW), which contains a digital ledger of transactions, plays a vital role in blockchain security. In addition, Blockchain uses a changeable public key which ensures an extra layer of privacy by containing individual user identity. Although, malwares has been planted to encapsulate the single user identity and the keys included in it. Focusing on Blockchain's significant features, possible vulnerabilities in the system and cryptography, we will emphasize the deep learning process to generate hybrid public key in the blockchain system which will be binding with the face recognition features delivering to a compact & secured cryptosystem leading towards protection of sensitive data. After successfully generating the public & private key, we were able to implement those keys in blockchain system creating a person's wallet and verifying the signatures. Our success rate in generating keys was close to 100% which were implemented in blockchain.

Keywords: Blockchain; Deep Learning; Public Key Cryptography; Hybrid Public Key; Facial Recognition

Dedication

We would like to dedicate this research to our parents. Without their support we may not be able to do our studies. We also want to dedicate the research to our friends who helped us to do better. Our supervisor helped us throughout the year. We want to dedicate this to him also.

Acknowledgement

Firstly, All praise to the Great Allah for whom our thesis have been completed without any major interruption.
Secondly, we would like to say thanks to our beloved family members whom we will always be indebted to.
Thirdly, we thank our supervisor Mr. Moin Mostakim sir for all his help and constant guidelines.
And finally, we would like to thank all the faculty members and students for providing us with such a wonderful study environment where we could develop ourselves as well as do this research properly. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

$\approx$ Approximately

ϵ Epsilon

$\equiv$ Equivalent To

π Pi

CNN Convolutional Neural Network

DNS Domain Name System

$ECDSA$ Elliptic Curve Digital Signature Algorithm

FCN Fully Connected Network

GCD Greatest Common Divisor

GKM Group Key Management

IoT Internet of Things

$MTCNN$ Multi Task Cascade Convolutional Neural Network

NMS Non Maximum Suppression

$O-NET$ Output Network

$P-NET$ Proposal Network

PKC Public Key Infrastructure

$R-NET$ Refine Network

RNN Recurrent Neural Networks

RSA Rivest-Shamir-Adleman

GAN Generative Adversarial Network

Chapter 1

Introduction

1.1 Background Information

In the year 1991, a cryptographically secured chain of blocks was described for the first time by Stuart Haber and W Scott Stornetta.[32] However, back then people still couldn't find a suitable way to implement this peer-to-peer connection. Almost 17 years later, Satoshi Nakamoto had released a whitepaper named "Bitcoin: A Peer to Peer Electronic Cash System" in 2008 that described a purely peer-to-peer version of electronic cash known as Bitcoin which made the first debut of blockchain technology.[6] It still took a long time to understand the true potential of blockchain. Around 2014, additional usage of blockchain other than cryptocurrency was realized by the people. Furthermore, investments in blockchain started to increase and exploration of other possibilities for blockchain in different kinds of operations began to be discovered. In today's modern world, blockchain creates an extremely efficient process that dramatically reduces transaction's cost. It introduces the computer programs into the block, representing financial instruments such as bonds which are known as smart contracts. Blockchain integration flaws are also being detected across the world in recent years. Day by day, people are questioning its unique structure due to newly generated malwares leading to specific sorts of attacks. So, this was the basic idea for the authors to conduct this research for a novel approach to solve vulnerabilities in the blockchain system.
Deep learning is a part of artificial neural networks and a booming mechanism which is changing many things in this era. Deep learning which also contains convolutional neural networks works cooperatively for image processing, image detection, feature classification and much more.[12] This exciting technology has opened many possibilities which we are now implementing with blockchain technology. Initializing this theory can create a robust architecture is the background analysis outcome we have.

1.2 Problem Statement

Blockchain technology was considered robust, although security measures are not enough to keep away the attackers causing personal information to leak out. These attacks are caused by introducing malware into the user's systems. One of the types is a 51% attack.[24] Malware typically consists of code developed by cyberattackers, designed to cause extensive damage to data and systems or gain unauthorized access to a network. Ransomware, spyware, adware, cryptocurrency malware Etc.

are subsets of malware. Denoted as $\{S(R) \subset U(M)\}$, which breaks the structure of blockchain.
It is a type of malware that prevents users from accessing their system or personal files and demands a ransom payment to regain access. Since 2018, ransomware attacks worldwide have increased by 200%. Along with the increase in attacks came refined techniques and modifications. To make matters worse, ransomware is openly sold on the darknet. Keeping personal data safe is of utmost importance. As personal data breaches have become an increasing threat, it has become a number one priority to take measures against these heinous attacks on personal data. Besides, in the RSA algorithm, generated keys are retrieved from randomly generated prime numbers[30] which can be tampered with the greatest common divisor (GCD). As a result, it comes out as the biggest drawback of the RSA algorithm.[1]
Our research aims to strengthen security measures taken to safeguard personal information stored in public key { P(K) } in Blockchain. We propose a hybrid architecture consisting of Blockchain and deep learning to further strengthen personal data security. Our approach using CNN of face features is denoted as { F(C) }. The basic problem lies on the security failings of blockchain architecture which we are willing to solve. As a result, we have decided to focus on a new approach of key generation in our aim of research.

1.3 Research Motivation

Blockchain is the newest technology for cryptocurrency. Rather than that it has several unique features in terms of security, efficiency Etc. Since it is the latest technology and its immense popularity and large peculiarities, it is a strong field for study. The booming technology attracted a lot of attention in recent years and has been in our interest for a while. Blockchain's usage beyond cryptocurrency and it's distributed architecture is the most significant feature that we have considered. The public key and turning it to a hybrid architecture was the initial idea for our research. In a world of malware and ransomware, we have decided to make this technology robust and the timing was appropriate as well. Besides, we were inspired to integrate a new approach of facial recognition & analysis in the blockchain wallet so that it turns stronger against attacks. In a nutshell, the ongoing challenges to develop the robustness of blockchain wallet was the core inspiration for the authors to conduct this research. We believe that the novelty of the research lies on the integration of CNN towards the generation of public keys in the blockchain wallet. It creates the possibility of changing the blockchain security in a new phase.

1.4 Research Objectives

In our research, we are working on a deep learning process for the generation of public & private keys in blockchain wallet using hybrid architecture. We all know blockchain is a distributed ledger platform which is cryptographically protected against malware modifications. Though it has a well secured algorithm and trustworthy network, it still has some deficiencies in terms of securing personal information. Day by day, it is getting more vulnerable to malware attacks as these malwares are getting stronger which is now a big threat to the large implementation of blockchain

networks. In addition, nowadays these malwares are being sold openly in the darknet to destroy users personal information and break the backbone of blockchain networks. So, we have decided to build a hybrid blockchain architecture consisting of cryptography and facial recognition to make this vital technology more trustworthy and dependable. We will try to create a deep learning based public & private key generation mechanism and will try to give it a hybrid architectural form. In addition, we have also aimed to implement this mechanism to extract data from an existing data set to ensure our current goal. Furthermore, we will try to make this algorithm more efficient by reducing its complexity and proffering a fine-tune. Lastly, we will compare our study outcomes with the conventional accuracy measurement indicating the wellness of the proposed system. So the core objectives are recognised as novelty:

(a) Face detection with deep learning which will be used for prime number generation

(b) Public & private key generation for blockchain

(c) Wallet creation, signature creation & verification using generated keys

(d) Use of validation process for justification

Chapter 2

Related Work

2.1 Literature Review

Deep learning approaches for generating public keys are a new type of approach, although deep learning for key generation is not entirely new. There have been quite a few works conducted in this relevant field. We have studied some of the works to get a better understanding of the problems in this particular area. For a precise analysis of previous works, we have segmented the literature review part into Four Sections. The first part will describe the previous usage of blockchain in different fields alongside vulnerabilities in this technology. Moving on, the second part will analyze the outstanding technology called deep learning and it's relevance in this research work. Thirdly, the previous algorithms which are being used and lastly we will address the existing public key generation themes and security issues over here. Our selection of past works are based on the latest research works.

2.1.1 Blockchain Usability & Features

In the Blockchain review part we studied blockchain structure, previous works in terms of security and blockchain usage beyond cryptocurrency.
Paul J Taylor[28] and their team's core focus of their research is to explore the previous implementation of blockchain technology and its future mobility in terms of security in different sectors. Blockchain's significant trustworthy feature has drawn attention in the recent past, leading to its implementation beyond cryptocurrency. Some of the sectors are banking, agriculture, medical systems, and many more. The authors here analyzed the security features of blockchain and path directives for further research in this field.
Firstly, the authors focused on blockchain's usage beyond cryptocurrency and enlightened the reason behind it is the vital security feature of this booming technology. Furthermore, after discussing the previous works in this relevant field and showing the works were on the security basis. Secondly, the authors developed some research questions to find out, which significantly refers to this research's aim. The authors selected the research process by selecting the papers in this particular field, leading towards a detailed evaluation. In the evaluation process the authors extracted the data from the selective papers to better understand prior research. Finally, the authors reveal the findings of their study showing that almost half of the IOT devices' security concerns are affiliated with blockchain technology. Data Privacy and Pub-

lic key mechanism is the fourth in the trend while the second and third trends are blockchain in cloud-based security and blockchain for combating network threats.
In the end, the authors discussed encouraging the use of established blockchain networks such as bitcoin, ethereum etc. It also stated that proof of work mechanisms can be useful in IoT networks. Besides, Proof of possession and Proof of chain are some interesting findings which came out in recent research. To conclude, the security feature of blockchain is compact and further research in terms of security and successful implementation is appreciated.

In their paper, Asaph Azaria[15] and their team's main focus for the research is to create a secured medical information system which is mainly driven by blockchain technology. Blockchain's decentralized feature is the baseline over here for successful implementation in this research objective.
The first approach for the researchers was to find the reasons to conduct this particular research. They found the urgency of conducting this research because of the discontinuity and lack of coordination among the medical records that creates several complications and boundaries in the whole medical system. Besides, it is stated that medical data is crucial for the research purpose. After deeply analyzing the urgency, the authors started to provide solutions and focused the research on the basis of blockchain technology. It's public key cryptography feature has an extensive module to protect the contents. The distributed ledger technology which is the core feature protects the data from being manipulated. Every medical record is unique and the huge amount of data in this sector is also distinctive so protection and secured management for these data is highly recommended using blockchain by the researchers. The research MedRec have also the feature to mining the data so that it can be used for medical research in a protective manner.[15] Apart from that, Medrec provides the feature of identity confirmation of patients in blockchain using public key cryptography which is also binded with a DNS server for better efficiency.
The research then diverted towards a brief description of blockchain and it's unique features to strengthen the claim. Proof of work algorithm and it's peer to peer control mechanism is the state of art security which has a core protective feature for the whole system. The usage of etherum's smart contracts turns the whole technology to a complete and compact level. Finally, the researchers concluded with strong evidence in terms of completeness of the research by stating decentralization can be a power factor in the medical data management sector. Also, for conducting research the data mining feature adds more protection and uniqueness in this research. In my opinion, this is a significant research which can change many things in the coming days in terms of protection, decentralization & efficient data management.

Recently, blockchain technology has experienced few attacks. In the research of Congcong Ye[24] & their team highly focus on Blockchain and it's evolving security threats in the recent past. Wanna cry, 51% attack, eclipse attack are some of the recent attacks that make blockchain technology vulnerable. The researchers are highly committed to propose a model to combat against such attacks and turn this technology intelligent and more safe.
To begin with, in the introductory analysis, the researchers analyzed blockchain and it's functionality. They also emphasized on various types of threats in the recent

past, including Ddos attacks in this sector. The researchers enlightened the 51% attack and the way to combat this. Researchers have also used the blockchain from different aspects from miner's aspect and attackers aspect. Then later using custom made algorithms & passing them through the java interface came out with such probabilities.
Lastly, in the result part in support of their research objective they had a few drawbacks. One of them is there's no coordination between the blockchain pre-defined state numbers and attacking numbers. Their work needs further research to validate their whole work. Although, in my opinion, their work is pretty much a smart approach to make the blockchain robust, it will help to improve the security.
Nakamoto[6] proposed a particular paper that was highly appreciable in the whole world as a new method of transaction was proposed and it was highly accepted in the world because of the new security module. The researcher mainly focused on the peer to peer feature which refers to a trusted and secured framework for electronic transactions. Besides, the proof of work is the system that ensures any sort of further manipulation in the process.
The detailed analysis shows that the researcher firstly proposed the base requirements for an electronic cash system and refers to a trustworthy mechanism. This mechanism will be ensured by the proof of work feature of bitcoin. In addition, when a transaction is conducted a digital signature is to be done & a public key is binded in the end of coin to keep the identity. In this way, the cryptographic hash gets more secured. Blockchain uses the SHA-256 algorithm in the as for hash function. Secondly, The timestamp server which keeps the record of each block more specifically, each refers to a hash block and the information is open in public so that there is always a record and no further duplication being occurred. Moving on, the researcher deeply proposed the proof of work theorem which has the feature of security. If anyone wants to manipulate a hash block needs to manipulate the proof of work for each block in the chain so every time each block gets added it is much more difficult to manipulate. Lastly, the researcher analyzed the secured payment authentication and privacy features. By keeping record of the longest chain the authentication can be conducted & in the privacy section a new model is proposed which consists of flow of public information using a new key. Although the public key remains unnamed in this process. The entire research and methodology is an outstanding technology proposed in terms of technological advancement. It is the electronic cash system with more secured features that will change the financial technology industry in a rapid manner.

2.1.2 Deep Learning: Features & Usability

Deep Learning has been the technology that is a big part of the development of artificial intelligence. In the research conducted by Alfaisal Albakri & Chafic Mobel[25] the key objective of this particular research is to strengthen the protection of the blockchain wallet. Blockchain is the newest technology for cryptocurrency and it has several unique features in terms of security, efficiency and its usage is now beyond cryptocurrency. In this research, it was specifically driven to introduce a secured key which is driven by a deep convolutional neural network to extract biometric face recognition features for a specific wallet.
Firstly, the researchers deeply analyzed the urgency of this approach in terms of secu-

rity because in the recent past, blockchain technology has experienced a large number of cyberattacks. It is stated that to combat system vulnerabilities in blockchain is the main focus here. Specifically, the research was conducted to combat against transfer trojans, miner malware & manipulation of blockchain private keys. Furthermore, the researchers have proposed the dynamic technology of biometric authentication which includes two phases of enrollment and authentication phases and later analyzed the role of fuzzy vault tolerance systems. To make the process more compact, the researchers moved to few add ons and used face recognition technology in the authentication process. Finally, for turning the face recognition performance to a higher level the researchers trained a CNN to perform biometric features efficiently. Here, among three datasets, the first one was used for training the CNN and the other two refers to experimenting with the performance of face cryptosystems.
The Findings of this research are highly noticeable because the face recognition model achieved a perfect accuracy which is impressive. The overall outcome shows that the face cryptosystem can be a feasible solution in terms of protecting the wallet as well as private key in the blockchain technology. The research has extremely satisfactory evidence in support of the research approach. The features which can be improved here is to combat against deep fake which is a new emerging threat in the field of face recognition technology. Blockchain is highly vulnerable to single node attacks and it is a big concern which needs to be solved. This research can solve this issue in a smart manner.

Francesco Scicchitano[35] & his team's core aim of this research is to focus on the vulnerable sides of the booming blockchain technology which has been in the limelight recently. The authors find out the weakness of this technology in terms of security and data privacy. Moving on, the researchers proposed a deep learning model to combat these types of attacks in blockchain technology.
At first, the authors gave a brief introduction about the security threats in blockchain such as it being exposed to some ransomware attacks recently. They emphasised on making this whole mechanism more secure and later also initiated previous works in this relevant field. Although the Proof of Work algorithm has been considered as a compact mechanism for blockchain security. But in the past recent attackers have reversed transactions and took control of this secured peer to peer mechanism which was promising at first. After that the authors proposed an architecture which is driven by deep learning to erase such vulnerabilities detected in the research. Initially, the authors have worked with unlabeled data and created an encoder-decoder based model. In the process, RNN & RAE have been used to read previous events and create a new one to make it unique so that the process does not get vulnerable. In the dataset, the researchers have created seven subsets and also segmented the data in four parts as described.
Moreover, the results showed conclusive evidence that the model RAE is working to combat the attacks. Later on, if the types change, the pre-processing of RAE is also learning and shows decent results. To conclude, the researchers have stated that the research has two basic parts: the first one is the observation of blockchain logs and the second one is to train a neural network model to detect anomaly behaviors in a blockchain network. Lastly, it can be said that the research is highly remarkable to combat against such attacks in blockchain technology.
In their recent work of Shailendra Rahtode & team[27], The purpose of this study

was to introduce a system, namely BlockDeepNet, a blockchain-based secure deep learning that combines DL and blockchain to support secure collaborative DL in IoT. The authors proposed the system to overcome the challenges that occur while designing an effective DL paradigm for IoT devices relating to privacy and data handling.

The proposed system BlockDeepNet, tackles the four major problems that arise while designing an effective DL paradigm for IoT devices. They are (a) Single point of failure, (b) Privacy leak, (c) Training data insufficiency, and (d) Data poisoning attack. The authors claim that the DL task on an IoT device and blockchain can jointly solve these challenges. Therefore, the primary goal of the study was to integrate device intelligence and blockchain technique to support a collaborative, secure DL paradigm for IoT.
Firstly, the authors presented a collaborative DL paradigm that supported DL at the IoT device level and obtained enough data for DL. Secondly, it was deployed in a blockchain environment to provide a secure and reliable exchange of local and global updates. Lastly, the authors developed a prototype model of BlockDeepNet that was used in a case study of object detection to demonstrate its feasibility and compatibility in IoT. The results indicated that this system was very efficient in terms of accuracy, security analysis, time delay and computational complexity. Furthermore, it mitigated the existing challenges and obtained higher accuracy with acceptable latency.
The study shows that BlockDeepNet can be a viable solution to secure, efficient and accurate data handling in IoT devices. As deep learning is being implemented more and more, BlockDeepNet can mitigate the challenges faced in the effective implementation of DL in IoT devices. However, during the study, it became known that DL at the device level required higher computation power. So the devices with lower consumption could not benefit from BlockDeepNet. Thus, there is a need for focused research effort in this matter.

2.1.3 Algorithm Reviews

Public key cryptography algorithms are a big part over here. The primary purpose of the research initiated by Jasmin Ilyani Ahmad & her co-researchers[23] is to compare among the Public Key Cryptography (PKC) algorithms based on their usability and applicability in different sectors since they were invented. The author also predicts the most chosen PKC algorithms among them and also tries to find the pros and cons of each algorithm.
Cryptography is a data security technique by which anyone can ensure the security of an original message. Between the two types of cryptography, the public key cryptography helps to encrypt the plain text to cipher text for its security measures. Since the last four decades, there have been many PKC algorithms that have been used for many types of research. The main goal of this research is to find a significant algorithm among them and point out all the advantages and disadvantages of an algorithm.
The author informs about the research trends of the PKC algorithm, road maps of PKC algorithms in the last four decades and also the most chosen PKC algorithm till now. Firstly, the author shows the percentage of research in the PKC

algorithm which shows that where Discrete Logarithm, Integer Factorization, Lattices and Elliptic Curve algorithms show the drop percentage between the last four decades on the other hand, Lattices and Integer Factorization had shown a high ranking among other PKC algorithms. Secondly, the author shows the road map of the PKC algorithm, where it is shown that in the last ten years, most of the researchers focused on the dominant algorithm, which are Integer Factorization and Lattices. In addition, this research paper shows that day by day, the researchers are getting more interested in PKC and all PKC algorithms are showing an increment in the number of research done previously. Last but not the least the author claims that the most used algorithms in the last four decades are Integer Factorization and Lattices algorithms.
Furthermore, the author concludes that in the last ten years the PKC was used so much aggressively, but it was very passively used within the thirty years since it was invented. In the last ten years, the most focused algorithm was Lattices and Integer Factorization algorithms. Lastly, it has been expected that in the near future, a few new algorithms will be established to eliminate all the disadvantages of previous algorithms.

In Kapoor & Yadav's paper[16], the researchers have initiated a new architecture of the cryptographic process so that in the data interchange procedure, it is privacy bound. This research is conducted to improve the process and increase its durability features in the regulated aspects.
Firstly, the authors have given an analysis of public and private key cryptography. Moving on, reviewed past works on public-key encryption algorithms. Later the researchers have proposed a new hybrid architecture that contains the RSA algorithm procedure. The encryption and decryption time showed significant changes as well as there were changes in the memory as well.
Lastly, the researchers have drawn conclusions by supporting the proposed algorithm, which has significant results in support of this. RSA SHA1 and DES algorithm being used in the hybrid process. Bit discarding procedure makes the system more secure and the authors have shown that there is lesser consumption in the memory section in this process. The hybrid process is significant as a robust technique.

2.1.4 Key Management

In the research of Ompal & team[26], their purpose of this research was to show the role of Public Key Infrastructure (PKI) in Blockchain technology which has emerged in the field of blockchain wallet and it's beyond usability. Besides, the research was highly focused on proposing a group key management scheme for maintaining confidentiality over group communication.
Firstly, in the research paper, the authors analyzed the handling process of cryptographic keys for bitcoin wallets which can be beneficial for the IoT systems in many aspects. Secondly, the research deeply moves towards the key management process for bitcoin wallets which includes several approaches like local key storage,password-protected wallet, offline key storage, password-driven keys etc. where the researchers focused on drawbacks also. Finally, the researchers enlightened the field of Public key infrastructure in blockchain which refers to the multi-layered approach, Guard-

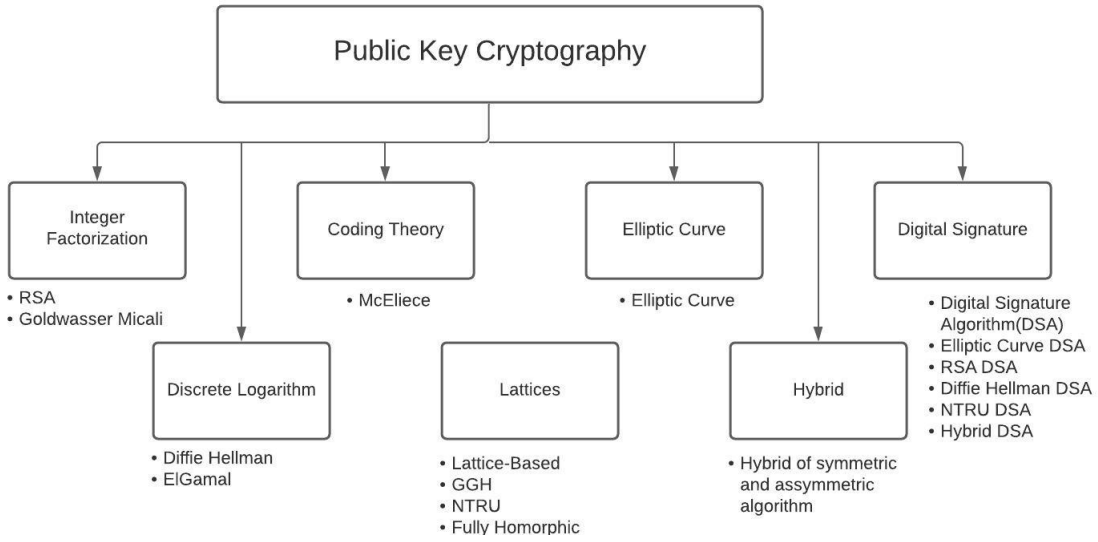


Figure 2.1: Public Key Cryptography Classification

time approach & instant karma PKI. After analyzing all the aspects of public key infrastructure, the researchers focused on specific outcomes for better security and efficiency. The major proposal of this research is Group Key Management (GKM) for the successful use of blockchain technology which includes the feature of confidentiality and efficient encryption of payload. It includes the feature of restriction on openness for non-members who are not included in the particular group of operations. The research has conclusive evidence in support of proposed mechanisms with a base theme of public key infrastructure in blockchain technology. The research should have a real-time implementation on a dataset which will justify the research and can establish the mechanism in a broadway.

Besides, we have studied private key mechanisms also. In the research by Mehmet & co researchers[31], the researchers have defined a problem regarding blockchain keys and proposed a mechanism to solve the issue. The research objective is to make the blockchain technology robust and more privacy-oriented.
Firstly, in the introductory section, the authors have discussed the role of asymmetric key in blockchain. Later on, they found problems on saving the private key information. Blockchain is the technology which has enormous responses in terms of it's privacy features. The private and public keys are vital in this technology. Private keys are the validation keys of each process, while public keys are the personal information that completes the process. Digital signatures are the process for verification in blockchain technology. Both pair keys are generated using the ECDSA algorithm over here. After a brief discussion in the blockchain mechanism, the researchers have proposed a fingerprint mechanism for encrypt and decrypt the private keys in blockchain. The authors also emphasized on the latest changes in this sector for privacy improvement. Private key recovery using the Shamir secret sharing scheme is a popular method these days.
Moreover, the researchers have concluded and stated that their first approach for recovering keys is a novel approach in this sector. In my opinion, it would be more feasible if worked in a dataset and implemented for real-time results. Finally, the research is conducted on the theme of privacy and security aspects.

Chapter 3

Proposed Methodology

3.1 Proposed architecture

Our proposed architecture is an output of critical thinking to make the blockchain architecture more robust & perfect. As a result, we are proposing to generate keys consisting of hybrid architecture. At first, we will try to generate prime numbers from the extracted CNN features which will lead to generation of keys. The hybrid architecture will consist of the possible mathematical operation of a public key cryptography algorithm. We will also cross check the key generation & evaluate the performance of them in the blockchain wallet system.

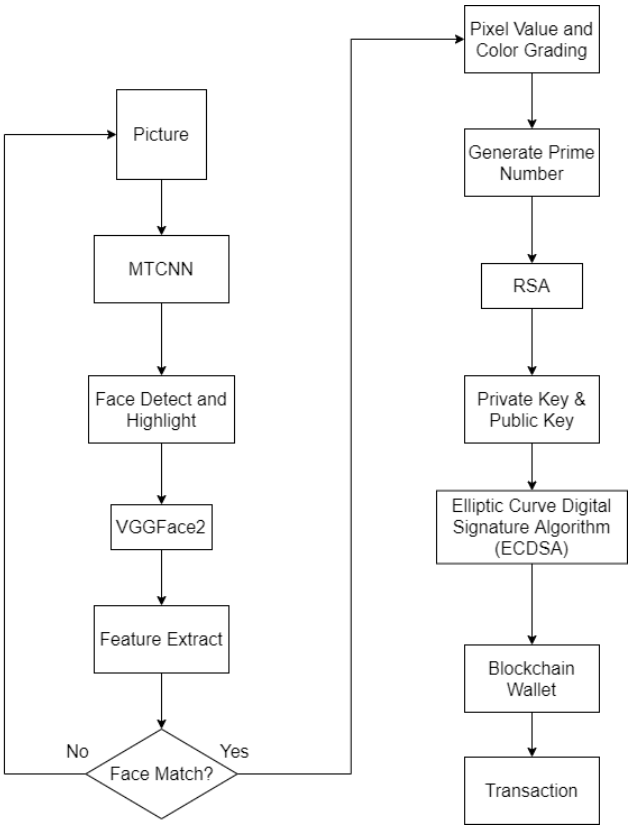


Figure 3.1: Proposed Architecture Flowchart

3.1.1 Blockchain Architecture

Blockchain technology has the core characteristics of decentralization, accountability, and security. This technique can improve operational efficiency and save costs significantly. The demand and usage of applications built on blockchain architecture will only evolve. The blockchain technique provides data security, transparency & trust because it acts like a distributed ledger. The structure of blockchain technology is represented by a list of blocks with transactions in a particular order. These lists can be stored as a flat file (txt. format) or in the form of a simple database. Two vital data structures used in blockchain include pointers and linked lists. Logically, the first block does not contain the pointer since this one is the first in a chain.[21] Simultaneously, there is potentially going to be a final block within the blockchain database that has a pointer with no value. The core components of blockchain are: Node, transaction, block, chain, miners & consensus. Node refers to a user or computer within the blockchain architecture containing an independent copy of the whole blockchain ledger. Transaction is the smallest building block of a blockchain system which holds records, information, etc. that serves as Blockchain's purpose. Block refers to a data structure used to keep a set of transactions distributed to all nodes in the network. A Chain is the sequence of blocks in a specific order. Furthermore, Miners are specific nodes that perform the block verification process before adding anything to the blockchain structure.[20] Lastly, consensus protocol is a set of rules and arrangements to carry out blockchain operations.

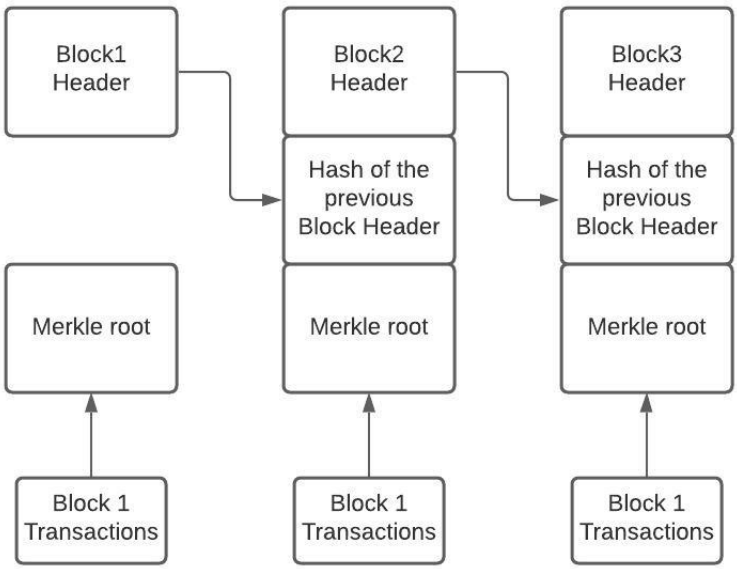


Figure 3.2: Blockchain Architecture

Step 1: Passing pictures of a user P Read P Step 2: F{C} from P where F{C} is the facial features Step 3: Generate prime numbers from facial features denoted as F1, F2 Genkey using F1, F2 stored in variable Key1, Key2 Step 4: Create blockchain for transaction & create instances of receiver, amount, message & signature Step 5: Create signature from Key1 & verify

Table 3.1: Blockchain Pseudocode

3.1.2 Deep Learning Architecture

Deep learning is the latest technological advancement in this era. Although it is a subset of machine learning, it has more specificity because artificial neural networks are trained by it. The core advantages of deep learning are: finding non-linear relationships, learning complex patterns, reducing time complexity, big data scalability etc. Some of the common deep learning models are Multi layer perceptron, Convulotional neural networks, Recurrent neural networks & much more.
Convolutional neural organizations will in general be utilized in computer vision arrangements utilizing pictures as info. They catch the spatial parts of the information; instead of each pixel being observed as an independent component, the way that pixels are close to one another or inside nearness can be mulled over. Types of CNN layer includes : Convolution layer, Pooling layer & Fully connected layer.
The CNN architecture is given below:

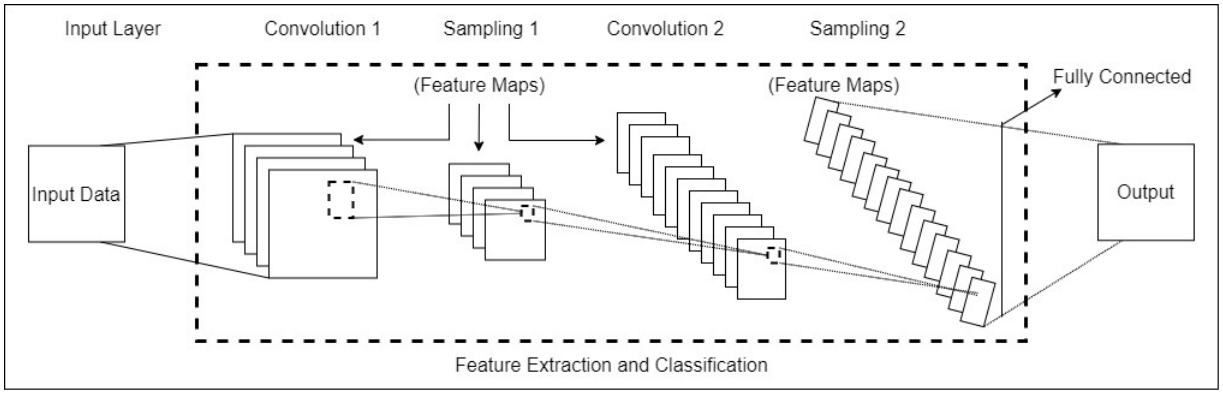


Figure 3.3: Convolutional Neural Network

Convolution Layer:
In this layer convolution operation is performed which multiplies an array of input data with a two-dimensional array of weights. The array of weights is known as a filter. The result of this multiplication is a dot product. The size of the filter is smaller than the input data so they can be multiplied multiple times at different points of the input.[19] The filter detects specific features in the input and using it at different points of the input enables it to detect features anywhere in the input. It gives a two dimensional array as output known as feature map as it is multiplied several times with the input.

Pooling Layer:
The feature maps obtained in the convolution layer are passed to this layer. It performs pooling operations on the feature maps and reduces their size by a factor of 2. This layer is crucial because the convolution layer cannot precisely record the position of features in the input. So down sampling is performed by this layer.[19] This down sampling of feature maps or pooled feature maps enables the system to locate the features in the same position of the input even if the location changes as result of cropping, resizing etc.

Fully Connected Layer:
This layer is a combination of convolutional layer & pooling layer both. To describe the process, if we go back to the convolutional layer where it extracts the highlights & further it was downsampled in the pooling layer both go for a mapping which are referred to subset of completely connected layers.[19]

Besides, CNN consists of different activation functions which are basically used for controlling gradients & learning rate of a neural network model. Some of the important functions are : Sigmoid function, ReLU. Leaky ReLU, TanH, Softmax etc.[19]

Moving on, CNN has 3 types of object detection models. Those are: Traditional image classification, classification with localization & detection. Detection in the context of object detection, different methods are used depending on whether we just want to locate the object or detect a more complex shape in the image. The two main ones are summed up in the table below: The models are bounding box detection & landmark detection. For example at the figure below, l1x, l1y... etc are reference points of a face which have been detected.

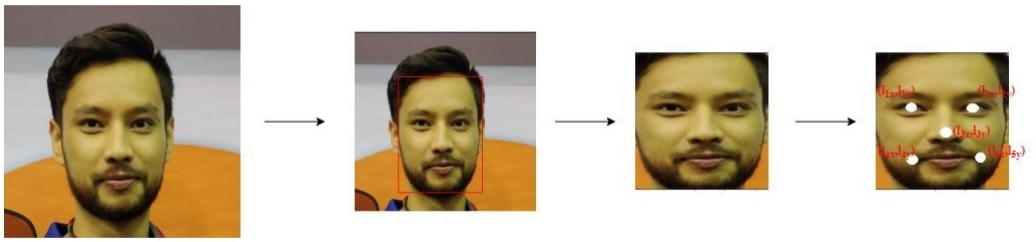


Figure 3.4: Landmark Localization

3.1.3 MTCNN & in depth analysis

Multi Task Cascade Convolutional Neural Network is an advanced architecture consisting of features towards face detection & facial landmark recognition.[3] This particular model is considered as an efficient & perfect tool for face detection in today's world. It has three layers, such as: P-net, R-net & O-net. We have selected this tool for fast forwarding detection because of its efficiency. We needed the durability of facial detection, from where the detected face will be sent for generation of prime numbers.

MTCNN is a python library which contains the module of face feature extraction techniques in an efficient way. MTCNN is available as a pip package, meaning we can easily install it using the detection process. MTCNN is divided into three parts. Firstly, P-net which refers to the proposal network also known as fully convolutional network(FCN). The difference between a CNN and a FCN is that a fully convolutional network does not use a dense layer as part of the architecture. This Proposal Network is used to obtain candidate windows and their bounding box regression vectors. Outputs from the P-Net are passed into the R-Net. Further refinement of the outputs from the P-Net are done by the R-Net and bounding boxes are reduced. Furthermore, Merging of overlapping candidates is done by applying NMS. Moving on, the third stage is similar to the R-Net, but this Output Network aims to describe the face in more detail and output the five facial landmarks' positions for eyes, nose and mouth.

The basic tasks of MTCNN are face classification, bounding box regression & facial landmark localization.

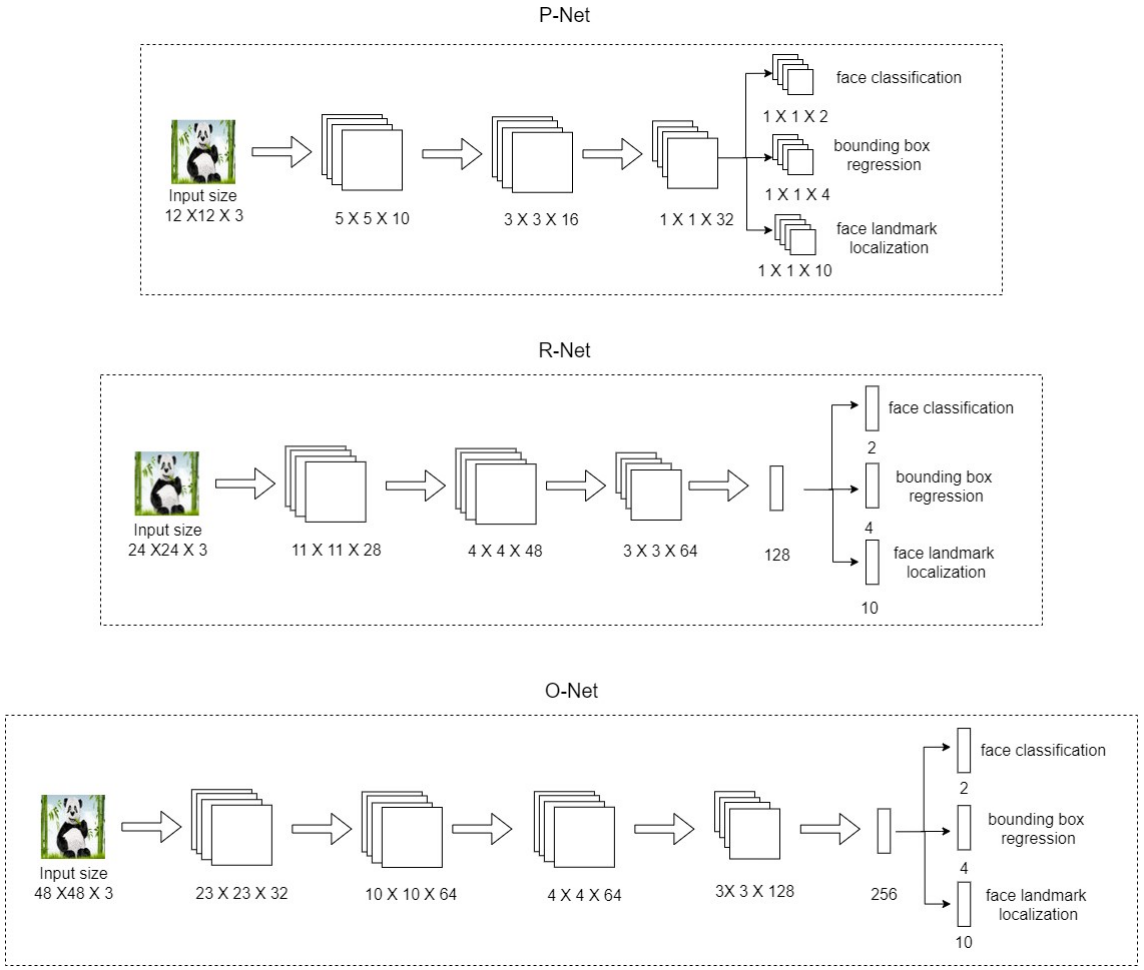


Figure 3.5: MTCNN Overview

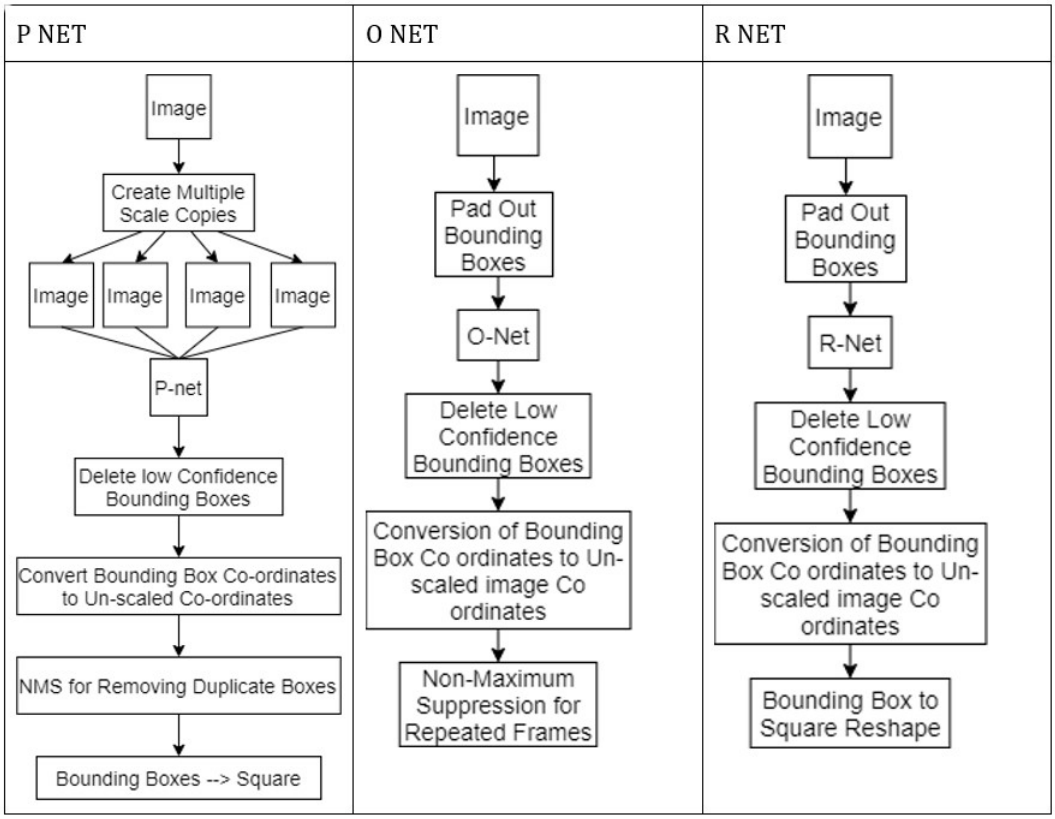


Figure 3.6: MTCNN in-Depth Process

From the above diagrams we can see that MTCNN works in three stages. In our process, we have kept our proceedings conventional & within the outlined framework. We have focused on MTCNN matching & for that we have created face embeddings using facial landmarks analysis. This helped us to compare for the face matching process only for the users who want to create a wallet Any other object or fake creatures are excluded using this model. The biggest advantage of this model is, it works based on facial vectors and converts them into cosine values which are unique for each face.

The pseudo-code of this part of our work is provided below:

```
read image using plt.imread(Path of the image)
initialize an MTCNN() object
detect faces using detect_faces() method

for each face in faces do
print "face"
end for

function highlight_faces(Path of the image, faces)=>
read image using plt.imread(Path of the image)
plot image through imshow()

for each face in faces do
draw a rectangle using the Rectangle() class
display the image and the rectangles using .show() method
end for

end function


function extract_face_from_image(path of the image, required size)=>
read image using plt.imread(Path of the image)
initialize an MTCNN() object
detect faces using detect_faces() method
initialize face_images empty array

for each face in faces do
extract the bounding box from the requested face
extract the face
resize pixels to the model size
return face_images
end for

display the first face from the extracted faces


end function

function get_embeddings(file name)=>
extract faces
convert into an array of samples
prepare the face for the model
create a vgg face model
perform prediction to that
return that
end function

function is_match(known embedding, candidate embedding, thresh)=>
calculate distance between embeddings to score
if(score <= thresh) do
print “face is matched”
else do
print “face is not matched”
end if
end function

function is_match(known embedding, candidate embedding, thresh)=>
calculate distance between embeddings to score
if(score <= thresh) do
print “face is matched”
else do
print “face is not matched”
end if
end function

define filenames
initialize embeddings using get_embeddings function
```

Table 3.2: MTCNN Process

3.2 Algorithm Analysis

3.2.1 RSA Algorithm

Developed by Ronald Rivest, Adi Shamir and Leonard Adleman in 1978 the RSA algorithm became a de facto standard.[11] RSA formed the basis for a number of encryption programs & this works for public key cryptography. In the blockchain system when two parties exchange any digital asset, the authentication phase is highly dependent on the RSA algorithm. The generated public keys are distributed & private keys are kept secret. We have used this algorithm because blockchain technology has been using this algorithm since it's creation. This algorithm is attached with the core concepts of blockchain wallet.

Step 1: Select the two large prime numbers as a and b where a is not equal b [bits are 1024 bits]. The Prime Numbers were generated from face features. Step 2: C Modulo value represented as n , n=a*b Step 3: Calculate the fi value (fi)=(a-1)*(b-1) Step 4: Select the integer of 'e' 'e' value must in small integer which is relatively prime to an (fi) and not equal to 1. Where 1<e<(fi) Step 5: Compute Greatest Common Divisor[gcd] gcd(e, (fi))=1 Step 6: Calculate d, which is an multiplicative inverse of e modulo (fi) d*e=1mod(fi) Step 7: Return Public Key pair (e, n) Step 8: Return Private Key pair (d, n)

Table 3.3: RSA Pseudocode

3.2.2 RSA Encryption & Decryption

RSA Encryption, the plaintext is represented as 'Pt' which is encrypted with a public key pair (e,N) and stored in an variable 'Ct' which is represented as a cipher text. On the other hand, with RSA Decryption, the ciphertext 'Ct' is decrypted with a private key pair (d,N) and generates a plain text as 'Pt'. The original text is decrypted by the user.[11]

Step 1: Enter the plaintext as Pt Step 2: Encrypt the plaintext Pt with a public key pair (e, n). Ct=Pt**e mod n Step 3: Return the Ciphertext Ct	Step 1: Enter the cipher text as Ct Step 2: Decrypt the ciphertext Ct with a private key pair (d, n) Pt=Ct**d mod n Step 3: Return the plaintext Pt

Table 3.4: RSA Encryption & Decryption

3.2.3 ECDSA Algorithm

We have used the ECDSA algorithm for the transaction signing process in the blockchain system.[4]

ECDSA SIGNATURE VERIFICATION: To verify an entity A_'s signature (r, s) on m, B obtains an authentic copy of _'s domain parameters D = (q, FR, a, b, G, n, h) and associated public key Q. It is recommended that B also validates D and Q.
B then does the following:

Step 1. Verify that r and s are integers in the interval [1, n-1]

Step 2. Compute SHA-1(m) and convert this bit string to an integer e

Step 3. Compute w = sˆ-1 mod n

Step 4. Compute u1 = ew mod n and u2 = rw mod n

Step 5. Compute X = u1G + u2Q

Step 6. If X = 0 then reject the signature
Otherwise
convert the x-coordinate x1 of X to an integer x1'
and compute v = x1' mod n

Step 7. Accept the signature if and only if v = r

Table 3.5: ECDSA Pseudocode

3.2.4 Comparative Analysis

Both RSA & ECDSA algorithms are used in cryptography. In the blockchain system it uses RSA algorithm & ECDSA for digital asset exchange. Both the algorithms are recognised as secured and robust but in recent years malwares have been planted to make the blockchain wallet vulnerable. Both algorithms consists of some flaws which are described below:

Flaws of RSA:

- If the prime numbers can be retrieved (e.g using gcd) the secrecy can not be maintained
- The message length must be less than the length of bits of generated keys
- It's not time efficient in the concept of generating keys
- The biggest flaw of RSA is the selection of random prime numbers. This point inspired us to generate prime numbers from facial recognition analysis.[11]

Flaws of ECDSA:

- In terms of time complexity, ECDSA takes more time than RSA in the signature process
- ECDSA is more suitable for mobile based applications because of its key length attributes
- If the cipher's get broken the ECDSA keys turn out to be vulnerable.[9]

3.2.5 Advantages & Origination of Hybrid Operation

The core advantages of both the algorithms are these are considered as secured algorithms still in the cryptographic world. But, instances of newly generated malwares are getting vulnerable as well. Besides, in the RSA algorithm it has the feature of irreplaceable digital signature which can not be reverted. After a thorough verification, we have observed that both algorithms are suitable for blockchain wallet to undergo the implementation of wallet creation.
Furthermore, for the sense of increasing security we proposed a hybrid public key generation which will be robust in the context of recent vulnerabilities exposed to the blockchain system.[10] The hybrid procedure process flow is given below:

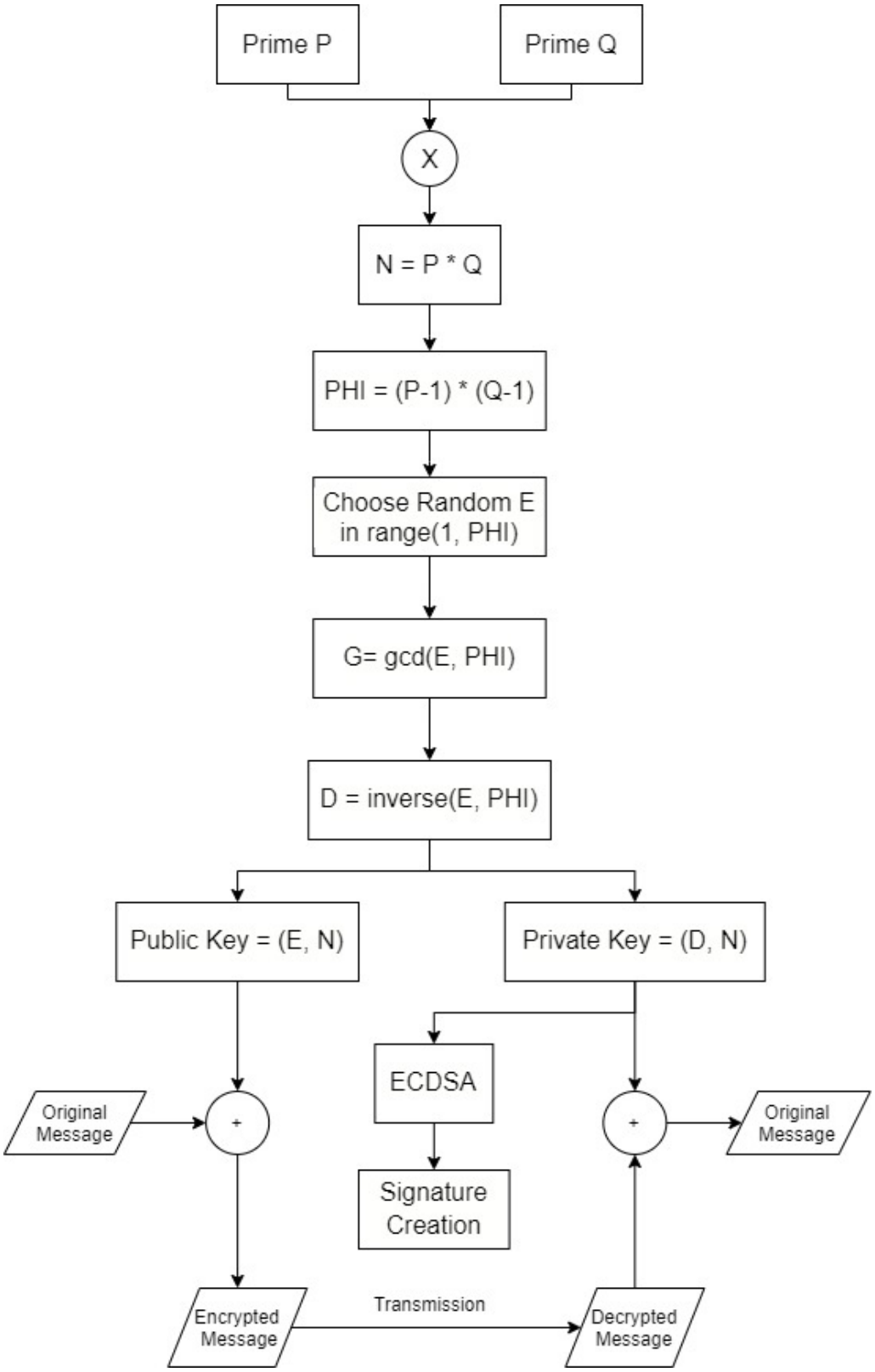


Figure 3.7: Hybrid Operation

3.3 Image to Prime Generation

3.3.1 Color Grids

In this step, we will be analyzing the core procedure of generating prime numbers from an image which we have extracted using the MTCNN model. The MTCNN model gives us the specific facial outlined face which we will be passing for the generation of prime numbers. One of the specific features of using MTCNN is that the extracted face & cross matches the facial feature using facial landmark dimension. After the matching process is completed it goes to the prime number generation step. This creates an extra layer of security in the system.

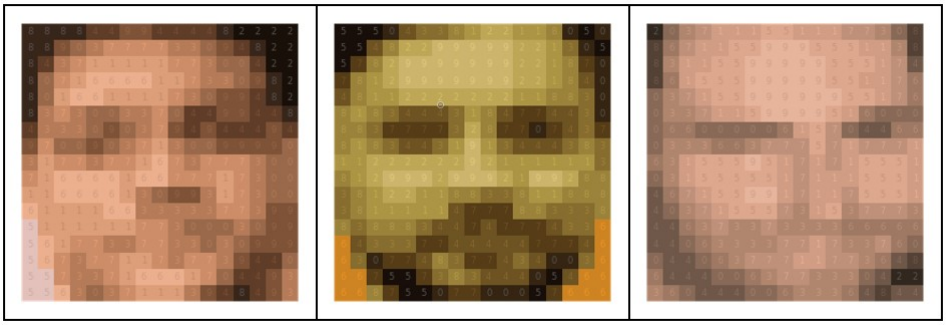


Figure 3.8: Color Grids

Following the above images, these contain an extraordinary feature which is it's pixel values. Starting from the left top point we gradually put the color value for each pixel in a consecutive manner. We represent each pixel by decimal numbers then the color space is composed of our expected pixel values.

For the color space depletion we will convert the picture color towards ten dimensional color space by grouping of colors. More specifically, we will be converting the color space based on density which will help us to reduce the color dimension. To ensure the randomness of prime generation, we have established a method to add noise to the picture. This method will enable us to select a random index of the digit and put a random number in that index which in turn will ensure random prime generation every time. This method will only work when a prime is not primarily found in the variants. The pseudo-code for this process is given below:

Step 1: Add random noise to the image
If noise count != none
then initialize noise using noise count
else
initialize noise by multiplying noise length with noise ratio and round it off
For each i in range of noise
If any digit = 0 then
Add random radiant in each digit
Return all the joined digits
 	
Step 2: Get possible prime variants from variants by adding random base value to variant using .append method
Return variant
 	
Step 3: Get contrasting colors by appending modified hsv colors to contrast colors
Return contrast colors

Table 3.6: Color Grids Pseudocode

3.3.2 Dimensions & Prime Length

For generation of 2048 bits RSA private & public keys we need a prime number which length will be 309 digits.[33] We will be using 18*17 pixel dimension for generation of prime numbers. This will be a downsized version of the original image. Moving on, we will catch the prime numbers, for each picture we have selected to produce two prime numbers. Both of these prime numbers will be directly referred to RSA key generation.

The probability of getting prime P(n) stands as:

According to Prime Number Theorem, the count of prime numbers not bigger than N is stated as $\pi(N) \approx \frac{N}{log(N)}$.
The probability of a randomly chosen n-digit number being prime is:

$$P_n = \frac{count\ of\ n\ digit\ primes}{count\ of\ n\ digit\ numbers}$$

count of n digit primes $= \pi(10^n) - \pi(10^{n-1})$

count of n digit numbers $= 9 \times 10^{n-1}$

$$P_n = \frac{\pi(10^n) - \pi(10^{n-1})}{9 \times 10^{n-1}}$$

for instance, a randomly picked 1000-digit number has 0.043%(4.3 in 1000) chance of being a prime

Search technique for the prime numbers & the whole procedure is described within the pseudo-code below:

Step 1: Load the image Step 2: Rescale the image to 18*17 dimension Step 2: Get variants from the image Step 3: Search possible prime variants and append it to variants If appended variants is not equal to explored variants If appended variants are primes then Save appended variants to found_primes Return found_primes

Table 3.7: Prime Searching Pseudocode

The process of checking prime is used by the PrimeChecker method to check variants if they are prime or composite. Later on, the variants return if they are prime or not.

3.4 Key Generation

We used the two prime numbers we got from the extracted face and used them as input for the RSA algorithm. Using the RSA algorithm we generated the public and private key.

3.4.1 Encryption & Decryption methods

We used the RSA encryption method to encrypt the message that was passed using the public key(e, n). The following formula was used to encrypt the message:

$$ciphertext = [(pow(ord(char), e, n))\ for\ char\ in\ plaintext]$$

Here (unicode value of every character in the plaintext)^e is performed and then the mod value is calculated by (unicode value of every character in the plaintext)^e % n. This is how the text is encrypted.[13]
To decrypt the text RSA decryption is performed. The following formula was used to decrypt the message:

$$plaintext = [str(pow(char, e, n))\ for\ char\ in\ ciphertext]$$

Here the private key(d, n) is used to decrypt the ciphertext. Power of each character is calculated in ciphertext by using the d part of the private key and then mod is calculated using n part of the private key. ie. char ** e % n.

3.4.2 Signatures

Signatures are created and verified using the Elliptic Curve Digital Signature Algorithm(ECDSA). We used the private key we created from the RSA algorithm to create the signature using ECDSA. This signature is used as a digital signature when a transaction occurs. The signer also has a public key with which the signature is verified.[18]

3.5 Validation

3.5.1 Hashing

At first glance, when the user wanted to create the wallet we selected the face embeddings and used one matching operation as a pre-analysis step. Moving on, when the wallet has been created & a potential user wants to make a transaction in the blockchain system we have selected the hashing operation in this instance. The hashing operation was selected to proceed with image hash authentication which creates an additional security layer in the blockchain wallet. Some of the specific hashing techniques are : Hex to hash, average hash, pHash, dHash, wHash, Color hash, Image multi hash etc.[22] In our procedure, we have selected wHash.

Color hash	Image multi hash	dHash	wHash
This hashing technique generates a color based on a value. The same value will always result in the same color	This hashing technique assigns a unique hash value to an image. Duplicate copies of the image all have the exact same value	This hashing technique is a Python library that generates a difference hash for a given image	wHash or wavelet hash is a hashing technique which works in the frequency domain as pHash but it uses DWT instead of DCT

Table 3.8: Different types of hashing

Step 1: Convert image by resizing and rescaling it to make it appropriate for WHash computation 8x8 Step 2: Remove low level frequency using haar filter Step 3: Use LL(K) as frequency LL(k) is the lowest haar frequency Step 4: Subtract median and compute hash each pixel is compared to the median and the hash is calculated Step 5: Return hash through ImageHash by Gen Hash method Image 1 vs Image 2 is compared when the transaction is initiated If Image 1 hash = Image 2 Hash then Proceed for transaction

Table 3.9: Wavelet hash pseudocode

3.5.2 Cross-Authentication Process

After the image hash been created we have compared the image hash in between to cross authenticate & adding up extra layer of security. In the cross authentication process, we have created a hash of the extracted image which was initially processed into the MTCNN model. Subsequently, when the wallet was created & at the transaction beginning process that image was taken as input again. We have created both image hashes & compared them by attributes. If the face is not being matched inside the wallet, the transaction will not go further.

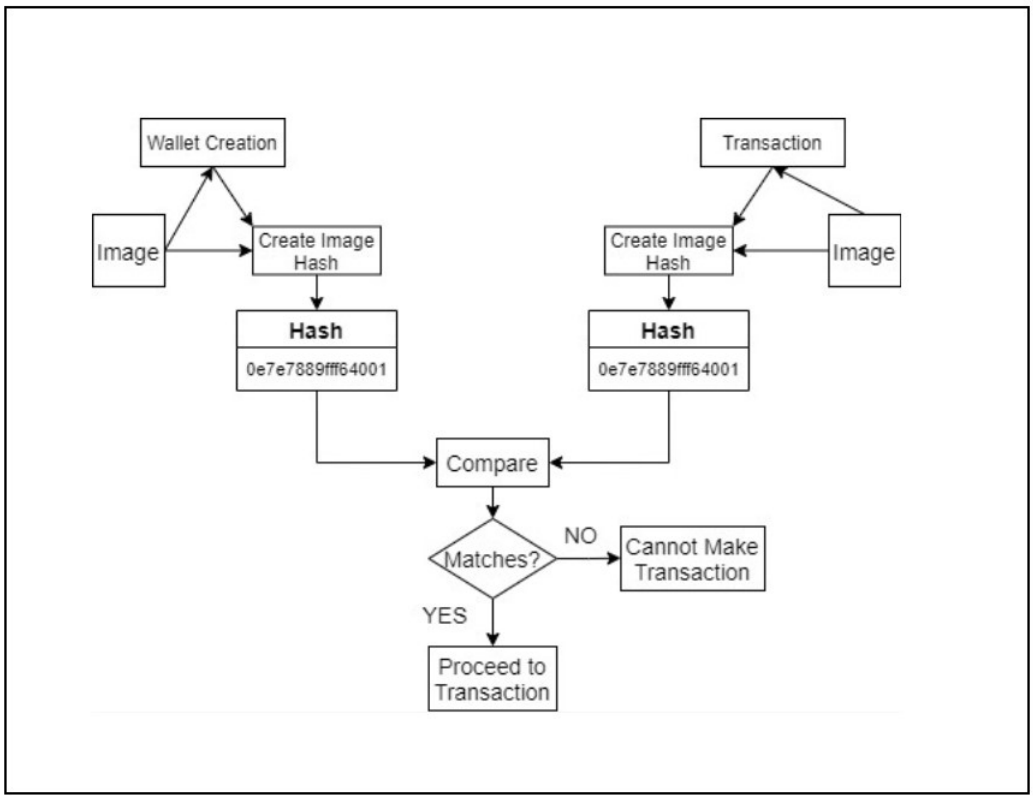


Figure 3.9: Cross Authentication Process

The initial idea of this process was to make sure to keep the identity of an individual user's wallet unique. The core theme is this individual wallet will represent the single entity which will forecast the core theme of the blockchain system.

Chapter 4

Dataset & Experiment

4.1 Dataset

We have selected the facial dataset for the face recognition process. A sample demonstration of the facial data set is given below. This data set was retrieved from the VGG face dataset. https://www.robots.ox.ac.uk/~vgg/data/vgg_face/#publi where[34] it contains 2622 individual identities for the facial recognition process. Here is the compilation:

Figure 4.1: Representation of Dataset

Besides, we have selected some samples of data using our own faces which are not from the dataset. This is because we splitted the data to show the effectiveness of our experiment results. We have taken 60% of data from the pre-defined dataset & 40% from the real life samples. Here is the demonstration of real life picture samples:

Figure 4.2: Representation of real life images

4.2 Experimentation

We have implemented the whole procedure to generate the prime numbers and consecutively generate the public & private keys for each user. Moving on, the keys are processed in a hybrid architecture binding with ECDSA algorithm which have been implemented in blockchain wallet. In the result & analysis part we will calculate the accuracy based on the facial landmark analysis. We will do the accuracy in three steps : a) MTCNN matching using facial dimensions b) Prime Length & c) Generated Keys & their Length. To achieve our goal we have given input on scattered images from the dataset and our real life images as well. Besides, we will do performance evaluation by penetration testing.

One of the extensive way of guessing password & keys which are used as penetration testing is to create combinations of prime numbers & generated keys.[29] This way we can get the estimated time & amount we need to crack the data. Theoretically, it was pre determined that the generated keys need trillions of years to crack which is quite impossible.[7] Still, we want to emphasize cryptanalysis attacks[14] which are used to reveal the cipher texts. This is the basic technique of evaluation for a cryptosystem. We will see the effect using Factorization based attacks & Attacks on RSA functions. See the table below:

Technique	How it Works	Remarks
Quadratic Sieve Algorithm[2]	$O(log^2\ N^{(1+O(1))}\ \sqrt{logN\ log\ logN}$ Where N is a large Number	Factorization Method
Combination Assumption of Prime Numbers[17]	Create Combinations using integer 0-9 & proceed for guessing	Penetration Testing
Combination Assumptions of Generated Keys	Create Combinations of keys using it's characters	Penetration Testing

Table 4.1: Factorization based attacks & Attacks on RSA functions

After initiating the process of creating combinations we will basically look forward to the estimated amount of time & usage to break down the 2048 bit RSA keys which we have generated. This will evaluate the robustness of our keys which are used in blockchain.

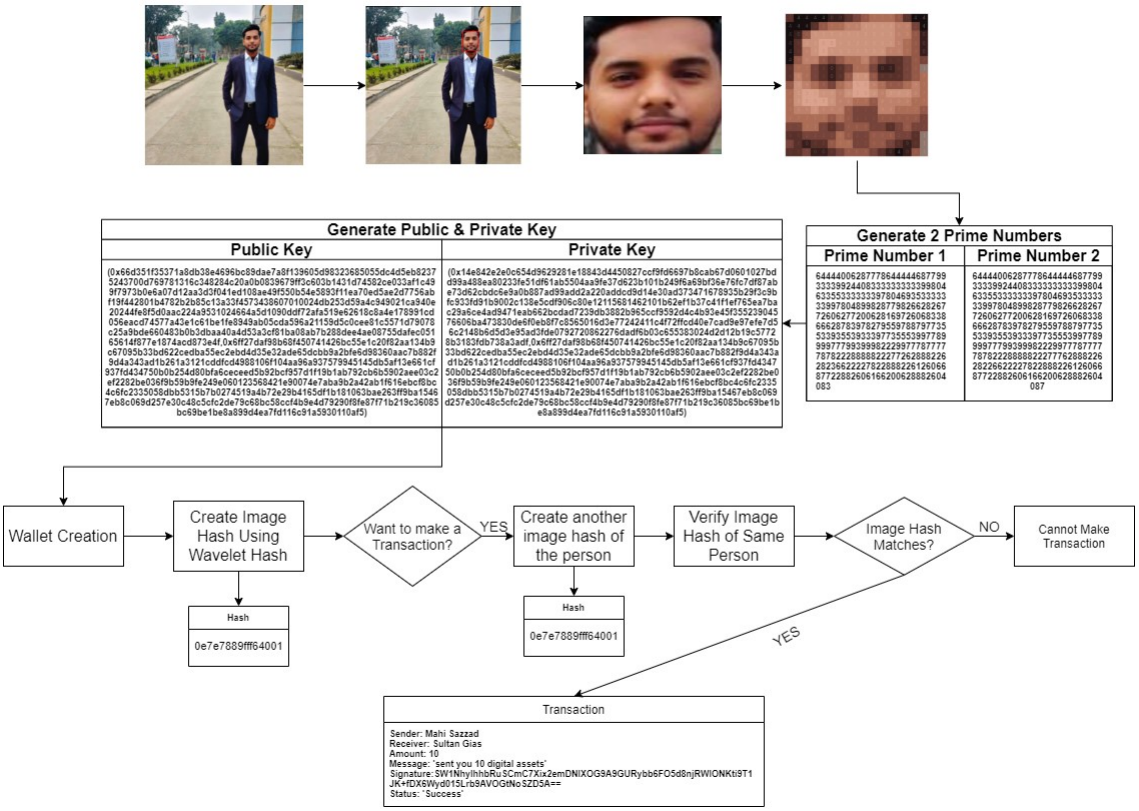


Figure 4.3: Full Experiment Architecture

Chapter 5

Experimental Result & Analysis

5.1 Results

We have come out with our results in different stages of our research that we have conducted. The results are given below:

5.1.1 MTCNN Outcomes

We have retrieved samples from our dataset & after running on the model it was working successfully. The results we have obtained using our data is provided below:

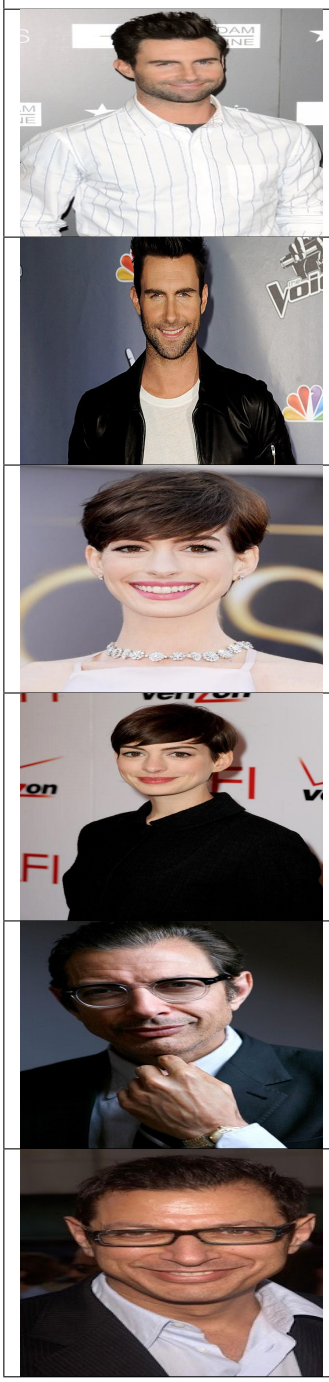
Picture	Face Detect	Extracted Face
		

Table 5.1: MTCNN Outcomes

Besides, we have implemented real life images. The samples were not from the dataset. The results are given below:

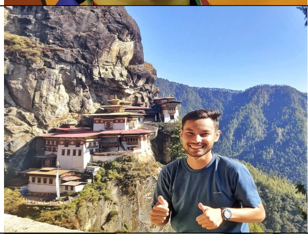
Picture	Face Detect	Extracted Face
		
		
		
		
		
		

Table 5.2: MTCNN outcomes using real face

We used VGGFace2 to compute model scores from the extracted faces as vectors. Cosine() function from SciPy was used to calculate the distance between those two vectors to determine whether the faces are a match or not. We used 0.5 as the threshold. When the distance between the vectors is less than or equal 0.5 that means that the faces are a match. On the other hand, if the distance is greater than 0.5 then the faces do not match. Once the face finds a match we move forward to generate keys using the facial outcome. The matching table is provide below:

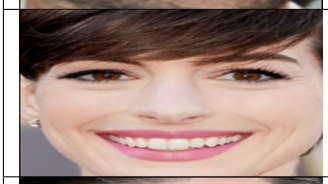
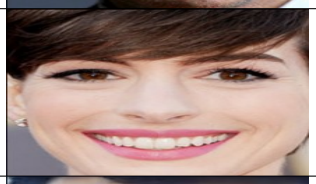
Face 1	Face 2	Distance between two vectors using cosine() function	Verdict
		0.107 <0.500	Face is a match
		0.145 <0.500	Face is a match
		0.299 <0.500	Face is a match
		0.725 >0.500	Face is NOT a match
		0.768 >0.500	Face is NOT a match
		0.195 <0.500	Face is a match
		0.327 <0.500	Face is a match
		0.203 <0.500	Face is a match
		0.519 >0.500	Face is NOT a match
		0.767 >0.500	Face is NOT a match

Table 5.3: Matching using the facial outcome

5.1.2 Corresponding Prime Numbers & Generated Keys

After the MTCNN process, we have successfully passed the extracted face to generate prime numbers using the color grading & pixel values. Moving on, using the RSA algorithm we have generated keys. The results are shown below:

User	Prime Number	Private Key(d,n)	Public Key(e,n)
	1) 203711155111773088671155999555113883115599999955517486155599999551176037155999999551760371559999995517603371159999951600006600000615704466033366377757177716315559517171555103155555171115551467155991711755514837155573153777344637337333366666440633337717337600440637711773648644400637733304223604440063336400 1) 203711155111773088671155999555113883115599999955417486155599999551176037155999999551760371559999995517603371159999951600006600000615704466033366377757177716315559517171555103155555171115551467155991711755514037155573153777344637337333366666440633337717337600440637711773648644400637733304223604440063336400	(0x525efd9affc871748e55a04a4ea36709de43586f29f57df49df6acfc772691fee9f5e13b13724582dce6b9fd62d704a2aba7152e0266be3283e4d0e733dc221d76fed76ce18d70fa077666ad025411b528f234af2b1e6f3bab9659323ba50814a123d73c25818a47e3a782a0bf852d247f3a4dbcf42079f6669e9d2147c53d83fadd1226a51e766ac065b8fdaa45603baa3542febbd543135a51bc71fd80b9ee480761dabd5d513a8a4d654561cfb220d5751323eacd35ad25b6a44632648574ae123b194fc42d35b269b1b7cd8c2ef06b756a5b95e0f66105ca90553076cc14eecd3ba1f061cd889d4a32d92182a5b, 0xb2fa2ed02609778319a6f275585e0371fecb1b03cfc8dbbfe810bb685817c260b3a1fd47ecb300662a8f84409722336bc55bd12da17b5bf3573b0eaee5550e1e18b4be54b1ab95ced484247443c38da70889d11b006722edd2d9c3c8bbccdadc785b88d3699f69982325b5e2d82c75f89d0041a38fb331c454bbfe20a2c9eba49d43a154851cfc1c62f933eb5158f109ee5179d928d7b10f29b036204c7d4e4f6ba25f6f15e694eba53825908096ee23610dbd7b52db2434afd12a4764573d0ba237bb7aeab6250282659a6078fcc4c0418a4c93a4fd5cc835b9097ed180d62a981e4e240a279ec65af38e61f258865)	(0x7516f8c8bd5c1b68f9834ff8dfbcd6eafe51ca9d348ae5b73472824a75da2ffa83237356f9060900b980459891a2f2b961560a86f4481908efcf53ea2dd0e7959249315331d095a5935ca5c3a58c29df7804db9e033cc105f96a8615970a0209b41ed1b96e9b7c20f0c0f914569da2ad63c83beec6fe0d18b355faed72dad319e3ee0caf4281b9e507c407554e502b8131ed4587195ef109f5bab715f1599f127174b3681528baaa2381cbcaa3460798dd1ec89a3c93bd2446a00cbd58158e9956ddf733bbfd4a47d530f6d294090ab594a513fa6c2a0d2e49fe6f290842606ee7ebfda5a1a813fb8981f99005df647, 0xb2fa2ed02609778319a6f275585e0371fecb1b03cfc8dbbfe810bb685817c260b3a1fd47ecb300662a8f84409722336bc55bd12da17b5bf3573b0eaee5550e1e18b4be54b1ab95ced484247443c38da70889d11b006722edd2d9c3c8bbccdadc785b88d3699f69982325b5e2d82c75f89d0041a38fb331c454bbfe20a2c9eba49d43a154851cfc1c62f933eb5158f109ee5179d928d7b10f29b036204c7d4e4f6ba25f6f15e694eba53825908096ee23610dbd7b52db2434afd12a4764573d0ba237bb7aeab6250282659a6078fcc4c0418a4c93a4fd5cc835b9097ed180d62a981e4e240a279ec65af38e61f258865)

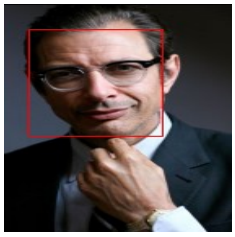
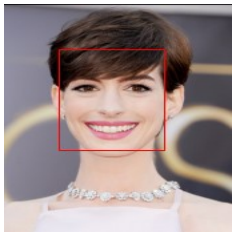
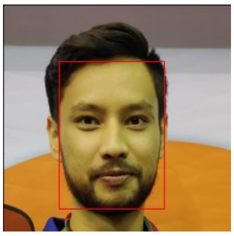
User	Prime Number	Private Key(d,n)	Public Key(e,n)
	1) 4447968521111287744096852113331577400968552133338940096822521111594496668588696677944000096569968828440400092288821150490999923252132940966995315213127900990082225511277000900620612852777700009685285227779004406852552277779000982225217777700096855521 2) 4447968521111287744096852113331577400968552133338940096822521111594496668588696677944000096569968828440400092288821450490999923252132940966995315213127900990082225511277000900620612852277700009685285227779004406852552277779000982225217777700096855521	(0x199732be9f59f55dc3e8df117fa1a5cc6d3a1fe386a71d0bab376f39f80187bb56e2af3801b4e8e02e2cbbd107ec123cbb5798068d1dc76255cc7491afe3fdb3b7a1e674933234ea1f36fd51bcf87b6149a630d45bd5078f1cd4c2c486a53fd8a938af83330b1aa02d47d1f62a75850cfe365982499cdfbda1ddd85368f21c9bbc567db46595cefe1f628eeb85dd2f2d0b0d2f59e84f2db6d4cc1cba2449a5a73d9466ae85b4293cbd5280a6e139ea9f332a668c87374d609a00c1a8c7af75be5ce9c069780d06e1b500065070573 39e817c9b14c2c4edce23552ed96180cabad08c7d69c5fc11cd7695328d1b0cb295, 0x1dfe1b7a7b442999ba25b8de685a1aa06f32e949d57ab540ce8ace52c9a29dd4bd89d4261291a15386a30234c648a74658ccdd6db247a90e45e1189a58279a94bdc157590c6102564a4b8d70eb9968059db701876e0e65260bf32cb75ea2e52beeb556171eeb7f96ddf1fb5f327b3c9fd10d2c19d11c541f897649f17e9dd14a1949e63e3a52621780dd079c47731752bc180c1fa542c1e56277f3e87deb52e6c5db89f63792d45f34552cc1aa4d9bdcdbf1df8e285d0f2fe1467f7a27f65208448c6ccc3931bda1d6e3655be5f468ec3d6c8c8b5656b5e6e23f898d0266948af4432cda30d030abbae296845d9166fd)	(0x17a8f6a2888303dcb6e96f4d8c9ef1fa401ef4d8580d2e7baddaa0e49648659b6b233b428937a3b66421d3ca7527fedf34120a2c0e4d43a198336c5ba1cbc6e36e4cf002bbaf2231092db0bd786008d630d933bd40d1ce730ab743216b24b4784038eaecc54ab89bf5b8f93ddaa0c9a7055655f73d650ce3d18f851fd3675cc7625c9fdcff37d693ad5f1f8d6bc3f70f507feaf4531aee932a52f88c9b26616123778555bf03c495ab9e5fd0084e12d78215a736aa7eb9d628a7a7db473adc25894f6bf24578217c601091c1a6576212ca32b223048ced8cadbb88488e53b2635b24b4e927d6a04dcf0e518c03d5561f9, 0x1dfe1b7a7b442999ba25b8de685a1aa06f32e949d57ab540ce8ace52c9a29dd4bd89d4261291a15386a30234c648a74658ccdd6db247a90e45e1189a58279a94bdc157590c6102564a4b8d70eb9968059db701876e0e65260bf32cb75ea2e52beeb556171eeb7f96ddf1fb5f327b3c9fd10d2c19d11c541f897649f17e9dd14a1949e63e3a52621780dd079c47731752bc180c1fa542c1e56277f3e87deb52e6c5db89f63792d45f34552cc1aa4d9bdcdbf1df8e285d0f2fe1467f7a27f65208448c6ccc3931bda1d6e3655be5f468ec3d6c8c8b5656b5e6e23f898d0266948af4432cda30d030abbae296845d9166fd)

Table 5.4: Prime Numbers & Generated Keys

User	Prime Number	Private Key(d,n)	Public Key(e,n)
	1) 6666655666666666622 2555556666666657266 6555656666655222665 5555556534322266533 3335534032634434777 4434707634353701074 5580070077701911077 7110119111991191999 1111999999919999111 0111111111111111000 0011000110001187004 7777777700108801070 1110001110887011000 0001117788801911111 1110778887019999991 0177 2) 6666655666666666622 2555556666666652266 6555656666655222665 5555556534322266533 3335534032634434777 4434707634353701074 5580070077701911077 7110119111991191399 1111999999919999111 0111111111111111000 0011000110001187004 7777777700108801070 1110001110887011000 0001117788801911111 1110778887019999991 0179	(0xb5c7f3e3427a97f38d13f6fd9108e 0d5933a6f66594be5f6ad4838ac3d064 58ee8cd875bc978b3c23648cfcbf8e3f 90223ab4f6a552ac3c9c244117025e9b 34fa9ab5e52c0c28f09a23d83bdbad95 75182e59388bb294cd0582cc98a7a0b7 c05ecd375044326d5539d30f5765ac5b 08b31183bf65b1dfaee705149be4176f 166131cea5dbf5ccf271cfa1c1e1495f 7071846517f25758baa80c2690ab7a7f 0519358703fa42af05094ed9c5512f14 24aaf609fbc473172df07255b25e44a9 dcefc6b73107877a1c2041e6817f1b3c 6667275ac5e80172cec7ffa1e348b942 b38890532aa8655ddef4ef61a718a519 5ed, 0x10d8dc46efb379ffd24e34eba67191 4894beb525e30855a57367e64eebe5ee cd2682d141088dadea2627c8cd3d458d d242681c182a2e458ce9144d072d8ef9 3f677719213f6520a56629bab9d61222 99482c6f3924491fed0b852b25a100ee 3e4b0b68ae71fbf3cbffa10a5e4fdfc1 7705a494dc3a92f9842e8361574e18db 318e8adc6c2a67a497aae7286df73514 48f425e4ad8bfebf41579d31e3679f2d dd2f16343349a4e650135d5909c8fcea 240544017282f72bf96fc483be60b458 8ac915b67821e9d5c808215f1579f49e 84407ece697b6c3d23dc27637a7d3321 fe7d37cdbbd8f9b3c59e24df5f6aa9c8 f99)	(0x9d4deba9b81d90ef5f93179e35f7fc8 374a6027f9e6d02330f8831f06af7f857f 207825b8340b034bfb9203dfefc057244e fda510c47c3154dddd21e46b9341519bbf f1daa1cba5a69990209de264f3e353715f a678f783616b59f649418b04de55d43bbf bd6ccfc6548816d1f0db7c2cbd4450bc6c 3688446bd059604607dbeaa225334b4965 ddf20801fe3bdac87560241e6065c941b3 3c3978b959200a4c97b32d70caca270b26 3124f42b13f6b28401fa0c272c6cbb85fa 27e171c8dfec9e806ddb0b4990fbc6860f 92960aab985abbd4476ddde79945cfd8e4 620e1481124ec812c143547abc38eeb9bc e154149, 0x10d8dc46efb379ffd24e34eba6719148 94beb525e30855a57367e64eebe5eecd26 82d141088dadea2627c8cd3d458dd24268 1c182a2e458ce9144d072d8ef93f677719 213f6520a56629bab9d6122299482c6f39 24491fed0b852b25a100ee3e4b0b68ae71 fbf3cbffa10a5e4fdfc17705a494dc3a92 f9842e8361574e18db318e8adc6c2a67a4 97aae7286df7351448f425e4ad8bfebf41 579d31e3679f2ddd2f16343349a4e65013 5d5909c8fcea240544017282f72bf96fc4 83be60b4588ac915b67821e9d5c808215f 1579f49e84407ece697b6c3d23dc27637a 7d3321fe7d37cdbbd8f9b3c59e24df5f6a a9c8f99)

User	Prime Number	Private Key(d,n)	Public Key(e,n)
	1) 64444006287778644444 68779933339924408333 33333339980463355333 33339780469353333333 99780489982877982662 82677260627720062816 97260683386662878397 82795597887977355339 35539333977355539977 89999777993999822299 77787777787822288888 22277762888226282266 22227822888226126066 87722882606166200628 882604087 2) 64444006287778644444 68779933339924408333 33333339980463355333 33339780469353333333 99780489982877982662 12677260627720062816 97260683386662878397 82795597887977355339 35539333977355539977 89999777993999822299 77787777787822288888 22277726288826282266 22227822888226126066 87722882606166200628 882604087	(0x1b148b5b31dd269ce1910e67162 3fc1d17eab8de495c2b6dad7136e4a f8f2cda15ed92584a2b83ac870897b de0be925b1b405a5e648be5eb4c775 14e245034fe9cd3f4251297ae2f4ab 6698d14c2659e2dc8007bd5b06579d 64241c3d238b68562e44a502e476cd 4be4a9709974c20bd6c4d38be095ff 6097d788b392a6b579dfd4adf5c687 422a0051571ce329079f2d3e77f346 7f44b58a474aac393de64dcf0333cb a5f6926cdfbd90b3ea69f2d2b84ddf c451fed0634343b165953147ad3a1d 2b998f2c13c9d7bec0b14a94b2f98d 9552a02ad05abf7a5bdfdd2863c0ae bfe2761bc89d1dd34468a47e6fa6a8 8a7, 0x6ff27daf98b68f450741426bc55e 1c20f82aa134b9c67095b33bd622ce dba55ec2ebd4d35e32ade65dcbb9a2 bfe6d98360aac7b882f9d4a343ad1b 261a3121ccf6e443611b8e5e35a57e ec43ce965b962ce4fd1d626696e72c 6aaebd4dc9f263b267da8079ddd7d6 a772bc97fde8caa70b2a0111cd9797 1d289e141e328bb17ea1a065d82e76 7c69249db2e0fc7005c81e9a359ba4 5457ed58836daf2fb6f16107bf3575 1be6fe34ecd426c988046acaa40218 5b84338504f7419ee8cbe75a6bcc49 af313bafc6f7b31def068b2119b95e 3a9e796635af7f01822310d873003a 17cc7ba1f17b503fe1f11a5930110a e9)	(0x1a4860e9903d7f474afa3c7a644cca 6e8dbfdcc4755e333b68a513b162a750d 56ee7872fc8a0f0b90334cc92b6687282 1cb0d9bb156272f8d6c94632243e3f06e 4f7b284e4ffcf4fe9a14549e880795840 5ee559b7228042f322cef96711a1708ea ab07a74c2e4fc89d0581440c02413b374 37345aa0fcd9d6763691ef908f251e02f ce3d29d2b524666b34506306452df586f 90f0e0557aa2e2e3d27c3745651f3802f b2b5a0ca4b76b4d63b2348e21c2b4c1a8 a7b383d781189d02435b80f8dbffcb464 1067024c72334c546f9adbfbbec9aa51b caa8b1770a105a28aa5e5f006162a5aec 526228fc59784cedb4a97, 0x6ff27daf98b68f450741426bc55e1c2 0f82aa134b9c67095b33bd622cedba55e c2ebd4d35e32ade65dcbb9a2bfe6d9836 0aac7b882f9d4a343ad1b261a3121ccf6 e443611b8e5e35a57eec43ce965b962ce 4fd1d626696e72c6aaebd4dc9f263b267 da8079ddd7d6a772bc97fde8caa70b2a0 111cd97971d289e141e328bb17ea1a065 d82e767c69249db2e0fc7005c81e9a359 ba45457ed58836daf2fb6f16107bf3575 1be6fe34ecd426c988046acaa402185b8 4338504f7419ee8cbe75a6bcc49af313b afc6f7b31def068b2119b95e3a9e79663 5af7f01822310d873003a17cc7ba1f17b 503fe1f11a5930110ae9)

User	Prime Number	Private Key(d,n)	Public Key(e,n)
	1) 88884499444482222889 03777733094822843711 11177300422807166661 17330982801661111730 99486837300337044994 48433309037949444949 70090371300999903177 37716737773007166661 66777173001166661309 93173006111166333337 73995111111773000309 95617733773303099956 73771173773944055733 71666170449355630371 113948491 2) 88884499444482222889 03777733094822843711 11177300422807166661 17330982801661111730 99482837300337044994 48433309037949444979 70090371300999903177 37716737773007166661 66777173001166661309 93173006111166333337 73995111111773000309 95617733773303099956 73771173773944055733 71666170449355630371 113948491	(0x924b03439341afadbd621610f1ab3 61f2bd7dbd3120a956a9368efc6dd8a1 5c0bd9ad0cf04aed24eb52cbf52dad52 a808c71990776b3b7f14b483dcd56833 f3680d194f25ec0a0413b2af484662f3 61f379253925b2d21e724ac90bf29535 d8bb581c8b19a706cfb48e770e24538c e801f649814e52b9c3997041bcc880ae 6cdeb530dd8b160ee547f24777e2ebe1 7b44f568fb7951e7609b34699a4f9089 00eb19123a51b22d080a4d9cea72612f 24c6b4a8f56a48d7063032e2989c39a3 7152c1dd7ce10d490e49dc80b4934de3 087cd00775963df0fa07cb3d06a1da94 0e85220a6f49513016845996e4d41aab 457, 0xd4f60cef6dfee15fef5c3ddb2ff875 e9f274dc5bd42db787312fbc23736f5a 2332bda8f37b8b28f57435c7aabec7c2 a8727974f28bb45bac4f7404b4e1680a 984a9cf877fb945d455bd9130c647fee 2092c2d8269a5a48b21867c904bb9ad0 f90d68e27430a525d1274d688be272ff 80560e09582a881f7ad58d618afd7287 845adf5342c51e39e71aec81ad4af7d6 b781c4675099217edc2ae331e0396ebd 7710372275ef03ab0c7e92c4ef4fdd51 603a20df563720495376f3bfd94003d1 739ae1e2856869999fa85d61e79c5246 53992408dac4168cfcdc2a89113432d0 995daec3d3fa67ab9e80dfd52cfe0c38 e9)	(0x83b91ce0f8227deb6535ea05d4fdfd2 c43d0d3db93a3952b13da453bffc3c7021 6c1eb19cc61a8cd3f6e6885b53da122fd6 cf9140fda5c578083f9a9541289f4a2910 111e8056045893d6fcd229756a15826311 22a26adbbd8b46dacdc2183bcf722fcd68 7a66a8c953680147bdd8336b00ce7b6d8c 44dcf0b53f14ff8167d66d6e23bf2fbfd1 fe9ba2d1ffea132d8da2743fdb8387eccb 5cb28d9269e9a0cfbbe8bf6c83604c4592 8ab057b82b078b6c69650c47301afc6d51 448a01ac5b7c8982be013153e370efe364 8aa0f677539fc4437a861b31b7f66312d3 5d7fce68e322e7ecb64ddec7c470126c7d f62eaff, 0xd4f60cef6dfee15fef5c3ddb2ff875e9 f274dc5bd42db787312fbc23736f5a2332 bda8f37b8b28f57435c7aabec7c2a87279 74f28bb45bac4f7404b4e1680a984a9cf8 77fb945d455bd9130c647fee2092c2d826 9a5a48b21867c904bb9ad0f90d68e27430 a525d1274d688be272ff80560e09582a88 1f7ad58d618afd7287845adf5342c51e39 e71aec81ad4af7d6b781c4675099217edc 2ae331e0396ebd7710372275ef03ab0c7e 92c4ef4fdd51603a20df563720495376f3 bfd94003d1739ae1e2856869999fa85d61 e79c524653992408dac4168cfcdc2a8911 3432d0995daec3d3fa67ab9e80dfd52cfe 0c38e9)

User	Prime Number	Private Key(d,n)	Public Key(e,n)
	1) 55504338122118050554 82299999221305578299 99999221870781299999 99221840481122222221 18830818344431847744 30883777732877074378 18344319134443831122 22229211111138129992 99921992883812211883 38118883811111477788 73388388834474777433 66433377444447774663 07748377434003666755 73834440546666875507 700057669 2) 555033381221180505548 229999922130557829999 999221870781299999992 218404811222222211883 081834443184774430883 777732877074378183443 191344438311222222921 111113812999299921992 883812211883381188838 111114777883338838883 447477743366433377444 447774663077483774340 036667557383444054666 6875507700057667	(0x28f760a80ac2d9256d927c3046afd 794e1832926d5bf23a83a73f9f61f02c fc73a9f6e70dc30bd9191c08f660c65d 8e35dc20ed9ce2475a794e0d71b73d0d 9ce8423e610851be9693d7483ef8e07c 6070148d5c0d2dcb8ec2e32e04a7122d cc62fa6bddd76de2c5339822d9a157ff 21138627dcdbe4515a47eaee366d4d46 e93a5f5f4fa85095914283ec74874d27 61dd6abe0ab298d3e6cc239d9b5ce048 746a8c31ee2a9d12b2d015b86dafa54c 550ec0c9da5358b3ddfff519aa24a797 fa1007862d8d8f0eed1cc0ac32cab34a 705854fbcfe7180d5fa9eebdadbefb95 6b18adddb788304506e68ca3010345e3 6fd, 0x530af6faacf9b7c3e916f8a4802158 3c3fab46eb0323c6a04c513b0c70187b 1981b00deead473b94023daecfef51e3 f67e19f5fadbcffb5bdc1834373588ca 20e2f0babfbdf92d3b097212d9a69767 3b5e4ae9f55e9bd132c064cf1a0ed9e9 238d400088e3b45f87be24fbd0e64573 41b201b44b376dcc2bae29fd45588503 6cd5636f7d197632eb42259ddc3f54d9 e6c886046300bcfefd56d3cc6cd985bd 7e72c2f175d6e2ec109dfbb4d86236a6 d57e8575ee73b31f8a8c9218a0f1d352 a3e58b5fdf37ad6461c85595cecdf272 070cbc94d27d2a69fd5df476b12580a8 f55b8674cccec9260c5d9e5917fd2228 fb)	0x18ce7011432ba291314d07a73b42 45952b55f99a6ad0a29d6588f0603e bd9d5eee2f173902bf234c6fb75e6f c741e2638c9d624aa96959a8715dca f77b4e5165eec4d3a637de801306f0 7d25281c7f50076c2ff67ae7d08d85 eec7ec83eed6b4cb87be260f68e455 a06a5d53cc1d902297063e813e1d0d e124457bfe6541ab5246529b1a0476 9664fbce584162cfd9137fddb66832 e06079a7007dfa3e634d22611a1eae 1c6f668db944ac3f818355c5bbdaaa 82bc7513a54e462869ed94a66bc838 04fe00f945dcef7ef9da7529ccc842 2d60ac8f4cacebec329af6211c05bb 94b179b1b1f21859ce525b1c799918 ed, 0x530af6faacf9b7c3e916f8a48021 583c3fab46eb0323c6a04c513b0c70 187b1981b00deead473b94023daecf ef51e3f67e19f5fadbcffb5bdc1834 373588ca20e2f0babfbdf92d3b0972 12d9a697673b5e4ae9f55e9bd132c0 64cf1a0ed9e9238d400088e3b45f87 be24fbd0e6457341b201b44b376dcc 2bae29fd455885036cd5636f7d1976 32eb42259ddc3f54d9e6c886046300 bcfefd56d3cc6cd985bd7e72c2f175 d6e2ec109dfbb4d86236a6d57e8575 ee73b31f8a8c9218a0f1d352a3e58b 5fdf37ad6461c85595cecdf272070c bc94d27d2a69fd5df476b12580a8f5 5b8674cccec9260c5d9e5917fd2228 fb)

5.1.3 Message, Cipher Text & Signature

Our ciphertext is very large because the generated prime numbers from the images are very large. So to make it convenient we performed the sha256 hashing algorithm from hashlib to create a hash of our ciphertext and showed it in the following table. But we decrypted the actual ciphertext we got in the output of our system. The table shows the output-

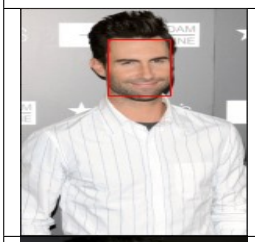
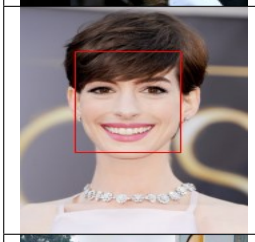
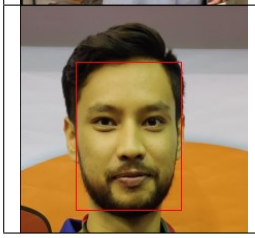
User	Message	Ciphertext (Encrypted using private key)	Messagetext (Decrypted using public key)	Signature
	sent you 5 digital assets	1b0e3bde5a83ec710 6a520f5c8c7beecbcb 9a0ee4a69c401deebf 96c859e13b1	sent you 5 digital assets	HUlQ+MGuYg1Bc 0HPJEVHd9ke05rB 7oJRKvH+U5Ks55 qk6VSScSEHJglMs Lk3O1ZjM6nXs6n QNW+pFjEMsHYdJg==
	sent you 6 digital assets	f00a9004549b0cf23 f30e7c746f65991b4 700cddfd411c00173 3686a9bb857ae	sent you 6 digital assets	udazYeK5U+MYGu mWk/ISDF7vW5LH RzbMeNG/SgXtVeu N4BiGrXC6FQRQ1 4mxpQ0RZPl+dmag JVjmzcnDJQFGPQ==
	sent you 7 files	cf7aaab5c2f5b6239a de6912a09761512c8 a3dca4dea240a1643 3e0af56cbce7	sent you 7 files	KMI8K11/tQu9+w G0OApGXxAC4Q JJq9NEJ9jUqVu6m Spo03sRPWwJgJrY zs0xeU85o0FhehW HWPBfqrxjEXNnkg==
	sent you 500 Tk	27ccf3f211a38918aa d4c7665f40541f636f 1c794b9879aa0490f7 4b0ea2aa92	sent you 500 Tk	6I9UN7pG/z1jIzhpI CFRY2W1Lu7AYus tq9Iy9xuiTMWAcU qoIt447MxP1RTKk 8iPbJZF0wTfqh4i2S wZf7Gq+Q==
	sent you 200 Tk	deb473e7d13b2d60b 457753c489f4151e0 2309bee2bd4880bbf ebba4c35b6479	sent you 200 Tk	Z9P1wSKG9xqVhx3 NqD7lfONEim6afOR h77jF3WXiWYhN59 57u/H5SknyAb7kIJX T4xVbQjjStUN8Ea4Z Njpytw==
	sent you 10 digital assets	ed30a005663e8eab41 4dd2c9fa97aa71b033 26ececf34dca9befe0f d4c55170a	sent you 10 digital assets	oWZl1vKQeo9IVTw hO3luxFnrcYsA5MT mkOME/suxPtZoRM OPx9+iZISGJHgt3YY Y31u/GTK5yOgiiQ9M 5vgwbw==

Table 5.5: Message, Cipher Text & Signature

5.2 Analysis of Experiment Results

5.2.1 Facial Dimensions

MTCNN adds an extra layer of security by pre matching procedure. At first, we selected facial landmarks and measured the distance vector in a face feature. In this procedure, we have selected two samples per person to make the model efficient. Here, we have emphasized on the facial key points to go forward of further matching

process-

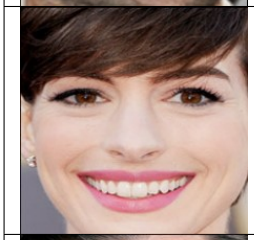
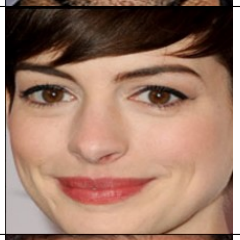
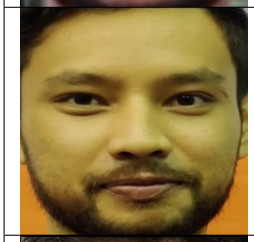
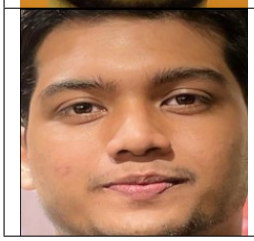
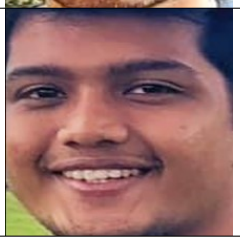
Extracted Face1	Face1 Features	Extracted Face2	Face2 Features
	'left_eye': (144, 109), 'right_eye': (184, 104), 'nose': (171, 137), 'mouth_left': (148, 149), 'mouth_right': (184, 145)		'left_eye': (281, 106), 'right_eye': (338, 103), 'nose': (312, 151), 'mouth_left': (284, 166), 'mouth_right': (342, 162)
	'left_eye': (148, 181), 'right_eye': (234, 180), 'nose': (187, 232), 'mouth_left': (145, 259), 'mouth_right': (233, 258)		'left_eye': (234, 231), 'right_eye': (326, 230), 'nose': (269, 289), 'mouth_left': (240, 317), 'mouth_right': (322, 317)
	'left_eye': (104, 186), 'right_eye': (207, 168), 'nose': (163, 239), 'mouth_left': (135, 291), 'mouth_right': (215, 274)		'left_eye': (87, 148), 'right_eye': (155, 144), 'nose': (123, 198), 'mouth_left': (90, 209), 'mouth_right': (158, 207)
	'left_eye': (608, 196), 'right_eye': (649, 194), 'nose': (629, 215), 'mouth_left': (612, 239), 'mouth_right': (648, 237)		'left_eye': (535, 405), 'right_eye': (596, 410), 'nose': (559, 439), 'mouth_left': (533, 468), 'mouth_right': (583, 473)
	'left_eye': (352, 470), 'right_eye': (545, 468), 'nose': (459, 587), 'mouth_left': (373, 692), 'mouth_right': (536, 692)		'left_eye': (661, 550), 'right_eye': (714, 554), 'nose': (683, 587), 'mouth_left': (656, 601), 'mouth_right': (710, 604)
	'left_eye': (1278, 626), 'right_eye': (1479, 608), 'nose': (1390, 721), 'mouth_left': (1310, 832), 'mouth_right': (1486, 816)		'left_eye': (1104, 1081), 'right_eye': (1174, 1078), 'nose': (1129, 1116), 'mouth_left': (1108, 1144), 'mouth_right': (1177, 1145)

Table 5.6: Facial key points

5.2.2 Length Calculation of Primes

From the above table we have seen the generated prime numbers where it was generated by the pixel value of 18*17. The calculated length of generated primes are given below:

Prime	Calculated Length
203711155111773088671155999 555113883115599999555174861 555999995511760371559999995 517603371159999516000066000 006157044660333663777571777 163155595171715551031555551 711155514671559917117555148 371555731537773446373373333 666664406333377173376004406 377711773648644400637733304 2236044400633364849	306
55503338122118050554822999 9 92213055782999999921870781 29999999221840481122222211 88308183444318477443088377 7 73287707437818344319134443 8 31122222292111111381299929 99 21992883812211883381188838 11 11147778833388388834474777 43 36643337744444777466307748 37 74340036667557383444054666 68 75507700057667	306
64444006287778644444687799333 39924408333333333399804633553 33333397804693533333339978048 99828779826621267726062772006 28169726068338666287839782795 59788797735533935539333977355 53997789999777993999822299777 87777787822288888222772628882 26282266222278228882261260668 77228826061662006288826040 87	306

Table 5.7: Length of generated primes

5.2.3 Keys & their Corresponding Lengths

The generated keys using the prime numbers which were implemented using the RSA algorithm. We have calculated the corresponding keys bit length which are provided below:

Private Key	Bits
(0x525efd9affc871748e55a04a4ea36709de43586f29f57df49df6acfc772691 fee9f5e13b13724582dce6b9fd62d704a2aba7152e0266be3283e4d0e733dc2 21d76fed76ce18d70fa077666ad025411b528f234af2b1e6f3bab9659323ba5 0814a123d73c25818a47e3a782a0bf852d247f3a4dbcf42079f6669e9d2147c 53d83fadd1226a51e766ac065b8fdaa45603baa3542febbd543135a51bc71f d80b9ee480761dabd5d513a8a4d654561cfb220d5751323eacd35ad25b6a44 632648574ae123b194fc42d35b269b1b7cd8c2ef06b756a5b95e0f66105ca90 553076cc14eecd3ba1f061cd889d4a32d92182a5b, 0xb2fa2ed02609778319a6f275585e0371fecb1b03cfc8dbbfe810bb685817c2 60b3a1fd47ecb300662a8f84409722336bc55bd12da17b5bf3573b0eaee5550e 1e18b4be54b1ab95ced484247443c38da70889d11b006722edd2d9c3c8bbccd adc785b88d3699f69982325b5e2d82c75f89d0041a38fb331c454bbfe20a2c9e ba49d43a154851cfc1c62f933eb5158f109ee5179d928d7b10f29b036204c7d4 e4f6ba25f6f15e694eba53825908096ee23610dbd7b52db2434afd12a4764573 d0ba237bb7aeab6250282659a6078fcc4c0418a4c93a4fd5cc835b9097ed180d 62a981e4e240a279ec65af38e61f258865)	2048
(0x28f760a80ac2d9256d927c3046afd794e1832926d5bf23a83a73f9f61f02cfc 73a9f6e70dc30bd9191c08f660c65d8e35dc20ed9ce2475a794e0d71b73d0d9c e8423e610851be9693d7483ef8e07c6070148d5c0d2dcb8ec2e32e04a7122dcc 62fa6bddd76de2c5339822d9a157ff21138627dcdbe4515a47eaee366d4d46e93 a5f5f4fa85095914283ec74874d2761dd6abe0ab298d3e6cc239d9b5ce048746a 8c31ee2a9d12b2d015b86dafa54c550ec0c9da5358b3ddfff519aa24a797fa1007 862d8d8f0eed1cc0ac32cab34a705854fbcfe7180d5fa9eebdadbefb956b18add db788304506e68ca3010345e36fd, 0x530af6faacf9b7c3e916f8a48021583c3fab46eb0323c6a04c513b0c70187b198 1b00deead473b94023daecfef51e3f67e19f5fadbcffb5bdc1834373588ca20e2f0b abfbdf92d3b097212d9a697673b5e4ae9f55e9bd132c064cf1a0ed9e9238d40008 8e3b45f87be24fbd0e6457341b201b44b376dcc2bae29fd455885036cd5636f7d1 97632eb42259ddc3f54d9e6c886046300bcfefd56d3cc6cd985bd7e72c2f175d6e2 ec109dfbb4d86236a6d57e8575ee73b31f8a8c9218a0f1d352a3e58b5fdf37ad646 1c85595cecdf272070cbc94d27d2a69fd5df476b12580a8f55b8674cccec9260c5d 9e5917fd2228fb)	2048

Table 5.8: Bit length of Private Key

Public Key	Bits
(0x9d4deba9b81d90ef5f93179e35f7fc8374a6027f9e6d02330f8831f06af7 f857f207825b8340b034bfb9203dfefc057244efda510c47c3154dddd21e46b 9341519bbff1daa1cba5a69990209de264f3e353715fa678f783616b59f6494 18b04de55d43bbfbd6ccfc6548816d1f0db7c2cbd4450bc6c3688446bd05960 4607dbeaa225334b4965ddf20801fe3bdac87560241e6065c941b33c3978b95 9200a4c97b32d70caca270b263124f42b13f6b28401fa0c272c6cbb85fa27e1 71c8dfec9e806ddb0b4990fbc6860f92960aab985abbd4476ddde79945cfd8e 4620e1481124ec812c143547abc38eeb9bce154149, 0x10d8dc46efb379ffd24e34eba671914894beb525e30855a57367e64eebe5e ecd2682d141088dadea2627c8cd3d458dd242681c182a2e458ce9144d072d8e f93f677719213f6520a56629bab9d6122299482c6f3924491fed0b852b25a10 0ee3e4b0b68ae71fbf3cbffa10a5e4fdfc17705a494dc3a92f9842e8361574e 18db318e8adc6c2a67a497aae7286df7351448f425e4ad8bfebf41579d31e36 79f2ddd2f16343349a4e650135d5909c8fcea240544017282f72bf96fc483be 60b4588ac915b67821e9d5c808215f1579f49e84407ece697b6c3d23dc27637 a7d3321fe7d37cdbbd8f9b3c59e24df5f6aa9c8f99)	2048
(0x17a8f6a2888303dcb6e96f4d8c9ef1fa401ef4d8580d2e7baddaa0e 49648659b6b233b428937a3b66421d3ca7527fedf34120a2c0e4d43a19 8336c5ba1cbc6e36e4cf002bbaf2231092db0bd786008d630d933bd40d 1ce730ab743216b24b4784038eaecc54ab89bf5b8f93ddaa0c9a705565 5f73d650ce3d18f851fd3675cc7625c9fdcff37d693ad5f1f8d6bc3f70 f507feaf4531aee932a52f88c9b2661612377855bf03c495ab9e5fd008 4e12d78215a736aa7eb9d628a7a7db473adc25894f6bf24578217c6010 91c1a6576212ca32b223048ced8cadbb88488e53b2635b24b4e927d6a0 4dcf0e518c03d5561f9, 0x1dfe1b7a7b442999ba25b8de685a1aa06f32e949d57ab540ce8ace52 c9a29dd4bd89d4261291a15386a30234c648a74658ccdd6db247a90e45 e1189a58279a94bdc157590c6102564a4b8d70eb9968059db701876e0e 65260bf32cb75ea2e52beeb556171eeb7f96ddf1fb5f327b3c9fd10d2c 19d11c541f897649f17e9dd14a1949e63e3a52621780dd079c47731752 bc180c1fa542c1e56277f3e87deb52e6c5db89f63792d45f34552cc1aa 4d9bdcdbf1df8e285d0f2fe1467f7a27f65208448c6ccc3931bda1d6e3 655be5f468ec3d6c8c8b5656b5e6e23f898d0266948af4432cda30d030 abbae296845d9166fd)	2048

Table 5.9: Bit length of Public Key

5.2.4 Cross-Validation

Our cross validation process was based on image hash. The image hash was created & wavelet hash process has been followed. While initiating the transaction we have gone through creating image hashes & compared them. We have observed that in this part we need to provide exact pictures to cross authenticate otherwise the same hash can not be produced & the face can not be matched. Here is the output below:

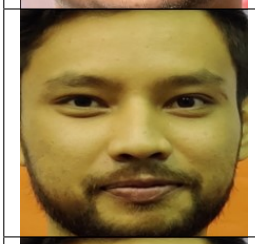
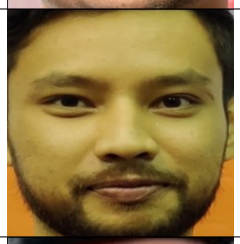
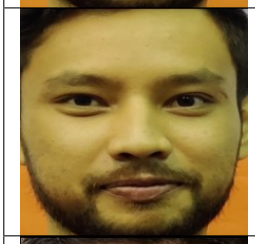
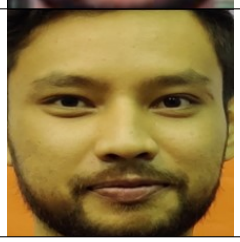
Face in MTCNN	Face in Wallet Creation	Generated Hash (MTCNN)	Generated Hash (Wallet Creation)	Compare Result
		0e7e7889fff64001	0e7e7889fff64001	Face is a match, Proceed to transaction.
		007c7018fef6f0f8	007c7018fef6f0f8	Face is a match, Proceed to transaction
		3e7ed898ffe10081	3e7ed898ffe10081	Face is a match, Proceed to transaction
		3e7ed898ffe10081	0e7e7889fff64001	Face is Not a match. Cannot make transactions
		007c7018fef6f0f8	3e7ed898ffe10081	Face is Not a match. Cannot make transactions

Table 5.10: Cross Validation

5.2.5 Performance Evaluation

We have evaluated performance based on time, generated keys & variations as well. The graphs below give us the whole idea about the generation part:

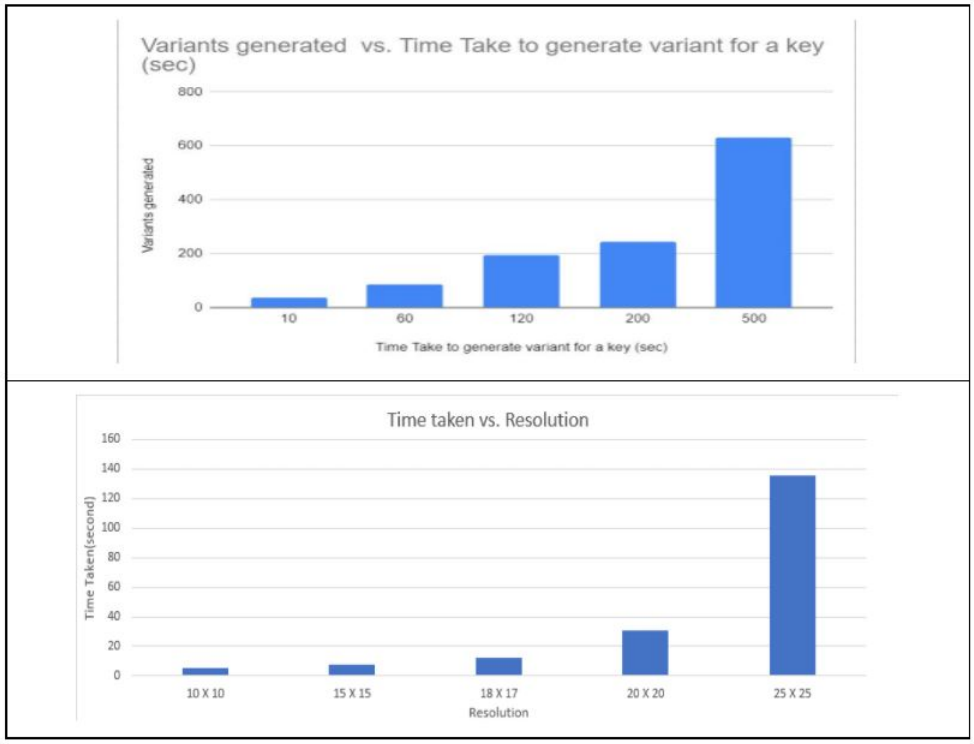


Figure 5.1: Performance Evaluation (Time based)

The above figure represents that, if we change pixel values from a picture it varies while generating keys. The higher the resolution is provided for slicing the picture the higher time utilization occurs. Thus, it also generates bigger size keys as well. Besides, we have gone through a time based analysis where we have observed the key generation time taken vs no. of iterations. Additionally, among the variant lengths we have calculated the generated variants as well.

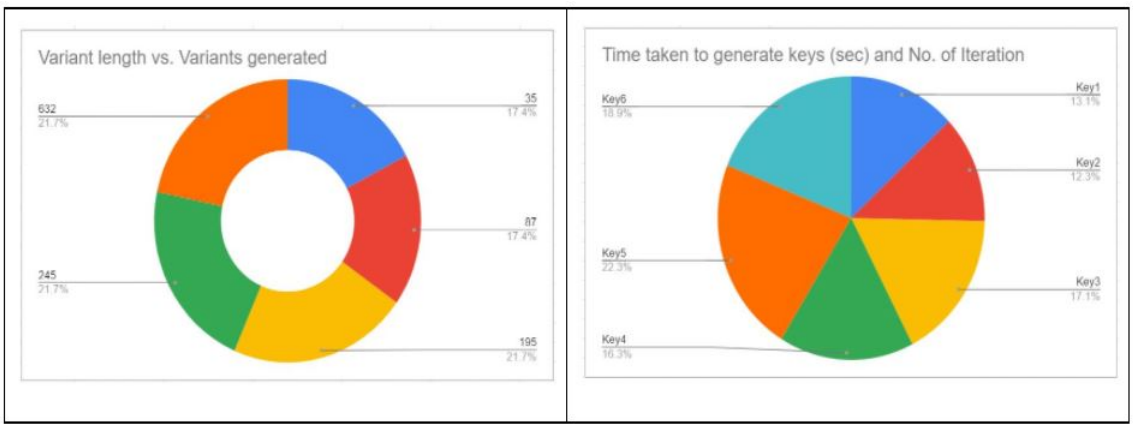


Figure 5.2: Performance Evaluation (Generation based)

Once generated we went through the Quadratic Sieve algorithm, Pollard's P-1 algorithm & Number Field Sieve algorithm which are used for large prime number factorization. Subsequently, we wanted to crack our numbers using these algorithms. Although, these three algorithms require high amount of computations we have undergone with our minimum resources to check if they give factors against our generated prime numbers. Luckily, there were no factors found.

The details of the algorithms are given below in brief:

Pollard's P-1 [8]	**Number Field Sieve[5]**	**Quadratic Sieve**
Polland's p-1 method depends on a nontrivial factor of N the algorithm utilizes a chosen value $a \in Z_N$ at random by selecting a positive integer K that is divisible by many prime powers. Then, $a_k = a^k (mod\ N)$ and find the greatest common divisor (gcd) of $f = gcd(a_k - 1, N)$ where $f = 1 < f < N$ otherwise it repeats again until the factors are obtained.	The Number Field Sieve is for the integers of the form $N = C_1 r^t + C_2 s^u$ where two irreducible polynomials $f(x)$ and $g(x)$ with small integer coefficients for which there exits an integer n such that $f(x) \equiv g(n) \equiv 0(modN)$ are formed. The polynomial should not have a common factor Q.	The Quadratic Sieve is based on Fermat's factorization method which attempts to find integers, say a and b such that $a^2 \equiv b^2\ mod\ N$ And $a \neq \pm b\ mod\ N$, but N is not divisible by $a^2 - b^2 = (a + b)(a - b)$. However N neither divides $a - 1$ nor $a + 1$ then $gcd(a - b, N)$ is the nontrivial factor of N.

The above techniques are considered threatening for the RSA system because they were specially driven techniques to factorize RSA numbers.

Chapter 6

Conclusion & Future works

6.1 Conclusion

To conclude, we have suggested a robust & unique approach to generate public keys which can be implemented in Blockchain. We have demonstrated the implementation of these keys in the Blockchain system. In a bigger context, our approach worked well in not only extracted features from the dataset but also it worked effectively when we used real life images as well. In addition, this approach indicates that a very large number which is used for large keys is quite impossible to crack & this increases security for the blockchain as well. We are satisfied that we have successfully been able to propose a new unique approach of generating primes where primes used to generate on a random basis. Lastly, our approach does not face any such difficulties while generating the keys. Although, we are committed to improving our system on a regular basis.

6.2 Future Works

The effectiveness & robustness of our procedure is undoubtedly secured & well managed. There are few things that we would like to address for our future work. The major obstruction of this research is this might lead to a growth of using deep fakes to generate the faces which can be used for generating keys & wallet creation. Besides, we faced complications while hashing the faces before the transaction. Additionally, the user's face needs to be exactly the same when the cross validation using hashing was going through. So, we want to emphasize reducing this complexity as well. Finally, there is another concern of lack of randomness in the public key generation phase. We have selected three future goals to achieve the perfect outcome from our research:

i. Creating an additional model to combat deep fake images
ii. Smooth Hashing procedure while making the transaction
iii. Addition of randomness in the public keys

We believe that adding these features & few fine tunings will help us to reach our ultimate goal. We will make sure that no unethical approach can make our system vulnerable using Generative Adversarial Network (GAN) to create a look alike wallet to confuse the blockchain system and manipulate data. Lastly, security upgradation will be our core concern while improving our system for further research.

Bibliography

[1] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 26, pp. 96–99, Jan. 1983. DOI: 10.1145/357980.358017. [Online]. Available: https://web.williams.edu/Mathematics/lg5/302/RSA.pdf (visited on 12/23/2019).

[2] C. Pomerance, "The quadratic sieve factoring algorithm," *Advances in Cryptology*, vol. 209, pp. 169–182, 1985. DOI: 10.1007/3-540-39757-4_17. (visited on 05/31/2021).

[3] T. Matsumoto, T. Yokohama, H. Suzuki, R. Furukawa, A. Oshimoto, T. Shimmi, Y. Matsushita, T. Seo, and L. Chua, "Several image processing examples by cnn," *IEEE International Workshop on Cellular Neural Networks and their Applications*, pp. 100–111, 1990. DOI: 10.1109/cnna.1990.207512. (visited on 05/31/2021).

[4] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ecdsa)," *International Journal of Information Security*, vol. 1, pp. 36–63, Aug. 2001. DOI: 10.1007/s102070100002. [Online]. Available: http://www.cs.miami.edu/home/burt/learning/Csc609.142/ecdsa-cert.pdf.

[5] A. Joux, R. Lercier, N. Smart, and F. Vercauteren, "The number field sieve in the medium prime case," *Lecture Notes in Computer Science*, pp. 326–344, 2006. DOI: 10.1007/11818175_19. (visited on 05/31/2021).

[6] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf.

[7] S. Y. Yan, *Cryptanalytic attacks on RSA*. New York, Ny Springer, 2008.

[8] B. Źrałek, "A deterministic version of pollard's p-1 algorithm," *Mathematics of Computation*, vol. 79, pp. 513–513, Jan. 2010. DOI: 10.1090/s0025-5718-09-02262-5. (visited on 05/31/2021).

[9] A. Barenghi, G. Bertoni, A. Palomba, and R. Susella, "A novel fault attack against ecdsa," *2011 IEEE International Symposium on Hardware-Oriented Security and Trust*, Jun. 2011. DOI: 10.1109/hst.2011.5955015. (visited on 05/31/2021).

[10] M. J. Dubai, T. R. Mahesh, and P. A. Ghosh, "Design of new security algorithm: Using hybrid cryptography architecture," *2011 3rd International Conference on Electronics Computer Technology*, pp. 99–101, Apr. 2011. DOI: 10.1109/icectech.2011.5941965. (visited on 05/31/2021).

[11] X. Zhou and X. Tang, *Research and implementation of rsa algorithm for encryption and decryption*, IEEE Xplore, Aug. 2011. DOI: 10.1109/IFOST.2011.6021216. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6021216.

[12] F. Q. Lauzon, "An introduction to deep learning," pp. 1438–1439, Jul. 2012. DOI: 10.1109/ISSPA.2012.6310529. [Online]. Available: https://ieeexplore.ieee.org/document/6310529 (visited on 03/21/2021).

[13] D. P. Mahajan and A. Sachdeva, "A study of encryption algorithms aes, des and rsa for security," *Global Journal of Computer Science and Technology*, vol. 13, Dec. 2013. [Online]. Available: https://computerresearch.org/index.php/computer/article/view/272.

[14] A. Abubakar, S. Jabaka, I. Bello, B. I. Tijjani, A. Zeki, H. Chiroma, M. Joda Usman, S. Raji, and M. Mahmud, "Cryptanalytic attacks on rivest, shamir, and adleman (rsa) cryptosystem: Issues and challenges," *Journal of Theoretical and Applied Information Technology*, vol. 61, 2014. [Online]. Available: http://www.jatit.org/volumes/Vol61No1/5Vol61No1.pdf (visited on 05/31/2021).

[15] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," *2016 2nd International Conference on Open and Big Data (OBD)*, Aug. 2016. DOI: 10.1109/obd.2016.11.

[16] V. Kapoor and R. Yadav, "A hybrid cryptography technique for improving network security," *International Journal of Computer Applications*, vol. 141, pp. 25–30, May 2016. DOI: 10.5120/ijca2016909863. (visited on 10/02/2019).

[17] D. Savić and S. Damjanović, "The attacks on the rsa algorithm," *Proceedings of the International Scientific Conference - Sinteza 2016*, 2016. DOI: 10.15308/sinteza-2016-131-136. (visited on 05/31/2021).

[18] A. Sghaier, M. Zeghid, and M. Machhout, "Fast hardware implementation of ecdsa signature scheme," *2016 International Symposium on Signal, Image, Video and Communications (ISIVC)*, pp. 343–348, 2016. DOI: 10.1109/isivc.2016.7894012. (visited on 05/31/2021).

[19] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, Aug. 2017. DOI: 10.1109/icengtechnol.2017.8308186. [Online]. Available: https://ieeexplore.ieee.org/document/8308186/.

[20] P. Bailis, A. Narayanan, A. Miller, and S. Han, "Research for practice: Cryptocurrencies, blockchains, and smart contracts; hardware for deep learning," *Communications of the ACM*, vol. 60, Apr. 2017. DOI: 10.1145/3024928. [Online]. Available: http://www.cs.unibo.it/~montesi/CBD/Articoli/Research%5C%20for%5C%20practice-%5C%20cryptocurrencies,%5C%20blockchains,%5C%20and%5C%20smart%5C%20contracts%5C%3B%5C%20hardware%5C%20fo%5Cr%5C%20deep%5C%20learning.pdf (visited on 08/31/2019).

[21] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, "Blockchain," *Business & Information Systems Engineering*, vol. 59, pp. 183–187, Mar. 2017. DOI: 10.1007/s12599-017-0467-3.

[22] D. Petrov, *Wavelet image hash in python*, Medium, Nov. 2017. [Online]. Available: https://fullstackml.com/wavelet-image-hash-in-python-3504fdd282b5.

[23] J. I. Ahmad, R. Din, and M. Ahmad, "Analysis review on public key cryptography algorithms," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 12, pp. 447–454, Nov. 2018. DOI: 10.11591/ijeecs.v12.i2.

[24] C. Ye, G. Li, H. Cai, Y. Gu, and A. Fukuda, "Analysis of security in blockchain: Case study in 51%-attack detecting," *2018 5th International Conference on Dependable Systems and Their Applications (DSA)*, Sep. 2018. DOI: 10.1109/dsa.2018.00015.

[25] A. Albakri and C. Mokbel, "Convolutional neural network biometric cryptosystem for the protection of the blockchain's private key," *Procedia Computer Science*, vol. 160, pp. 235–240, 2019. DOI: 10.1016/j.procs.2019.09.462. (visited on 05/31/2021).

[26] O. Pal, B. Alam, V. Thakur, and S. Singh, "Key management for blockchain technology," *ICT Express*, Aug. 2019. DOI: 10.1016/j.icte.2019.08.002.

[27] S. Rathore, Y. Pan, and J. H. Park, "Blockdeepnet: A blockchain-based secure deep learning for iot network," *Sustainability*, vol. 11, p. 3974, Jul. 2019. DOI: 10.3390/su11143974.

[28] P. J. Taylor, T. Dargahi, A. Dehghantanha, R. M. Parizi, and K.-K. R. Choo, "A systematic literature review of blockchain cyber security," *Digital Communications and Networks*, Feb. 2019. DOI: 10.1016/j.dcan.2019.01.005.

[29] Venkatraman and Overmars, "New method of prime factorisation-based attacks on rsa authentication in iot," *Cryptography*, vol. 3, p. 20, Aug. 2019. DOI: 10.3390/cryptography3030020. (visited on 10/08/2019).

[30] E. Vidhya, S. Sivabalan, and R. Rathipriya, "Hybrid key generation for rsa and ecc," *2019 International Conference on Communication and Electronics Systems (ICCES)*, Jul. 2019. DOI: 10.1109/icces45898.2019.9002197. (visited on 05/31/2021).

[31] M. Aydar, S. Cemil, Ç. Etin, S. Ayvaz, and B. Aygün, *Private key encryption and recovery in blockchain*, 2020. [Online]. Available: https://arxiv.org/pdf/1907.04156.pdf.

[32] E. Hechler, M. Oberhofer, and T. Schaeck, "Ai and blockchain," *Deploying AI in the Enterprise*, pp. 253–271, 2020. DOI: 10.1007/978-1-4842-6206-1_11. (visited on 06/01/2021).

[33] P. B. B. OBE, *So how many bits does the prime number have?* Medium, Jul. 2020. [Online]. Available: https://medium.com/asecuritysite-when-bob-met-alice/so-how-many-bits-does-the-prime-number-have-e5dbbdf568ea.

[34] O. M. Parkhi, A. Vedaldi, and A. Zisserman, *Visual geometry group - face dataset*, www.robots.ox.ac.uk. [Online]. Available: https://www.robots.ox.ac.uk/~vgg/data/vgg_face/#publi (visited on 05/31/2021).

[35] F. Scicchitano, A. Liguori, M. Guarascio, E. Ritacco, and G. Manco, *A deep learning approach for detecting security attacks on blockchain **. [Online]. Available: http://ceur-ws.org/Vol-2597/paper-19.pdf (visited on 05/31/2021).