Transaction Verification Model for Peer-to-Peer Service-Oriented Digital Currency Transactions Based on the Foundation of Blockchain Architecture

I. D.Rubasinghe

Instructor, Computer Science, University of Colombo School of Computing, Colombo, Sri Lanka
Email: idr@ucsc.cmb.ac.lk

Abstract

Digital payment systems are an evolving field in present day with the recent enhancements in seamless digital currencies. Thus, despite the benefits of cryptocurrency based digital payments systems, their adoption and diffusion within general payment platform domain are significantly hindered. Blockchain architecture is widely recognized as a promising mechanism to support the management of cryptocurrency related transactions. However, ensuring the security of digital payment transactions is a challenging task due to various security threats and existing prevention mechanisms that are either computationally expensive or domain dependent. Among many, the Man-in-the-Middle attack and Double Spending are identified as key security vulnerabilities.

The purpose of this study is to investigate the means of addressing the said security issues by proposing a feasible transaction verification methodology; targeting a common payment platform that integrates different vendor based digital currencies together. The currency miners and the user applications are identified as the core components that cooperate with transactions. Accordingly, a scenario based transaction verification model is designed by considering transaction patterns among miners and user applications. The bitcoin-similar concept of 'trust network' is adopted in verifying transactions via building a trusted network among currency miners in the payment platform using digital signatures along with SHA-256 hashing and RSA algorithm. In strengthening the verification level, an approach of acknowledgements is defined associated with a minimum required level of probability. Furthermore, a time constraint is set depending on the peer-to-peer network conditions for a particular transaction to get completed with proper verification.

It is explored the strength and feasibility of the proposed methodology in the perspective of transaction verification over man-in-the-middle attack using a probabilistic evaluation where the possibility of a transaction getting verified decreases proportionally when the trusted network of miners getting unhealthy. Also, the double spending prevention is evaluated using the implementation of a mobile-based digital wallet as the proof-of-concept.

Keywords: Peer-to-Peer, Man-in-the-Middle Attack, Double Spending, Anonymity, Cryptocurrency, Digital Payment Platform

1. INTRODUCTION

The convergence of digital currencies, digital wallets and peer-to-peer payment systems has caused a fundamental upheaval. Digital currency payment transactions are immediate regardless of the payment method, payer's location or payment currency. Many consumers select these advantages and move away from traditional payment services. Bank and merchant service providers are disrupted through mainstream acceptance of cryptocurrency payment services for peer-to-peer payments (S.Nakamoto, 2012). The costs of payment services have dropped for a number of reasons. Firstly, digital currency payments do not have the transaction costs as

traditional banking systems and payment services. Secondly, digital currencies do not have the same policing and enforcement costs as fiat currencies adding another transaction cost advantage (M.Bawa, et al., 2007). The security requirements for each involved party of a payment transaction vary but with an equal importance in achieving a higher security level. Protection against security vulnerabilities and the performance of transactions is significant in a payment related system. The requirement is not only in verifying the accurate destinations securely but also confirming the atomicity of each transaction (E.Nordstrom, 2015). Therefore from a small scale payment system to a larger digital payment platform a proper transaction verification model is identified as mandatory.

1.1 Problem overview

In the field of digital currencies, a digital transaction is a topic of ever increasing importance. Technically the blockchain architecture is a powerful foundation for handling digital transactions between peers. The effects of digital transactions have a strong impact on a vast number of categories including economy, security and can affect individuals' privacy too (J.Wells, et al., 2008). It is therefore of utmost importance to be able to identify properly and verify digital transactions with a higher level of accuracy. Understanding the key threats and attacks happening on digital currency involved transactions will play a key role in this pursuit. Two of the widely known such threats are man-in-the-middle attack and double spending problem. Hashing, public key-private key mechanisms are basic elements of addressing them. The most popular *bitcoin* has strongly addressed these security related issues for the computation power based digital currencies (S.Nakamoto, 2012). But still, the problematic security threats exist for non-computational powers based currency systems such as service-oriented digital currency platforms (See Appendix A – SCPP Common Payment Platform). Therefore a digital currency transaction verification model for a service-oriented payment platform is required to be built with the goal of providing greater insight into the process.

1.2 Aim

This research project will compare different transaction verification techniques and determine optimal solutions for a service-oriented digital currency system based on blockchain architecture where the transaction verification over man-in-the-middle attack and double spending problem are prioritized. The final outcome of the project is to have a secure peer-to-peer transaction model capable of verifying the transactions on possible scenarios. Security along with the speed and efficiency are also aimed to be considered in performing transaction verifications.

1.3 Objectives

In the process of achieving the aim of transaction verification, the following are the objectives extended as the research progressed;

- Determining possible scenarios of transactions in a common payment platform (See Appendix A – SCPP Common Payment Platform)
- Selection of optimal security measurements over man-in-the-middle attack in all possible transaction scenarios
- Determining prevention methodology for double spending of service-oriented digital currencies
- Designing and implementing suitable protocol structures in order to provide secure transactions among peer-to-peer nodes in the payment platform
- Designing a feasible transaction verification model with integration of security measurements and protocols
- Designing a mobile-based digital wallet in applying the transaction verification model and performing the evaluation

1.4 Scope

The research approach is constrained within the below mentioned scope based on the complexity of the questions to be handled, availability of resources and the time frame;

- The peer-to-peer network model is selected as the base environment for performing all transactions in the selected payment platform (M.Bawa, et al., 2007) (M.Srivatsa, et al., 2009)
- Designing a transaction verification model that is cooperated with the blockchain architecture which is actively used in existing digital currency involved systems
- The transactions are considered to integrate with computationally not complex digital currencies such as service-oriented coins that are mined per service requests made by platform registered users (N.T.Courtois & L.Bahack, 2014)
- Among various possible security vulnerabilities on a payment system, only the man-in-the-middle attack and the double spending problem are significantly addressed with a higher priority in this research work (Eriksson, 2004) (G.O.Karame, et al., 2012)
- The security concerns; anonymity and transaction atomicity are addressed with a minor priority in building the transaction verification model

The service-oriented coins mining and the peer-to-peer network management are excluded from the scope of this thesis, only the science behind them is discussed.

2. BACKGROUND RESEARCH

This chapter depicts the background of the problem, literature review and related work. Digital currency is a broad approach of monetary exchange in which the value is only transferred electronically. Most of the digital payment systems use peer-to-peer transactions where no third party to verify transactions (M.J.Casey & P.Vigna, 2015). Hence it is important to ensure the security and verification of digital currency transactions among nodes.

With the evolvement of information technologies in modern day lot of industries has adopted to automate the tasks. From small shops to big factories information technology is used to some extent. Some are fully automated and some function partially along with human tasks. Online shopping is popular among people due to this technological evolution in the modern day. With the popularity of online shopping, the necessity to use the technology in a secure manner is essential without exposing users' sensitive data. With the need of above requirements mobile payment protocols were introduced to manage digital currency involved transactions (R.P.D.T.Rajapaksha, 2015). Those protocol stacks and transaction verification models were developed for few years and it keeps on evolving and adopting new technology to stand with the modern day security threats. Hence the security aspect of the transaction verification is vital (A.Upadhayaya, 2012).

2.1 Issues in digital currency transactions

Taxonomy of major vulnerabilities at different layers and their effects on a mobile based payment system can be summarized as in following

Figure 2.1.1 Vulnerabilities Affecting a Mobile Payment System.

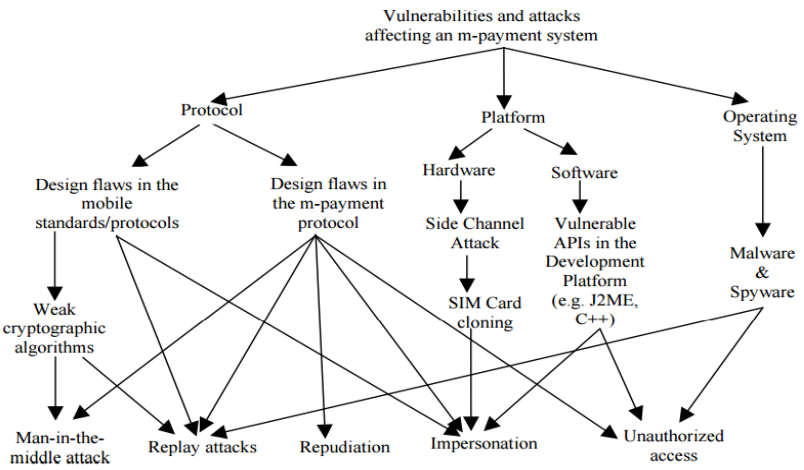

Figure 2.1.1 Vulnerabilities Affecting a Mobile Payment System **(D.Arthur & A.Bazaz, 2007)**

2.1.1 Double spending attack

Double spending attack which is transmitting same digital currency twice in different transactions is a major issue in most of the digital payment systems. Hence it is necessary to keep an identity to the generated digital coins based on some criteria. One way to safeguard from this attack is to maintain a third party authorized person to verify transactions within the network (M.Lei, 2015). However, it is not applicable in p2p networked payment systems subsequently there is no centralized control over the network (M.Bawa, et al., 2007).

2.1.2 Anonymity

General public attracted to electronic payment systems if there exists some degree of anonymity to the user. If payment systems are able to offer full anonymity to the customers, general public accepts such systems hence those systems preserve their privacy (J.Wells, et al., 2008). Currently available payment systems provide complete anonymity to vendors but partial anonymity to customers. Most of the existing payment systems achieve anonymity through third party institution where trusted third party will be provided with additional information on the coin and the user (Katina, 2009). In these kinds of systems, it is possible to trace the owner of a coin or to trace the coins originating from a specific withdrawal.

2.1.3 Man-in-the-middle attack

Ordinary users do not have a clear awareness of their responsibility and therefore make it possible for an aggressor to target a man-in-the-middle attack which is described especially when describing security in the cryptographic protocols. The man-in-the-middle attack is discussed as a main theoretical possibility but also practically inconceivable (Eriksson, 2004). Regarding the Internet, this has been discussed in different steps where IP-spoofing is considered as the first step towards a working man-in-the-middle attack. It can be identified as a technique where the source address of an IP-packet is forged. The issue in applying this technique is about the ability to get answers since they are sent to the forged addresses. It is proved that there exist scenarios where it is possible to exploit a trust relationship in a computer system by masquerading as a trusted counterpart via using IP-spoofing (S.Nakamoto, 2012).

2.1.4 Relay attack

Relay Attack is another security concern related to a digital currency that cannot be prevented by application-level cryptography such as encryption (S.Patil, et al., 2014). A relay attack is a simple range extension of the contactless communication channels which requires three components: a card emulator device to communicate with the actual reader, a reader device in close proximity to the card under attack and a fast communication channel between these two devices. The attack happens by bringing mole in proximity to the card under attack and at the same time card emulator also brought into proximity of a reader device. Every command that the card emulator receives from the actual reader is forwarded to the mole. The mole in turn forwards command to the card under attack. Then card's response is received by mole and sent all the way back through card emulator to an actual reader. Google Wallet has addressed this concern and has overcome this by installing secure element applets since June 2012 in new versions (R.Handa, et al., 2011) (M.Roland, 2013).

2.2 Existing cryptocurrency transaction verification mechanisms

The *CAFE* Consortium which consists of thirteen European institutions had applied cryptographic techniques and had produced a secure and open system for consumer payments using electronic money which consists of a *'CAFE* infrared wallet' and a card (Katina, 2009). This is developed as a public key system for electronic wallets which combines an electronic wallet with other applications such as digital passports, driving licenses and house keys. It allows consumers to confirm payments with their own devices.

Electronic Wallet is another type of digital wallet (also known as an E-wallet) which allows users to make electronic commerce-related transactions fast and securely (A.Upadhayaya, 2012). One of the popular examples for an E-wallet on the market is 'Microsoft Wallet'. The user needs to set up a Microsoft Passport in order to obtain Microsoft Wallet. Then after establishing a Passport, a Microsoft E-wallet can be established and E-wallets can be used for micro-payments. Microsoft Passport consists of many services including a single sign-in, wallet and kid's passport services.

Normally digital wallets are stored on client-side and are self-maintained. A server-side digital wallet is known as a thin wallet and is one that organizations create and maintains on their servers (Pentaho, 2015). Server-side digital wallets also are gaining popularity among retailers due to the efficiency, security and added utility it provides to end users that which increases their enjoyment of the overall purchases (T.Bamert, et al., 2014). One of the key points to take from current digital wallets is that they are composed of both digital wallet systems and digital wallet devices. There are dedicated digital wallet devices such as biometric wallet by *Dunhill*, where it is a physical device holding someone's cash and cards along with a Bluetooth mobile connection.

In E-wallets for further safety, it is encouraged to take backups of E-wallet files including all important information. The easiest way of doing it is by using 'Automatic Backup' feature available on Windows PC. On this platform E-wallet automatically makes a backup of the wallet file each time user closes E-wallet.

Bitcoin's blockchain wallets make use of universal public ledger known as 'blockchain' in order to transmit messages over the network whenever a transaction takes place. The transactions are secure because, by using cryptography, the messages that communicate in the network cannot be reversed, altered with, or corrupted (S.Nakamoto, 2012). Furthermore, by using a public ledger, the transactions can be verified publicly and communicated to all parties in the network. Because the blockchain ledger is not operated by a particular person or company, the *bitcoin* protocol enables transactions to take place without a central authority. *Bitcoin's* transaction verification mechanism is somewhat simplified though cannot apply for mobile based payment systems due to the heaviness of its blockchain (M.J.Casey & P.Vigna, 2015).

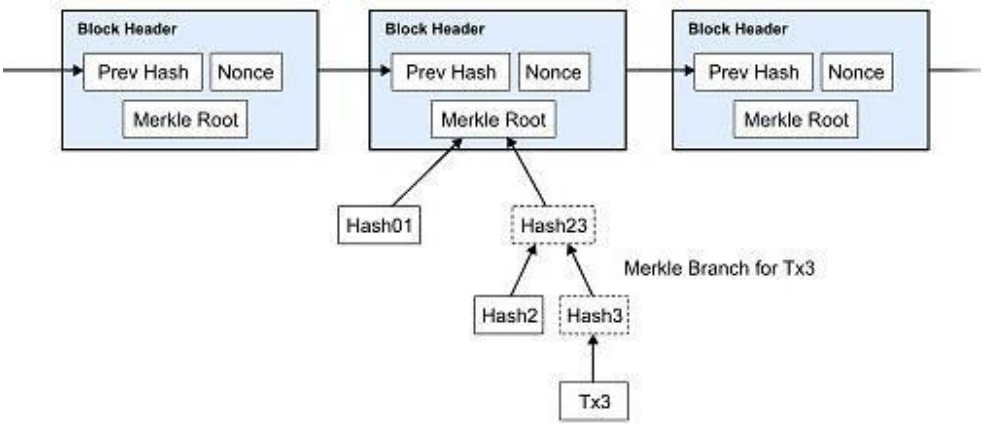

Figure 2.2.1 Bitcoin Transaction Verification Model **(S.Nakamoto, 2012)**

Here it is possible to verify payments without running through a complete network node. A user is only required to keep a copy of block headers of the longest proof-of-work chain. The user can get it by querying network nodes until it is convinced as that particular user is having the longest chain. But a user cannot check a transaction by himself unless by linking it to a position in the chain (I.Eyal, et al., 2016). Then the user can see that a network node has been accepted it. And the blocks added after it further confirms that the network has accepted it. As such the verification remains reliable as long as the honest nodes control the network as a trust network. But also is more vulnerable if the network is overpowered by a single attacker or by a group of united attackers since the network nodes can verify transactions for themselves (P.McCorry, et al., 2016) (M.Lei, 2015). A strategy to overcome this would be accepting alerts from network nodes when an invalid block is detected. It can be done by prompting a user's software to download the complete block and confirm the inconsistency to alerted transactions. But this would be not that feasible for mobile devices since downloading such heavy blocks into a mobile device would be problematic.

Most of the electronic wallet systems are based on digital signatures and cryptographic methods of certifying the origin of a digital message (T.Bamert, et al., 2014). In the majority of them, the signing key and

the authenticating key are the same and for protection stored in tamper-resistant hardware modules. But this symmetric security mechanism reduces the flexibility of a system and also security modules cannot be given out indiscriminately (R.Handa, et al., 2011). As a solution for that comes the asymmetric system using public key digital signatures where the signing and authenticating keys are different. This solution has become ideal for electronic wallets mainly because the authentication key can be made public and need not be secured. But this is not widely in use because as this is a one-way function it is impractical to invert though it is efficient to compute. Many cryptographic protocols are based upon assumptions of their intractability (D.Basin, et al., 2014).

RSA cryptosystem has overcome those problematic scenarios and it got to be used in digital signatures. However, the commercially widely used cryptosystem has been the *Data Encryption Standard (DES)* (M.Karpinskyy & Y.Kinakh, 2003). Though it is symmetric it has got the advantage of not relying on the time-consuming modular arithmetic.

SWAPEROO Architecture is a non-web-centric, symmetric and client driven architecture for digital wallets. The interaction among a peer wallet and a client wallet roughly works as follows: Once a session is initiated by client and peer wallet prepares to service the client (N.Daswani, et al., 2000). The client has the capability of deciding the instrument classes to be available on peer wallet. Then select an instrument class which is common to both peers. After that protocol management functions are asked to decide which available protocols can be used to conduct operations on an instrument of a selected class. A protocol is selected depending on what protocols are shared. The protocol supports some operations for that selected instrument class. And client may invoke those operations on an instrument instance.

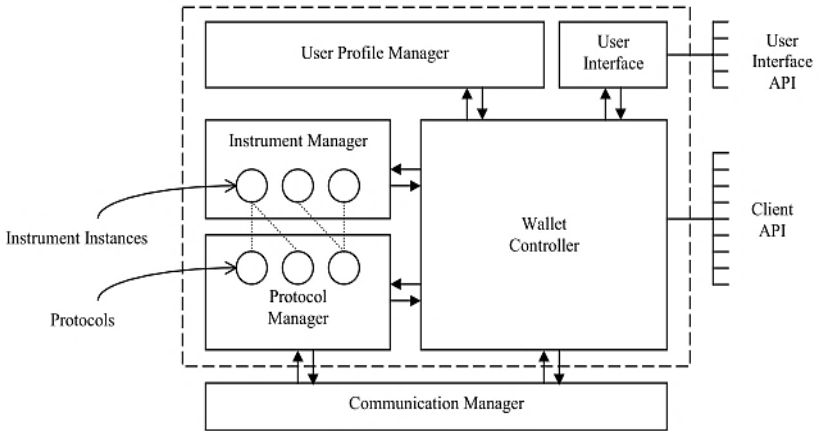

Figure 2.2.2 The SWAP Generalized Digital Wallet Architecture **(N.Daswani, et al., 2000)**

Splitting a single coin value is another core concern that is challenging to be achieved. The first practically divisible and untraceable off-line cash scheme whose cryptographic security assumptions are theoretically clarified is published via *NTT Laboratories*, Japan. This scheme that is 'single term' in which every procedure can be executed in order of $log\ N$, where N is the precision of divisibility such that;

$$N = \frac{Total\ Coin\ Value}{Minimum\ Divisible\ Unit\ Value}$$

Therefore this scheme was efficient and practical. For example:- when $n = 217$ (total value is about $\$1000$. The minimum divisible unit is 1 cent), this scheme requires only about $1\ Kilobyte$ of data be transferred from a customer to a shop for single payment and about 20 modular exponentiation for a single payment. In addition, it is also proven the security of this cash scheme under some cryptographic assumptions (M.Andrychowicz, 2015). Recent *bitcoin* has achieved this and their coins are divisible. A *satoshi* is one hundred millionth of a *bitcoin*. Also, it is possible to send a transaction as small as 5430 *satoshis* on the *bitcoin* network (H.B.Shadab, 2014).

Mobile-based transactions systems are identified in two types such as Remote Payment Systems and Proximity Payment Systems. In the most general scenario, the customer uses a mobile device and sends a payment request to a PSP over a wireless network which includes the details of the payee and the amount to be paid (M.Wachs, 2015). The PSP verifies those credentials of customer and the payee by checking whether the customer and payee had registered for such a mobile payment service. At that point, a PSP might ask the customer for more details such as a password for two level authentications to further verify the transaction. Once the credentials are verified the PSP requests the payee for a confirmation by forwarding the particular payment details and the payee then sends a confirmation message to PSP. After that two-way successful confirmation, the PSP performs certain backend processing in order to update the accounts of payer and payee. It might send a payment receipt to the payer and send a 'Transaction completed' message to the payee to notify the verification and completion of the transaction.

Authenigraph is another option to provide security against a variety of attacks known within the online transaction environment. It uses an image processing related methodology on a transaction's sensitive data verification process which can be either a numerical image or an alphabetical image (A.Upadhayaya, 2012). The research work has presented a succeeded conceptual model for authentication and transaction verification by using *authenigraph*.

2.3 Double spending attack and solutions

Mostly allocating payment transaction verification for a third party may introduce trustworthiness issues. Merchants provide reversible payments for users facilitating them to return services on any disagreement (M.Lei, 2015). In such situations, double spending problem occurs since payments are handled by a single trust party. Although digital signature provides a solution to double spending in transactions for some extent, still it is based on a trust based model. To achieve this without any third party, the community must agree on a single transaction history. And once any transaction has done participants must announce it to the whole community publicly. *Bitcoin* has proposed proof-of-work to record a public history of transactions in a peer-to-peer network (H.B.Shadab, 2014). But this design has a tendency to attack user's private coins if anyone gets access to the private key of the account. In *bitcoin* users execute the payments by digitally signing the own transactions. They are prevented from double spending their coins such that signing over the same coin to two different users through a distributed time stamping service. The service operates on top of *bitcoin's* peer-to-peer network which confirms that all the transactions and the order of their executions are available to all *bitcoin* users.

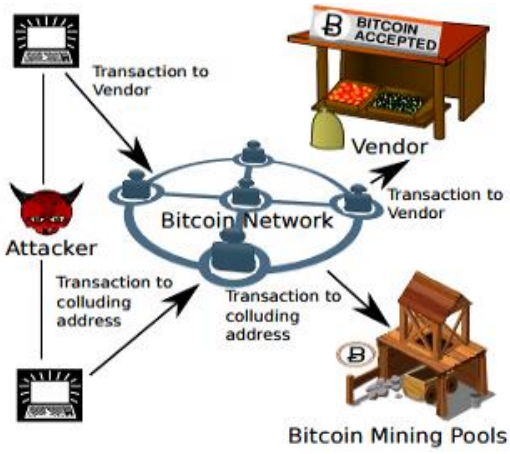

Figure 2.3.1 A Double Spending Attack against Fast Payments in Bitcoin **(G.Karame, et al., 2015)**

As diagrammed in the above *Figure 2.3.1 A Double Spending Attack against Fast Payments in Bitcoin*, in typical cases the attacker *A* dispatches two transactions which use the same *bitcoins* in the *BTC* network. The double spending attack is deemed if the *bitcoins* that *A* used to pay for vendor *V* cannot be redeemed such that when the second transaction is included in the upcoming *BTC* block. It is then convenient to determine if a transaction is valid and that does not lead to a double spending by checking if the items transacted have already been spent on the public ledger (M.J.Casey & P.Vigna, 2015) (N.T.Courtois & L.Bahack, 2014). However, it requires a guarantee that the ledger holds accurate information.

A blind signature scheme allows the user to get a message signed by a signer, without revealing the contents of the message to the signer. Messages are passed to receiver embedding it into an envelope. Therefore nobody can read inside and ensure the anonymity of the user. In e-cash system coins are signed by the bank. Hence bank sign in blinded manner bank cannot link user who withdraws the coin with whom finally it obtained. To safeguard transactions from double spending attack bank keeps a list with all spent coins and compare when the transaction is ongoing with that list (B.Pretre, 2005).

2.4 Man-in-the-middle attack and solutions

MitM attack spreads all types of transaction methods inclusive of mobile payment platforms. Research works have put a great effort in finding out possible scenarios and in figuring out applicable methodologies. MitM attack can successfully invoke attacks such as DNS spoofing, Denial of service and Port stealing (P.K.Mishra, 2012). Securing the exchange of public keys in SSP based on ARP spoofing is identified as a partial solution. In order to that, the exchange of public keys become more secure and the process of SSP would be secure (J.Wells, et al., 2008).

In the e-commerce, mobile payment systems in order to overcome MitM, it is essential to strengthen the authentication and the strength of the transaction network (Eriksson, 2004). Algorithmic tools for overcoming MitM attack regarding e-commerce related payment websites are in consideration in research works.

In the *bitcoin* model, MitM attack is minimized due to its trusted network technique. And there does not exist a currency in any form of a hash or vice versa. It is solely a note down in a ledger as a form of a text file. Also, *bitcoin* involves with a payment protocol called *BIP* which is accepted as a standard in *BIP70*. It describes a protocol for communication between a merchant and customer by enabling both a better security against MitM attacks on the payment process (P.McCorry, et al., 2016).

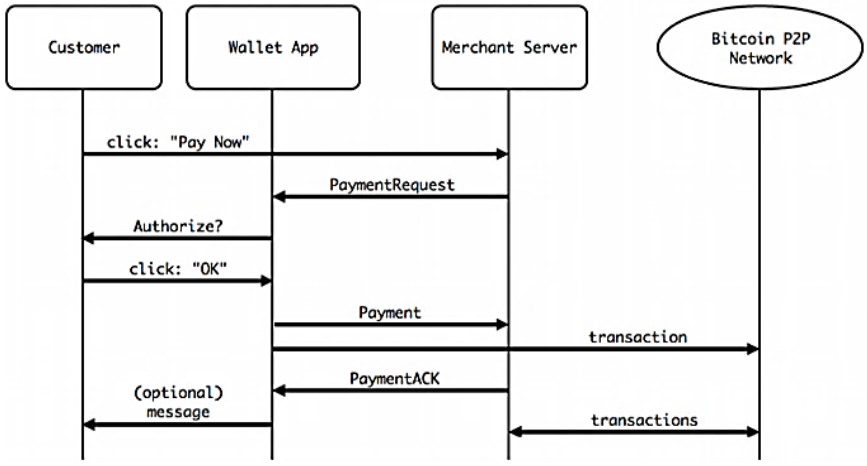

Figure 2.4.1 Overview of the BIP Payment Protocol **(P.McCorry, et al., 2016)**

Therefore technically a MitM attack is mostly prevented. But still, there exists a possibility of an attack by the re-arrangements of transactions organized by users. A scenario of re-arrangements can be designed as follows (R.Savita & U.Datta, 2015).

1. A scammer contacts a user on a trading related website saying wants to sell some *bitcoins*.
2. After the user initialized the trade with scammer; the scammer contacts user to buy the same amount of *BTCs*.

3. The user gives the bank account to scammer as in a normal trade procedure and then the scammer gives that bank account to the user.
4. The user makes a deposit/ transfer to own bank account.
5. The user receives the money and releases *BTCs*.
6. The scammer gets *BTCs* for free and disappears.
7. User loses money, reputation and could get investigated by police. Or user would have to return the money by losing *BTCs* since *BTC* transactions are not reversible.

As per *bitcoin* specification, such circumstances are advised to be reduced by the awareness among users.

2.5 Discussion

Based on these related works there exist many complexities related to digital currency transaction verifications. Double Spending and Man-in-the-Middle attacks are two issues among many others. The verification model needs to be solely based on the payment related currency-regulating model in a particular system. In the popular *bitcoin*, the both double spending and MitM attacks were minimized mostly due to the coin concept of it. But those particular verification mechanisms are not applicable for an exact mobile-based digital currency involved payment platform. And most of the other solutions are solely based on e-commerce related web portals transaction verifications which have to be re-considered when relating to mobile based payment platforms. Therefore it is quite observable that there are breaches in the existing approaches. A combined mechanism of these features would be essential when relating to a service-oriented digital currency platform to verify the transactions especially over double spending attack and MitM attack. It is considered as the emphasis of this research work and a possible solution is presented through the rest of the chapters.

3. DESIGN

This chapter is focused on the detailed design of the secure transaction handling to overcome Man-in-the-Middle attack and Double Spending in a service-oriented common payment platform. The target system is a common payment platform where digital currencies get mined per service requests as service-oriented coins. The system architecture of a common payment platform mainly involves with a cryptocurrency miner that mines currencies as rewards based on the service requests. And application nodes that represent platform's registered users who claim for rewards from services or spend previously collected rewards on services (See Appendix A – SCPP Common Payment Platform).

3.1 Design assumptions

The supporting hypothetical features that are excluded from the scope of this thesis are declared as assumptions in the design solution as stated in the following list.

- The payment platform is based on a peer-to-peer network (Bellovin, 2013)
- Blockchain architecture is involved at the currency miners to maintain transaction records history
- Only online payment transactions are included in consideration excluding offline transactions
- Payment platform's currency miners generate coins based on buying power, instead of high computational power. Hence less resource consumption is involved
- Currency mining is service-oriented which are triggered per users' service consuming requests as rewards
- A single coin get generated per time and only one coin is transferred at a time in transactions
- A trusted network of miners is established among all the currency miners in the platform
- Coin verification is excluded and only the transaction verification is considered
- The user applications are based on mobile smart devices with Internet connection

3.2 Conceptual solution

A set of protocols depending on the basic functionalities required to be triggered in a common payment platform is identified to be designed. The one-way hashing is adapted to the proposed model in order to bundle the protocol integrated data. The digital signature mechanism along with a strengthen asymmetric algorithm is designed to apply on top of the hashing and obtain a signature to integrate with each protocol. Asymmetric over symmetric is selected since digital currency is a core asset in the system (R.Tripathi & S.Agrawal, Comparative Study of Symmetric and Asymmetric, 2014). Digital signature is designed to use for verifying each protocol integrated data at each peer end when sent over the network. A transaction can either contain a coin or details of a payment. By analysing the content of the transaction it is identified that all possible forms of the content contain a significant importance since a payment system. Therefore the digital signature process is selected for each transaction. In the existing *bitcoin* system, the coins are stored in a *bitcoin* wallet (See Appendix C – Bitcoin Blockchain Wallet Users), which is also designated by a public key (Bitcoin.org, 2017).

A waiting time constraint is designed to apply for each transaction depending on the implementation and network performance. Any p2p transaction that does not complete within that defined time constraint are dropped and canceled. The MitM attack prevention is primarily addressed in these design steps based on the conducted literature reviews on similar systems.

The public-private key pairs are generated for each member of the payment platform; either a miner or a user application, at the registration with the adaptation of an asymmetric algorithm (R.Tripathi & S.Agrawal, Critical Analysis of RSA Public Key Cryptosystem, 2014). Each public key is designed to make available to every registered node within the payment platform. It is designed to distribute the public keys as new versions of the miners and user applications whenever a new component gets registered, without transferring public keys over the network. It is the foundation used in building a trusted network of miners in the payment platform. The miners are considered as trusted and eligible to verify any transaction per verification request sent by another component. The concept of *bitcoin's* blockchain architecture is adapted to maintain the transaction details history at every miner. Therefore the blockchain architecture is designed to be used as the foundation of this trusted network of miners (See Appendix B – Coin Wise Blockchain Architecture).

A probability level criterion of 75% is defined in the solution model to further enhance the trusted network accuracy. More than or equal 75% of verified positive responses are required from the trusted network of miners in order to completely accept a particular transaction as verified. A lesser probability transaction is designed to be dropped as a solution for double spending prevention.

3.3 Protocol design

Five major purposes of transactions are identified in a payment platform such as transferring a coin, sending an ACK, transferring a transaction related details, resetting the shared transaction-related details and dropping an invalid transaction. Therefore five main types of protocol designs are identified as essential (D.Roio, et al., 2015). The sender, receiver, time stamp and the particular digital signature are included in all five protocol designs as they are mandatory in order to verify a transaction. The designs differ by the set of integrated parameters and by the flag which signifies the exact functionality.

The 'SHARE' protocol is designed primarily to use in broadcasting/ multicasting purpose and for sending a coin/ transaction request. It consists of the following set of parameters as illustrated in following *Figure 3.3.1 Protocol Design: 'SHARE' Protocol*. *S_ID* and *S_PARA* denoted the service id and service specific further details such as a carpooling ride distance or a shopping bill id respectively. *S_LOCATION* represents the coin miner's location. *PROP_VALUE* stores the probability value associated with the trusted network concept which is a fixed value of 75% for this proposed verification model. The stated *PUB_KEY* denotes the public key and is designed only to use at the nodes registration.

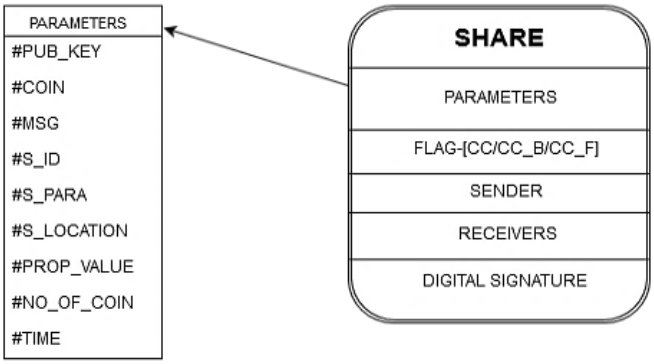

Figure 3.3.1 Protocol Design: 'SHARE' Protocol

The protocol design of 'PUT' protocol consists of the coin along with a set of coin related flags such as *CC – Coin Creation*, *CC_F – Coin Creation Failure*, *CC_F_B – Coin Creation Failure Block* and *CT – Coin Transfer*. This is designed to transfer a coin when a request receives or can be used to send ACKs as a response to a shared transaction.

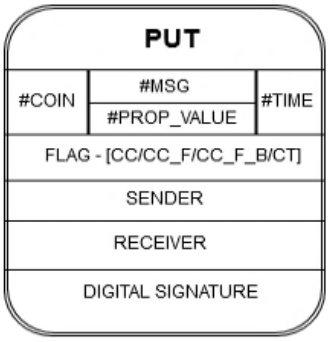

Figure 3.3.2 Protocol Design: 'PUT Protocol

The 'DATA' protocol is designed to send a coin without a coin request from the other peer or to send transaction details as a type of an ACK. It consists of three different types of flags as *CC_ ACK – Coin Creation Acknowledgment*, *CC_F_ACK – Coin Creation Failure Acknowledgment* and *B_CT_ACK – Coin Transaction Block Acknowledgment*.

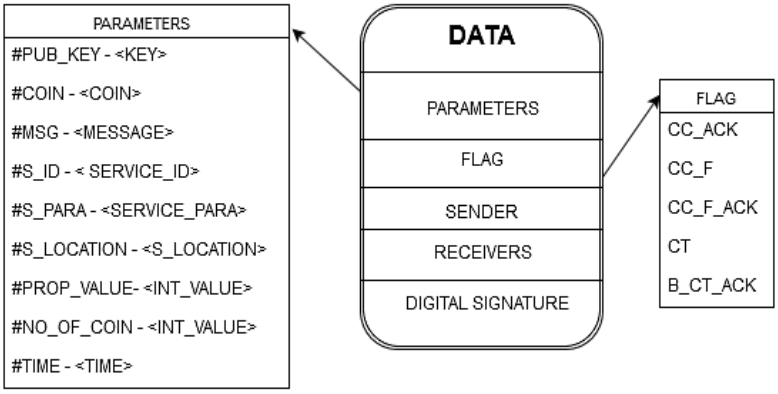

Figure 3.3.3 Protocol Design: 'DATA' Protocol

The 'DELETE' protocol is designed to delete a particular coin in any transaction verification failure or a coin verification failure. The unique flag named *CD* of it is designed to denote *Coin Deletion* functionality. 'UNSHARE' protocol is designed to unbind the previously shared parameter/ attribute values in order to maintain a proper consistency and atomicity of transactions. It is essential to avoid any unauthorized parties getting access or reusing the shared transaction related data.

3.4 Transaction verification scenarios

The major components involved in transactions of such payment platform are Miners and Application nodes as identified by the analysis of target system architecture (See Appendix A – SCPP Common Payment Platform). All the transactions are categorized under three main scenarios based on the involved parties in each transaction and are differentiated in the following *Figure 3.4.1 Transactions Overview in a Common Payment Platform*.

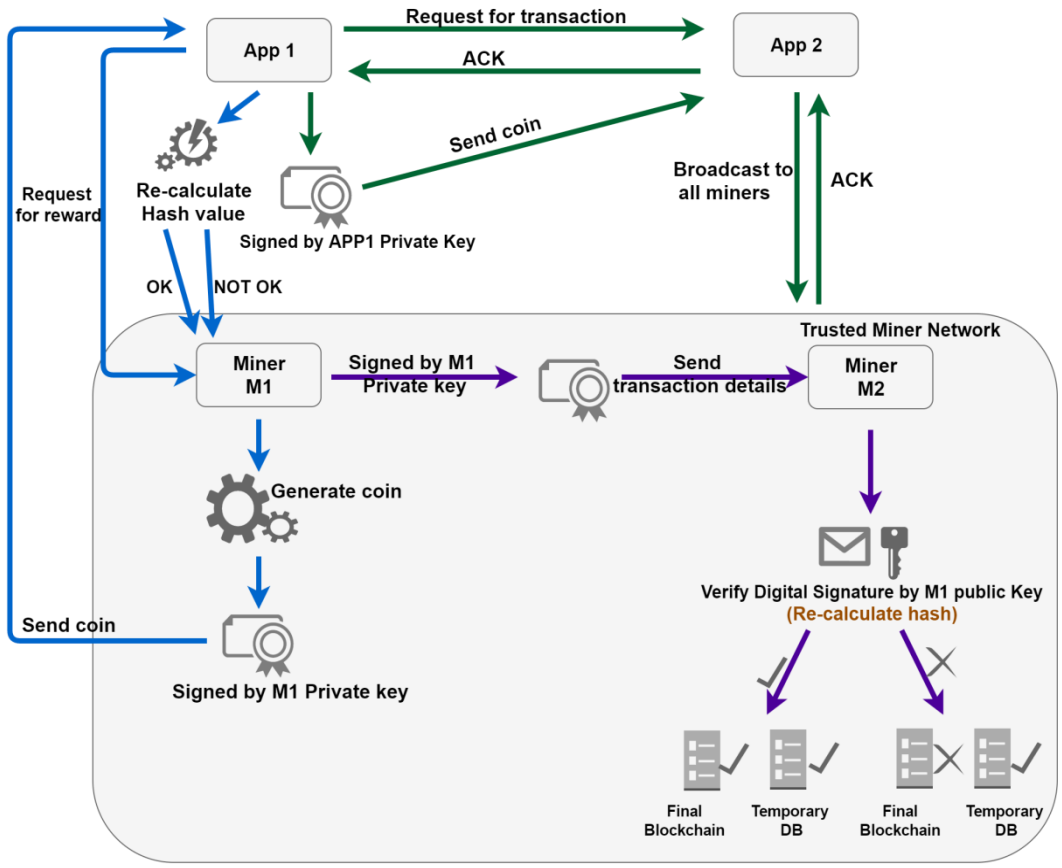

Figure 3.4.1 Transactions Overview in a Common Payment Platform

3.4.1 Scenario a: miner and application transaction verification model

This scenario is regarding transactions in between service-oriented currency miners and registered users who have the ability to earn/ spend currencies.

A.1. User earns coins when retrieves a particular reward involved service. A miner sends a coin as a reward to the user at the moment a particular service is served to that user. It is identified as a Miner → Application (User) Transaction.

A.2. User can spend the earned coins in return when paying for a particular platform registered service. It is not required being the same service since a common payment platform. A user application sends a coin as the/part of payment to the service holder's miner. It is identified as an Application (User) → Miner Transaction.

MitM attack is identified as one of the main possible threats in this scenario. An attacker can either obtain the transferring coins or can alter the acknowledgment messages. Therefore it is crucial to protect over MitM attack at this stage. The design solution for this scenario is illustrated as in below *Figure 3.4.2 Scenario A: Miner-Application Transaction Verification Model.*

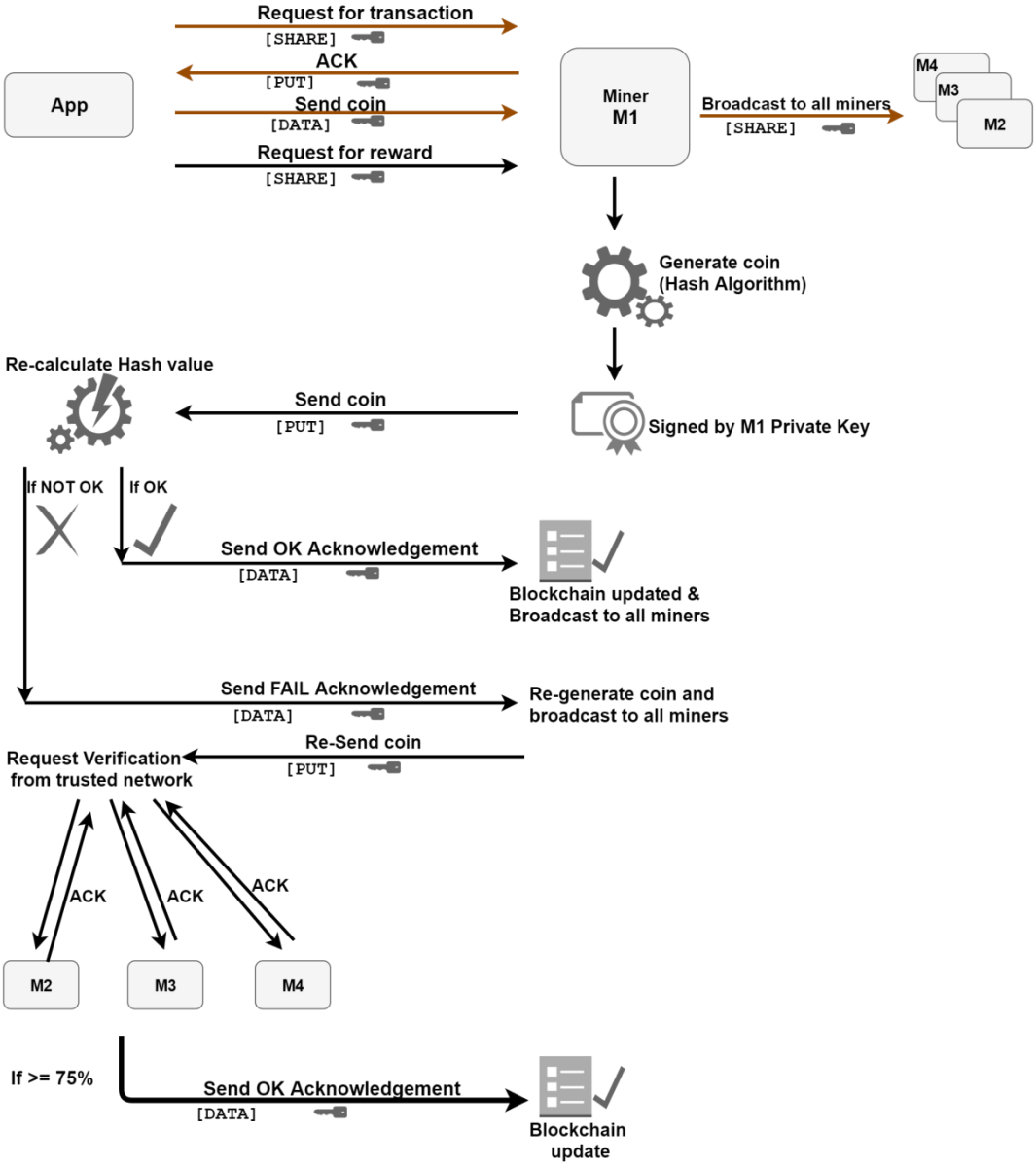

Figure 3.4.2 Scenario A: Miner-Application Transaction Verification Model

Therefore in this scenario A.1 above: It is designed as a user bundles the particular service related details into the protocol named 'SHARE'. At the same moment input the details into a one-way hash function and retrieve a hash value of the details. And input the hash value along with the application's private key into an asymmetric encryption algorithm and retrieves the digital signature. Attach the digital signature into the protocol and send the coin request to the particular miner. A specific time constraint is defined as a waiting time for the application node to wait till a response returns. At the other end when a particular miner retrieves the request, the miner checks the protocol's sender and receiver details and primarily verify whether it is an accurate delivery or not. In case not, it is designed to drop the coin request packet. Else it subsequently decrypts the containing signature using the relevant public key of the sender and inserts the protocol containing details into a pre-configured one-way hash function to re-calculate the hash value. If the re-calculated hash value and

decrypted signature values are unequal, the request packet gets dropped. Else the request gets completely verified successfully and move into coin mining process which is not in the scope of this thesis. As per the primary conceptual facts in this designed model; the digitally signing is mandatory whenever issuing anything to the p2p network. Therefore the response packet containing the coin is required to be digitally signed using the same procedure as discussed and sends back to the application node via the protocol named 'PUT'. If the application node does not retrieve the response coin within the defined time constraint the transaction gets completely canceled. Else the application node verifies the retrieved protocol packet by decrypting and re-calculating hash.

If the verification is failed: the application node should send an ACK back to miner notifying the failure by digitally signing. A MitM attack or a network failure is identified as the causes for such failure. Then the miner retrieves the ACK about the failure and verifies the ACK. At this stage, the trusted network concept is designed to get involved as a second layer of a particular transaction verification mechanism. The miner drops the previous not-verified coin, re-generates a coin and multicast it to all miners in the payment network. As all the nodes are maintaining the public keys of everyone; all the miners who retrieve that coin does the coin verification process. If the result of that coin verification process is positive, each positive miner sends a digitally signed positive ACK to the coin requested application user notifying to accept that coin. And those miners keep a record of it as a verified transaction in their blockchain. At this stage, a probability level is defined in this design as a fixed 75%. Therefore the application user accepts the coin and adds to the digital wallet only if more than or equal 75% of positive ACKs are received within the defined time constraint. And concurrently the application node sends a digitally signed positive ACK to the coin generated origin miner. Once the origin miner retrieves that ACK and verified, it updates its own blockchain by recording the transaction as a verified transaction.

If the application node could primarily verify the retrieved protocol packet containing the coin by decrypting and re-calculating the hash by itself; the application node adds the coin to own digital wallet and concurrently sends a digitally signed positive ACK to the origin miner. If the origin miner retrieves the positive ACK within the defined time constraint: the origin miner verifies the ACK, concurrently updates own blockchain with a new record of a verified transaction and send the verified transaction details to all the miners in the trusted network. As it is about a newly generated coin; the other miners does a complete coin verification process. If it is verified, that particular miners update their own blockchain with a record of a verified transaction. Else drops the coin and updates the blockchain with a record of a non-verified transaction.

Furthermore, if the origin miner does not retrieve the positive ACK within the defined time constraint: The origin miner drops the coin and updates the own blockchain with a record of a non-verified transaction.

In the scenario A.2 above: Prior to sending the coin it is identified as essential to check the availability of the miner at the other end. Therefore an availability checking is designed by sending a digitally signed 'SHARE' protocol packet to the miner at the receiving end. If the miner is offline the transaction gets dropped

since only the online transactions are addressed in this research work. Else if the miner is available online and is ready to accept the coin; it sends a digitally signed ACK back to the user application. If the user receives that ACK within the defined time constraint, the user sends the coin integrated into a 'DATA' protocol using the same conceptual fact of digitally signing. At this stage, the coin is not completely deducted from the digital wallet of the user until the miner verifies the coin and transaction. At the miner's end, it verifies the packet primarily and an additional coin verification process is also designed. If either of them does not verify, the transaction is dropped and a failure ACK is sent back to the user application. The coin remains in the same user wallet non-altered. Else if the miner completely verifies the user's packet along with the coin; the miner updates own blockchain with a record of succeeded transaction and multicast the transaction details to all miners and to the relevant user. All the other miners simply verify the packet and directly update their own blockchain by trusting the sending miner (K.Croman, et al., 2015). Subsequently, once the user receives and verifies the positive ACK sent by the miner, the coin permanently gets deducted from the user application digital wallet. The transaction is rollbacked and the coin is restored to the user wallet if a collision is detected by the involved miner within the next t seconds in case of a double spending attempt. The time constraint t vary upon the p2p network conditions.

3.4.2 Scenario b: miner to miner transaction verification model

This scenario is regarding transactions among service-oriented currency miners. As described above sections (See section 3.4.1 above); all miners are considered as in a trusted network.

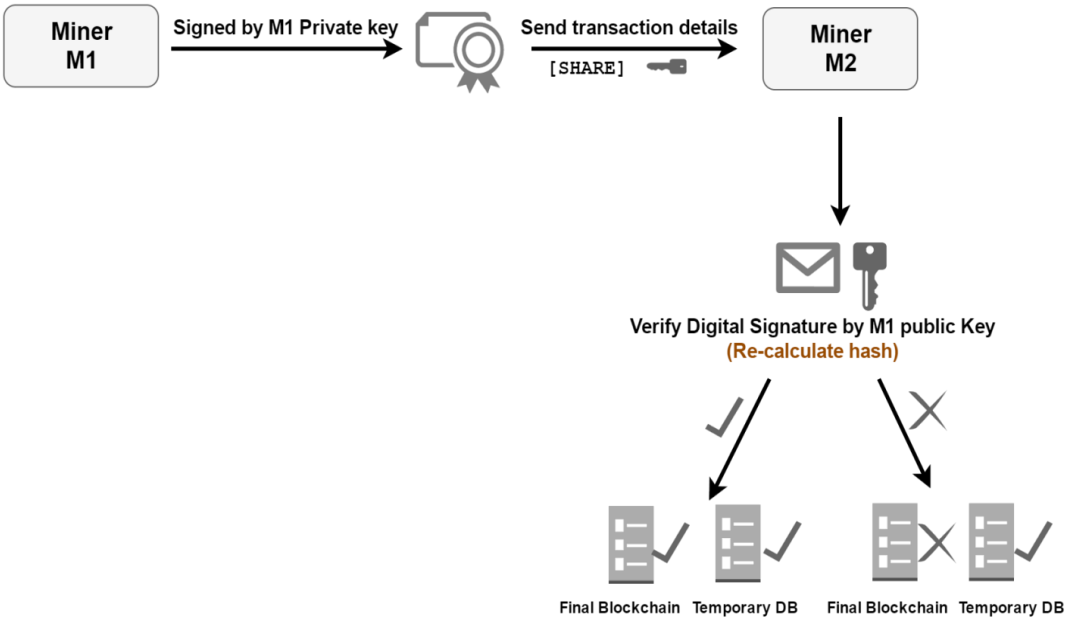

Figure 3.4.3 Scenario B: Miner-Miner Transaction Verification Model

Two possible requirements are identified to initiate a p2p transaction among service-oriented miners.

B.1. A user fails to verify a newly generated coin sent by a miner and sends a negative ACK to the origin miner. The origin miner re-generates a coin and multicast it to the user and to all the miners. Considering a single M to M p2p transaction; a miner sends the coin integrated into a digitally signed 'PUT' protocol. If a particular miner does not receive the packet it is not considered as critical since it is a multicast and there exist a considerable number of miners. The receiving miner primarily does the packet verification using previously described digital signature verification process (See section 3.4.1 above). Subsequently, it further does the coin verification since no records are available in the blockchain as it is a newly generated coin. The packet along with the coin is dropped if any of the verification fails. Else the miner updates the own blockchain with a successfully verified transaction record. And sends a positive ACK to the coin requested application user.

B.2. A miner retrieves a coin from a user application. Miner updates the own blockchain and multicast the digitally signed transaction details to all miners via a 'SHARE' protocol. A particular retrieving miner is designed to only verify the protocol packet using digital signature. The coin verification is excluded in this scenario since the coin is not a newly generated one. Therefore if the retrieving miner receives the packet within the defined time constraint, it verifies the packet and checks the blockchain records for further ensuring whether the last owner of the particularly mentioned coin equals to the current coin spending user. If the verification is succeeded each miner updates its own blockchain as a verified transaction. Else drops the transaction and records in the blockchain as a non-verified transaction.

MitM attack is identified as the most significant possible threat in this scenario. An attacker can either obtain the transferring coins or can alter the transaction details which are designed to be sent in a form of ACKs. Therefore the described design solution is addressing the protection over the MitM attack in this scenario.

3.4.3 Scenario c: app to app transaction verification model

This scenario is regarding transactions among user applications. In this scenario the MitM attack and double spending both are identified as crucial. Therefore the design solution is considered in prohibiting a user sending the same coin to more than one application users and to secure the transaction along the network without any alteration.

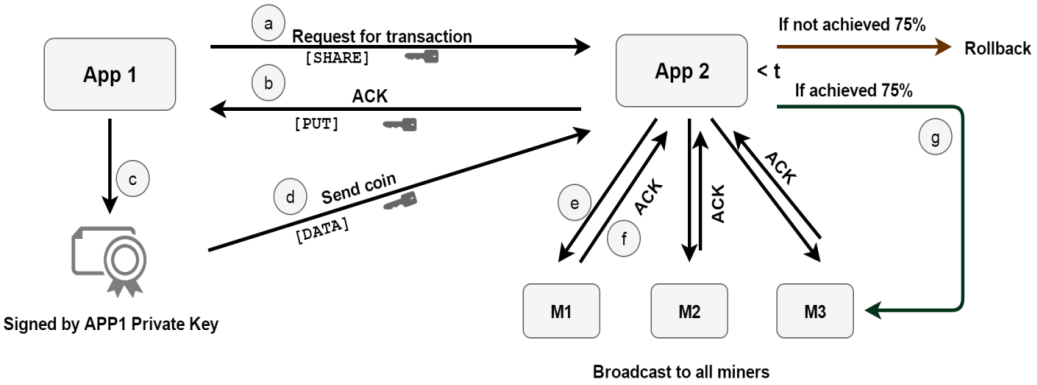

Figure 3.4.4 Scenario C:App-App Transaction Verification Model

Prior initiating a p2p transaction with another user application it is identified as essential to check the availability of the end node since only the online transactions are considered in this research work. Therefore the user application sends a transaction request in a digitally signed 'SHARE' protocol. If the receiving user is offline or the request is not verified; the transaction is designed to get canceled. Else if the receiving user is online; a positive ACK is sent back to the sender in a 'PUT' protocol using digital signature. After the verification of the ACK, the application node sends the coin via a 'DATA' protocol by digitally signing. The coin is designed not to get deducted from the sender's digital wallet until the transaction verification completes. The receiving application (App2) user verifies the packet primarily and drops the transaction in any failure. If the packet is verified the App2 again digitally sign it with own private key and multicast to the trusted network of miners by requesting to verify that coin. As discussed in section 3.4.2 above, the miners check for the details in the coin scrypt and look in their blockchain. If the last owner of that particular coin is identified as the App1 in their blockchain; the particular miner verifies the transaction and sends a positive ACK back to App2 while updating own blockchain. At this stage, the probability schema is again designed to invoke. App2 accepts the coin only if more than or equal 75% of positive ACKs are received from the trusted network of miners within the defined time constraint. If accepted; the coin adds to the receiver's application digital wallet and concurrently a positive ACK is forwarded to the sender application to deduct the coin completely from its wallet. App2 drops the transaction if App2 received a lesser percentage of verifications from the miners. The sender also designed to get rollbacked the transaction after the defined time constraint resulting the coin to remain non-altered in the sender's wallet. The double spending problem is addressed in these design steps because no more than one transaction involving the same coin can obtain a probability of 75% from the trusted network of miners.

4. IMPLEMENTATION

This chapter is intended to present the implementation architecture and details of the system that was developed, which serves as a proof-of-concept for the proposed design solution in a real-world domain. A detailed

description of the implementation architecture and its core modules is provided in this chapter while exploring the technologies and tools adopted.

4.1 Implementation assumptions and dependencies

The implementation is prioritized towards the application component consisting registration and digital wallet while the service-oriented currency miners and blockchain architecture are reused from existing modules. The identified development related assumptions and core dependencies are listed in the following *Table 4.1.1 Implementation Assumptions and Dependencies*.

Table 4.1.1 Implementation Assumptions and Dependencies

Assumptions	Dependencies
Currency miners are desktop based applications	Only the currency miners depend on blockchain architecture, not the user applications
Coin is a scrypt with consumed service details attached	The coins are generated only depending on user service consumptions
A once generated coin is vendor independent and can be used for any transaction within payment platform	A p2p network is the processing environment for all the transactions

4.2 Protocol implementation

The five protocol designs; SHARE, PUT, DATA, DELETE and UNSHARE discussed in the previous section 3.3 above are implemented in *Python*. It is selected over other programming languages due to the facilitated mathematical calculations, feasible security libraries and the majority code base of *bitcoin* protocol *BIP* is also in *Python* (Bitcoin.org, 2017). The *Python* built-in libraries and packages *TkInter*, *socket*, *threading*, *multiprocessing*, *Twisted*, *PyMongo* are supported in the development. The protocols are invoked based on the three transaction scenarios discussed in the previous DESIGN. The flags denoted in the designs of each protocol are primarily used in invoking the protocols for the exact requirement. The protocol invoking related to the Scenario A: Miner and User Transaction (See section 3.4.1 above) and Scenario B: Miner to Miner Transaction (See section 3.4.2 above) is illustrated in below *Figure 4.2.1 Miner to App and Miner to Miner Protocol Implementation*. The DATA – DELETE – UNSHARE invoking process signifies the miner to miner scenario.

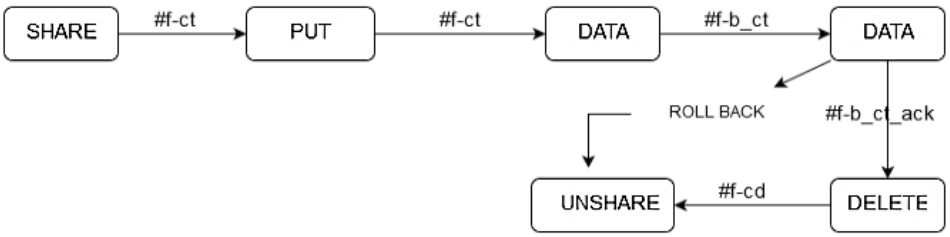

Figure 4.2.1 Miner to App and Miner to Miner Protocol Implementation

The protocol invoking procedure in between user applications are as follows based on the implementations. The rollback emphasizes canceling a particular, non-verified transaction.

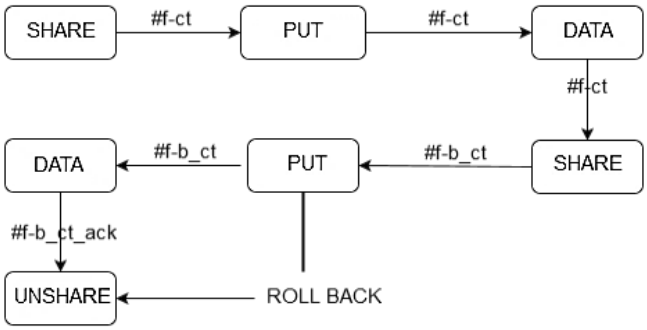

Figure 4.2.2 App to App Protocol Implementation

A single ping using a single protocol from a one member of the network to another spends 3 seconds approximately based on the implementation of the protocols and the performance of the reused switch component in establishing the p2p network. A complete transaction involving several protocols invoking costs 45 seconds approximately. Therefore the waiting time constraint t is declared as 45 seconds as discussed in the previous DESIGN. Any transaction that does not reach the defined destination within 45 seconds is canceled and discarded as a non-verified transaction unless a component retry from the beginning of that particular transaction.

4.3 Transaction verification implementation

The *SHA-256 (Secure Hashing Algorithm)* one-way hashing is involved in implementing the hashing of the protocol integrated data prior to digitally signing. *SHA-256* or above is recommended for applications where security is vital and it produces $32 - byte$ hash values (M.Stevens, 2012). Furthermore, it calculates a hash code for an input up to $2^{64} - 1$ bits and undergoes 64 rounds off hashing. Therefore the resulting hash code is expected to be a 64 digit hexadecimal value. Though *SHA-256* is considerably slower than the popular *MD5* the security is identified as more important than the performance since the digital currency is the main asset of the

payment platform (S.Aggarwal, et al., 2014). The *Python* in-built *base64* data encoding and *Crypto.Hash* sub package is supported in the implementation.

The asymmetric algorithm *RSA* is selected in the digital signature implementation in order to generate public-private key pairs for each registered component in the common payment platform. Though the symmetric algorithms are faster, the asymmetric is used since its security is powerful as long as the private key is secret none can decrypt the encrypted data (P.D.Harish, 2015). Among the asymmetric algorithms, the *RSA* is identified as the most appropriate based on the conducted background research on similar technologies. Accordingly, the *RSA* algorithm is based on the 'number theory of the ruler' which is identified as the most security system in the key systems. The sub package *Crypto.PublicKey* from the *Python Cryptography Toolkit* is used in the development.

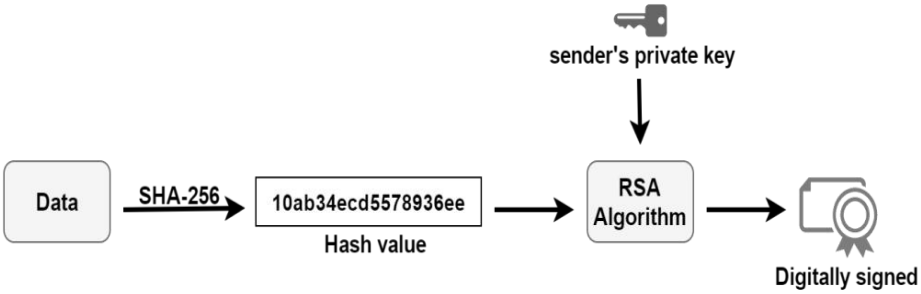

Figure 4.3.1 Asymmetric Digital Signature Generation

The public keys of all registered member nodes in the payment platform are embedded into each application and miners. As the reused miners are desktop-based, they are configured to store the public keys of all others in a secure *.key* folder structure. In the mobile based Android user applications, both private keys and public keys are stored using *SharedPreferences*. It is selected over *SQLite* since for storing key-value pairs and retrieving the data is identified to be simpler in *SharedPreferences* (IBM, 2016). When a new miner or a user application gets registered the new node's public key is embedded to all other members in the network including to the reused *Senz* switch module (GitHub:senzprojects/udp-switch, 2016). Furthermore, a *MongoDB* data structure is implemented and configured *to the reused Senz* switch for its requirement of storing the public keys. And the newly embedded miners and user applications are re-deployed as updates/ versions.

The fixed probability criterion of 75% in trusting the trusted network of miners is built-in to the Android user applications as a static value for the scope of this research work. The waiting time constraint of 45 seconds for a particular transaction to complete is also built-into the both miners using *Python* and to Android user applications using *Java*.

4.4 Proof-of-concept

The target system of common payment platform named ‘Social Currency Payment Platform (SCPP)’ is considered as the proof-of-concept for realization the proposed transaction verification model (See Appendix A – SCPP Common Payment Platform). The service-oriented currency miners and *Senz* switch module are integrated to the platform as reusing components. The Android user application component along with the digital wallet is implemented for two example service types namely a Carpooling Service Application and a Shopping Service Application. The two service types differ in the involved business logic while the security mechanisms and transaction verification model remain the same. The relevant core implementations are discussed briefly in this section and are illustrated in the following *Figure 4.4.1 Proof-of-Concept Overview*.

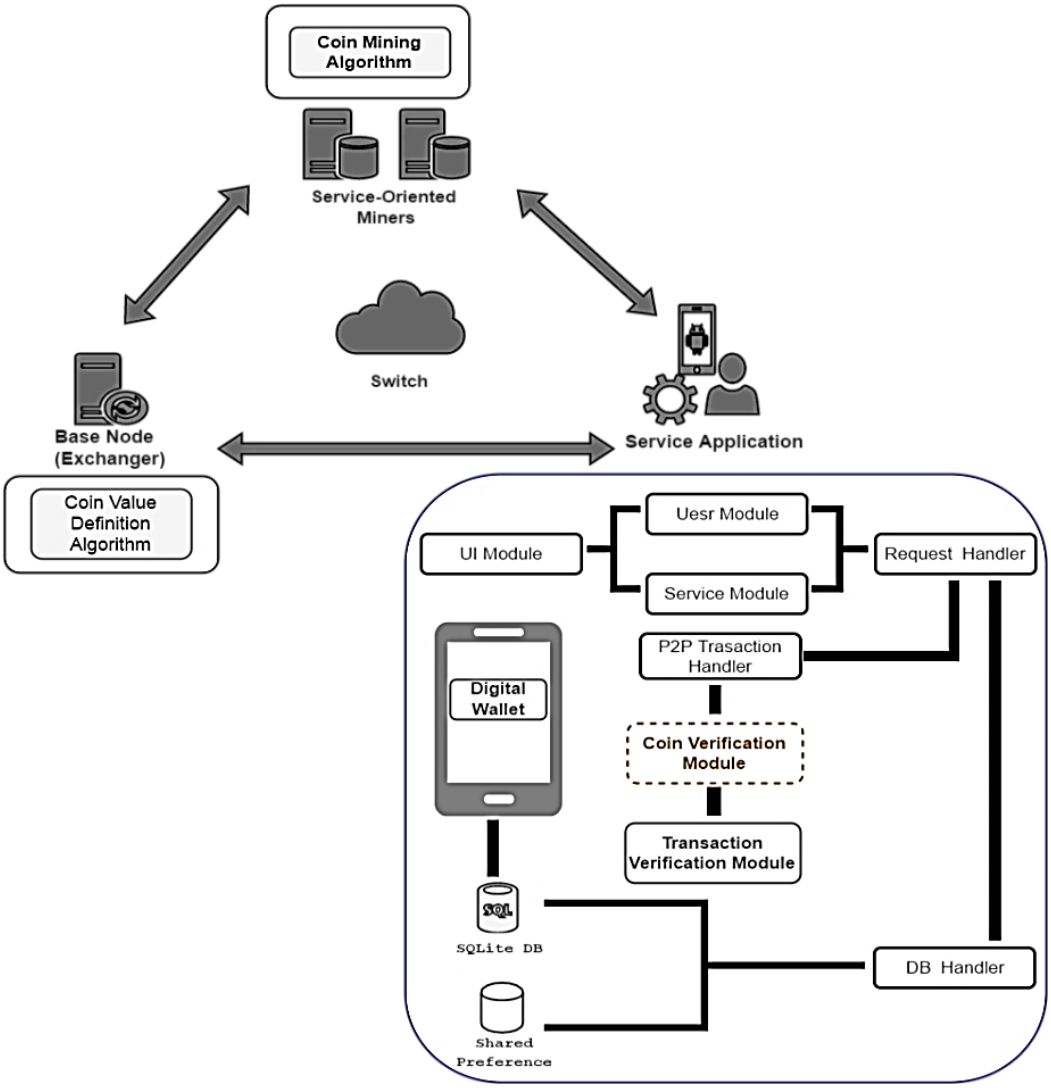

Figure 4.4.1 Proof-of-Concept Overview

4.4.1 User application registration

The registration process is implemented in providing multiple user registrations per single device. The username and password are managed in the application with *SharedPreferences*. A pair of the public-private key is generated in each successful user registration as discussed in section 4.3 above. A different pair of keys is generated when multiple user registrations occur and all key pairs are separately managed in the *SharedPreferences*. A copy of each public key is bind to each miner and application as a new update of each of them. A copy of the each public key is transferred to the plugged *Senz* switch also as its configurations request to store them. It is not a dependency for the proposed transaction verification model whether the public keys are stored in the network switch(s) or not.

4.4.2 Application digital wallet

The digital wallet is a core module in the user application and is implemented using the *SQLite* database supported in the Android development. The retrieving coins are stored in the digital wallet only after the completion of a verified transaction. And coins are deducted from the wallet only after the receiving party successfully accepted the coin with the completion of transaction verification as discussed in DESIGN. Once a coin is sent for a particular miner or to another user, the record of the coin is developed to become inactivated until the receiving node either accept or reject the coin. The coin record is developed to become active if the receiver rejected the coin or if the transaction is failed. The inactive coin record is completely removed once the receiver has been accepted it. The double spending is primarily addressed in the front-end level through that implementation. Regarding the implementation, the coin is considered as a scrypt file with necessary details though the format of the coin is not a dependency for the digital wallet but for the way of storing. The coin being a scrypt file solely storing in *SQLite* is not identified as efficient. Therefore the external storage space of the particular user device is involved by saving the coin scrypt and encrypting with the user public key. The storage path of a particular coin scrypt is maintained in an *SQLite* database as raw data. In accessing the stored coins for transactions the dedicated external storage space is decrypted by the user private key. Furthermore, it is identified an attacker cannot regenerate a coin with alteration of stored coin scrypts or paths due to the strength of the proposed verification model unless the user device is stolen at the worst case. An abstract insight into the described implementation is provided in the below two foremost figures.

Figure 4.4.2 User Application Home

Figure 4.4.2 User Application Digital Wallet

5. PROJECT EVALUATION

The previous IMPLEMENTATION presented the detailed description of the implementation modules of the proposed transaction verification model. It is expected to present the formulation details of the evaluation criteria and its experimentation results for that implementation in this chapter. The evaluation on the security level of the implemented verification model over the Man-in-the-Middle attack and Double Spending is discussed respectively.

5.1 Transaction verification over man-in-the-middle attack

The implemented transaction verification model is based on the foundation of the blockchain architecture consists of two abstract levels of verification.

1. The primary verification concepts applied are the *RSA* asymmetric digital signature mechanism along with *SHA-256* one-way hashing.
2. The other verification concept applied is the transaction verification via the trusted network of currency miners with an acceptance probability level of 75% and a time constraint of 45 seconds; a transaction is verified if and only if more than 75% of miners have verified the transaction within 45 seconds.

5.1.1 Evaluation methodology

The evaluation is conducted by measuring the possibility of verifying a transaction while attacking currency miner(s) in the payment platform which is the core component in the implemented trusted network of miners. As presented in section 4 above, the assumptions are set without alteration. Therefore the metrics and the measures involved are as follows.

- The defined maximum time constraint for a transaction to get verified from one peer to another is 45 seconds
- The defined probability level of accepting a trusted network verified transaction is 75%; ¾ of miners should have verified the particular transaction in order to be accepted as verified

The total number of miners in the trusted network, the number of attacked miners and the number of miners that have successfully verified the particular transaction among the non-attacked miners are varied in the evaluation. Though the average time based on the implementation for a single protocol ping in the developed p2p network is 3 seconds, it is considered as a varying parameter for the evaluation purpose.

Table 5.1.1 Man-in-the-Middle Attack Evaluation: Varying Parameters

Total number of miners in trusted network	Total_M
Number of attacked unhealthy miners	Attack_M
Number of healthy miners that have successfully verified a given transaction	Verified_M
The average time for a single protocol ping in the p2p network	Ping_T

The 'Transaction Probability' is measured as $\frac{\text{Verified\_M}}{\text{Total\_M}}$. And the target is achieving 75% of positive probability within 45 seconds according to the proposed and implemented verification model. Accordingly, the evaluation process is conducted in *Python* by varying the above-mentioned parameters in the *Table 5.1.1 Man-in-the-Middle Attack Evaluation: Varying Parameters*.

5.1.2 Evaluation results and analysis

According to the obtained evaluation results, the transaction completely gets non-verified when more than ¼ of the total number of miners in the trusted network of miners becomes unhealthy due to an attack. Therefore the transactions remain secure since the non-verified transactions are dropped and all the shared parameter details are also discarded using the UNSHARE protocol by recording in the relevant blockchain as log data. The obtained significant results are as plotted in the following *Figure 5.1.1 Man-in-the-Middle Attack Evaluation Results.*

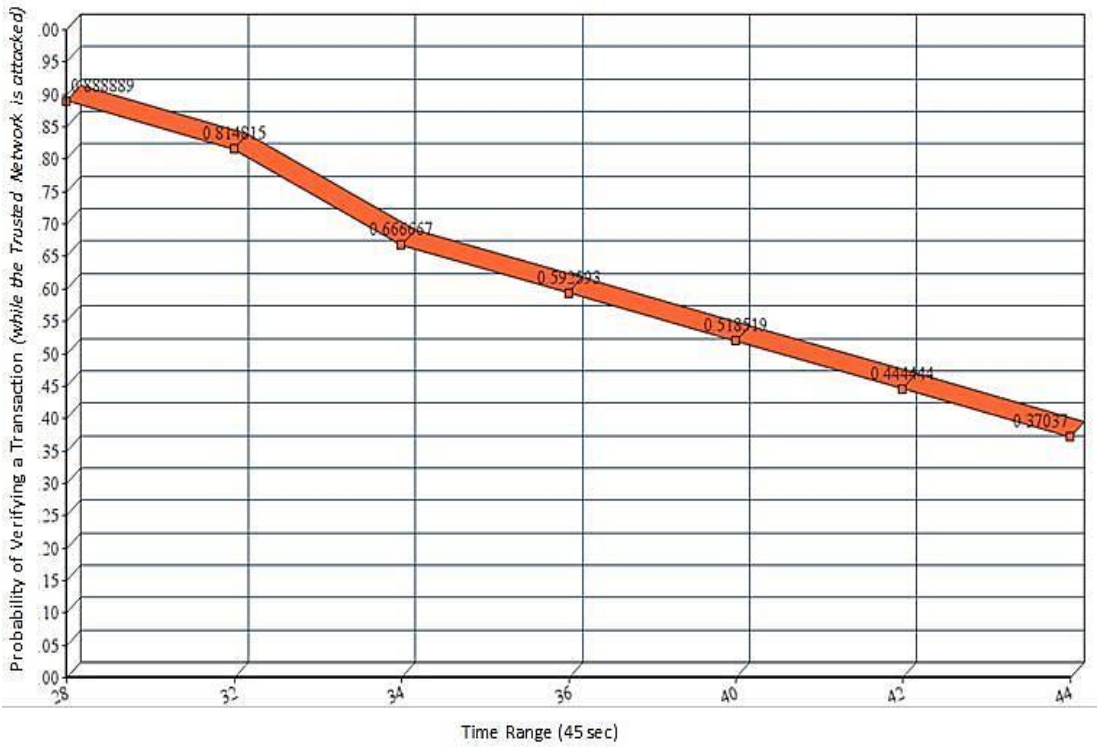

Figure 5.1.1 Man-in-the-Middle Attack Evaluation Results

And when more than ¼ of the miners in the trusted network of miners become unhealthy the probability of a particular transaction getting verified is decreased over the time as evidenced in the above *Figure 5.1.1 Man-in-the-Middle Attack Evaluation Results*. Since the time constraint is defined as 45 seconds the probability continuously gets decreased towards the time limit of 45 seconds. The probability is linearly decreased since the average time for a single protocol ping in the p2p network (*Ping_T*) is considered as a stable static value. But it is identified the linearity can be altered towards a non-linear result based on the stability of the network. Because the network can be either busy or downtimes are completely unavoidable. But even if the network is unstable the probability of verifying a transaction is identified to be decreasing further which is an expected positive result. It is considered as a positive result since the goal is not to mandatorily verify a transaction with a risk but to securely verify which can result in even a complete cancellation of a transaction too.

5.2 Double spending prevention

In the identified service-oriented common currency platform the double spending is related to two scenarios among the three scenarios of transactions discussed in DESIGN. The scenario of p2p transactions among a miner and a user application (See section 3.4.1 above) and transactions among user applications (See section 3.4.3 above) are the situations relevant in spending a particular same coin more than a once. The situation of a miner issuing the same coin to more than a single user is identified as an example sub-scenario under scenario A

(See section 3.4.1 above). But it is justifiable as the miners are the reputed vendors in the market and the concept of trusted network is established among all the miners. Furthermore, each miner is observed by all the other miners in the network by prohibiting a particular miner to act bogusly. Therefore a user trying to spend a single coin on multiple miners or multiple users are the identified possibilities for double spending. But once a coin is spent on a particular purpose, that coin is implemented to become inactive until the receiver either accept or reject it according to the implementation of the digital wallet. But as the wallet storage is associated with the user's mobile device storage a risk is identified that an intelligent user could replicate fake duplicates of a coin that would get visible in the wallet.

5.2.1 Evaluation methodology

As discussed in DESIGN and IMPLEMENTATION, the conceptual solution of trusted network of miners with 75% of probability level is focused on the challenge of double spending prevention (M.Lei, 2015). The trusted network of miners relies on the adapted blockchain architecture which is supposed to maintain the history of transactions (K.Croman, et al., 2015). Therefore as each miner is associated with a blockchain, a miner is identified to have the capability of verifying a particular coin by its own without any supportive transactions involved (See Appendix B – Coin Wise Blockchain Architecture).

In considering the scenario A (See section 3.4.1 above): transactions among miner and user application; two coin spending requests on a same single coin instance are triggered to two different miners (vendors) by involving an implemented user application.

Regarding the scenario C (See section 3.4.3 above): transactions among user applications; involving three instances of the implemented Android user applications, one user application is triggered to send two instances of a coin sending requests (two ACKs to check for the availability of the two destination nodes) for a same single coin. It is ensured the all three involved user applications are made available online continuously.

5.2.2 Evaluation results and analysis

In the scenario A (See section 3.4.1 above), the two miners received the coin spending requests from the user and primarily verified the user authenticity independently. As the user is a registered node in the payment platform the both miners sent positive ACKs to the user by notifying to send the coin for spending. Once the user application sent the same coin to two miners, both miners verified the coin ownership by checking the blockchain history at the back-end. Therefore both miners separately updated their own blockchain while multicasting the transaction details to the trusted network of miners. But in the feedback, both involved miners received ACKs of indicating duplicate transaction details for the same coin resulting a collision. As a result of

the identified collision both miners again rollbacked the transaction and broadcasted a negative ACK to the coin sender application and trusted network of miners. Therefore both spending attempts are canceled and duplicate coins get restored to the user wallet. Unless the collision is not detected within 45 seconds a single spending succeeds in the first come first serve basis depending on the stability and the performance of the p2p network conditions. Therefore either a user spend the coin for a service as a payment or exchange the coin into fiat currency via a miner, only a single instance of the same coin is allowed in spending.

In applying the evaluation methodology on the scenario C (See section 3.4.3 above): transactions among user applications are illustrated in the following *Figure 5.2.1 Double Spending Evaluation.*

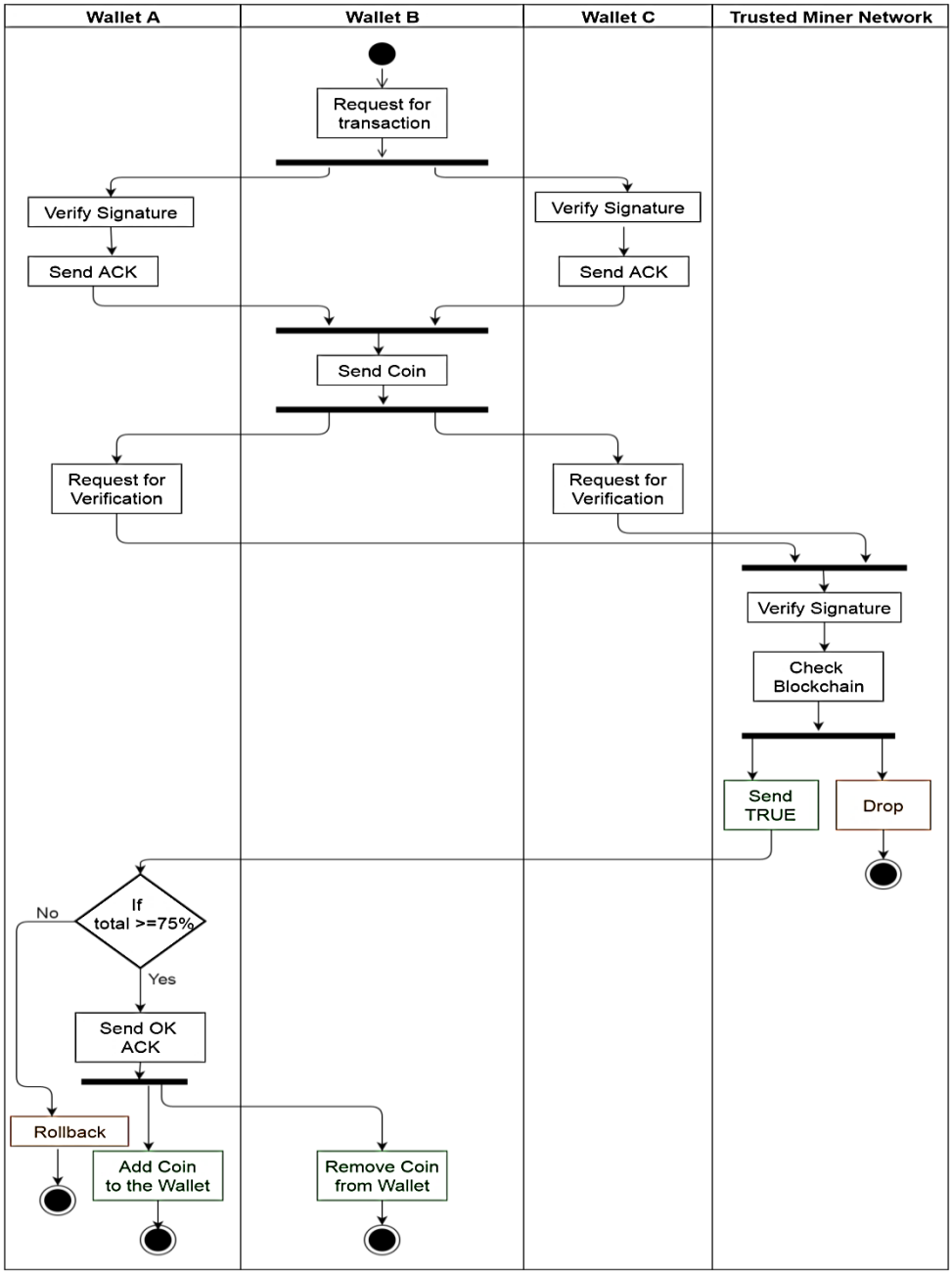

Figure 5.2.1 Double Spending Evaluation

The coin sending request is accepted by the other two online user applications and a positive ACK is sent back to the sender node indicating to send the coin. Thereafter the coin is sent from the sender's wallet to both users. According to the implemented verification model the two receiving users multicasted the coin to the trusted network of miners for verification and waited 45 seconds for miners feedbacks. But each miner only verifies a single coin verification request and drops the other duplicate request detailed about the same coin. As the result of that one waiting user received 65% of positive feedbacks from the trusted network of miners while the other user received a 35% of positive ACKs. But as 65% and 35% both are less than the defined 75% of probability level in the verification model, both the coin spending transactions are canceled and rollbacked. It is

identified that the p2p network quality is a core dependency in the trusted network and probability based verification level. As a result when repeated the same evaluation methodology the probability levels were altered to as 78% and 22% respectively. Therefore the user who received 78% of positive ACKs successfully accepted the coin and coin is added to the wallet of that user application while the other lesser probability transaction gets canceled by avoiding the double spending.

5.3 Discussion

The transaction verification probability over the Man-in-the-Middle attack is evaluated with a statistical and a mathematical methodology. It is identified that the probability of a transaction getting verified is decreased when the number of unhealthy nodes in the trusted network of miners is increased. Therefore either a transaction is completely verified or rollbacked in a malicious environment by maintaining the atomicity. The double spending possibility is evaluated in a scenario-based methodology via the implementation. It is identified that either a single transaction is allowed or all bogus spending attempts are rollbacked with a dependency of the peer-to-peer network condition. Furthermore, eavesdropping is weakened by eliminating the ability to regenerate or infer information via limiting all the transaction involved data integrated into the designed protocols.

6. CONCLUSION

6.1 Summary

This study is focused on ensuring the security of digital currency involved transactions in a service-oriented common payment platform with the presence of blockchain architecture. Catering to the limitations of the existing complex approaches and defining a unique transaction verification model to enhance security and feasibility were considered the objectives in achieving the said goal.

In assessing the requirement, current inconveniences and uncertainties experienced by users were also taken into consideration along with the literature reviews. In achieving the targeted level of security the proposed model of transaction verification was designed in three major scenarios by capturing all necessary forms of transactions required to be performed in a payment platform among users and currency miners. In the design, the Man-in-the-Middle attack and Double Spending security issues were addressed with higher priority as specified in the scope of the thesis. Anonymity, confidentiality, integrity, non-repudiation and availability are also ensured in the presented model with the aid of designed protocols to strengthen the level of transactions verification.

The verification model was implemented using hashing, digital signatures, acknowledgments along with five major protocol developments. *SHA-256* one-way hashing with asymmetric *RSA* algorithm was involved in digital signature implementations in order to prevent MitM attack. A trust network among the currency miners was implemented based on the blockchain architecture by letting miners to verify and track the transactions. Achieving a probability level of 75% from the trusted network was set as a verification constraint in the model as a major solution to prevent the double spending. *RSA* asymmetric key generation was applied and the key management at user side was implemented to be stored in a separate *SharedPreferences* data structure instead of the digital wallet of user applications to secure the users' endpoints.

The evaluation was done in all design scenarios via proof-of-concept common payment platform called 'Social Currency Payment Platform (SCPP)'. The transaction verification levels against MitM attack and double spending were evaluated with more priority. An evaluation algorithm was implemented in order to prove the strength against MitM attack. Double spending attack avoidance was justified using the introduced probability level criterion model on top of the implemented trusted network. Accordingly, the presented evaluation results justify the strength of the implemented transaction verification model for a service-oriented common payment platform based on the blockchain architecture.

6.2 Extension work

Considering different aspects and possibilities there are several future directions that can be suggested for the work of this thesis.

Defining a dynamic criterion algorithm to control the excessive transaction verification overload in the trusted network when the number of miners gets increased: In this presented research work, it is not enhanced to a scenario of an excessive number of miners within the provided scope though it is an important possibility for a research. The performance of transaction verification would decrease when the number of miners rapidly increases if remain with the presented static 75% probability criterion. Because it would consume a considerable time delay when waiting for the verification from a large number of miners in the trusted miner network. Therefore a dynamic criterion algorithm can be a solution where the probability required in verifying a transaction from the trusted network gets fluctuated.

A reputation-based model for building the trusted network of currency miners to optimize the performance in transaction verification: The implemented transaction verification model is designed in a way that all the registered miners of the payment platform are by default a member of the trusted network among miners. Therefore the transaction verification requests get broadcasted to all miners in p2p transaction verification scenario or in a collision occurrence at the first attempt. But it would be not feasible for the performance of transaction verification when the number of registered miners get increased. Therefore it can be identified as a possible aspect of research if a filtering mechanism could be applied for all the miners in a way

only a specific number of miners are provided the privilege to be a part of the trusted network. A reputation-based model would be a one such example filter where the reputation and the number of privileged miners could be calculated on a feasible algorithm in order to maintain a high performance.

A peer-to-peer transactions verification model for offline transactions: Only the online transaction verification is considered in the presented transaction verification model with a possible enhancement of improving the model to support offline transactions. An innovative model is preferred where the trusted network would verify the transaction and notify the user once the user becomes available online. It would be an interesting research aspect since there would be many problematic scenarios to identify.

Further, as the presented research work is involved with the general public users and a digital form of currency it holds possibilities on continuing the research in Economic aspects, Social Sciences or in Ethical aspects. Therefore it is expected that the extension works would always be for the betterment.

References

A.Upadhayaya. (2012). Electronic Commerce and E-wallet. *International Journal of Recent Research and Review, I*.

B.Pretre. (2005). *Attacks on Peer-to-Peer Networks.* Zurich.

Bellovin, S. (2013). *Security Problems in the TCP/IP Protocol Suite.* New Jersey.

Bitcoin.org. (2017, January 04). *GitHub-Bitcoin*. (GitHub.Inc) Retrieved May 24, 2016, from https://github.com/bitcoin

BLOCKCHAIN info. (2017, January 02). Retrieved November 23, 2016, from https://blockchain.info/charts

D.Arthur, J., & A.Bazaz. (2007). Towards a Taxonomy of Vulnerabilities. *2014 47th Hawaii International Conference on System Sciences*.

D.Basin, et al. (2014). *Improving the Security of Cryptographic Protocol Standards.*

D.Roio, et al. (2015). *Design of Social Digital Currency.*

E.Nordstrom. (2015). *Personal Clouds: Concedo.*

Eriksson, M. (2004). *An Example of a Man-in-the-middle Attack Against Server Authenticated SSL-sessions.* Stockholm, Sweden.

G.Karame, et al. (2015). *Misbehavior in Bitcoin: A Study of Double-Spending and Accountability.* ACM Transactions on Information and System Security.

G.O.Karame, et al. (2012). *Two Bitcoins at the Price of One? Double-Spending Attacks on.* Heidelberg, Germany.

GitHub:senzprojects/udp-switch. (2016, August 20). Retrieved April 30, 2016, from https://github.com/senzprojects/udp-switch

H.B.Shadab. (2014). *Regulating Bitcoin and Block Chain Derivatives.* New York.

IBM. (2016). *Learn How to Choose the Right Database for the Job.* IBM.

I.Eyal, et al. (2016). *Bitcoin-NG: A Scalable Blockchain Protocol.*

J.Wells, et al. (2008). Enhanced Security for Preventing Man-in-the Middle Attacks in Authentication, Data Entry and Transaction Verification. *Australian Information Security Management.* Perth, Western Australia.

K.Croman, et al. (2015). *On Scaling Decentralized Blockchains.*

Katina, M. (2009). In *Innovative Automatic Identification and Location-Based Services* (pp. 25-233). IGI Global.

M.Andrychowicz. (2015). *Multiparty Computation Protocols Based on Cryptocurrencies.*

M.Bawa, et al. (2007). *Peer-to-Peer Research at Stanford.* Stanford.

M.J.Casey, & P.Vigna. (2015, January 23). *Bitcoin and the Digital-Currency Revolution*. (THE WALL STREET JOURNAL) Retrieved April 2, 2016, from http://www.wsj.com/articles/the-revolutionary-power-of-digital-currency-1422035061

M.Karpinskyy, & Y.Kinakh. (2003). RELIABILITY OF RSA ALGORITHM AND ITS COMPUTATIONAL COMPLEXITY. *International Scientific Journal of Computing, II*(3).

M.Lei. (2015). *Exploiting Bitcoin's Topology for Double-spend Attacks.* Zurich.

M.Roland. (2013). *Applying recent secure element relay attack scenarios to the real world:Google Wallet Relay Attack.* Hagenberg, Austria.

M.Srivatsa, et al. (2009). *IBM Research:Privacy in VoIP Networks: A k-Anonymity Approach.* Retrieved June 5, 2016, from http://domino.watson.ibm.com/library/CyberDig.nsf/1e4115aea78b6e7c85256b360066f0d4/e8c17563bd14c48685257576005c4fb5!OpenDocument&Highlight=0,peer,to,peer

M.Stevens. (2012). *Attacks on Hash Functions and Applications.*

M.Wachs. (2015). *A Secure and Resilient Communication Infrastructure for Decentralized Networking Applications.*

N.Daswani, et al. (2000). *SWAPEROO: A Simple Wallet Architecture for Payments, Exchanges,Refunds, and Other Operations**. Stanford, CA.

N.T.Courtois, & L.Bahack. (2014). *On Subversive Miner Strategies and Block Withholding Attack in Bitcoin Digital Currency.* London, UK.

P.D.Harish. (2015). *Towards Designing Energy-Efficient Secure Hashes.* Florida.

P.K.Mishra. (2012). ANALYSIS OF MITM ATTACK IN SECURE SIMPLE PAIRING. *Journal of Global Research in Computer Science*.

P.McCorry, et al. (2016). *Refund attacks on Bitcoin's Payment Protocol.* UK.

Pentaho. (2015). *Designing Data Intensive Applications: The Big Ideas Behind Reliable and Scalable Systems.*

R.Handa, et al. (2011). *Google Wallet - A Glimpse into the future of mobile payments.*

R.P.D.T.Rajapaksha. (2015). *Secure Architecture for Distributed Micropayment System.* Colombo.

R.Savita, & U.Datta. (2015). Two Way Authentication in MITM Attack to Enhance Security of. *International Journal of Security and Its Applications, IX*, 267-273.

R.Tripathi, & S.Agrawal. (2014). Comparative Study of Symmetric and Asymmetric. *International Journal of Advance Foundation and Research in Computer (IJAFRC), I*(6).

R.Tripathi, & S.Agrawal. (2014). Critical Analysis of RSA Public Key Cryptosystem. *International Journal of Advanced Research in Computer Science and Software Engineering, IV*(7).

S.Aggarwal, et al. (2014). A review of Comparative Study of MD5 and SHA Security Algorithm. *International Journal of Computer Applications (0975 – 8887), CIV*.

S.Nakamoto. (2012). *Bitcoin: A Peer-to-Peer Electronic Cash System.* Retrieved May 2, 2016, from https://bitcoin.org/bitcoin.pdf

S.Patil, et al. (2014). Implementation of Open Core Protocol transaction Verification IP using System Verilog UVM methodology. *INTERNATIONAL JOURNAL OF SCIENTIFIC & ENGINEERING RESEARCH, V*(9), 246-251.

T.Bamert, et al. (2014). *BlueWallet: The Secure Bitcoin Wallet.*

Appendix A – SCPP Common Payment Platform

Social Currency Payment Platform (SCPP) is a hypothetical instance of a common payment platform. The goal of it is identified as to integrate different vendor based digital payment systems. It is focused on allowing the users to use the digital coins/ rewards they earn in one vendor based system in another different service related system independently. The below *Figure A. 1 Social Currency Payment Platform Architecture* illustrates the relationship between core components of the system in an abstract design.

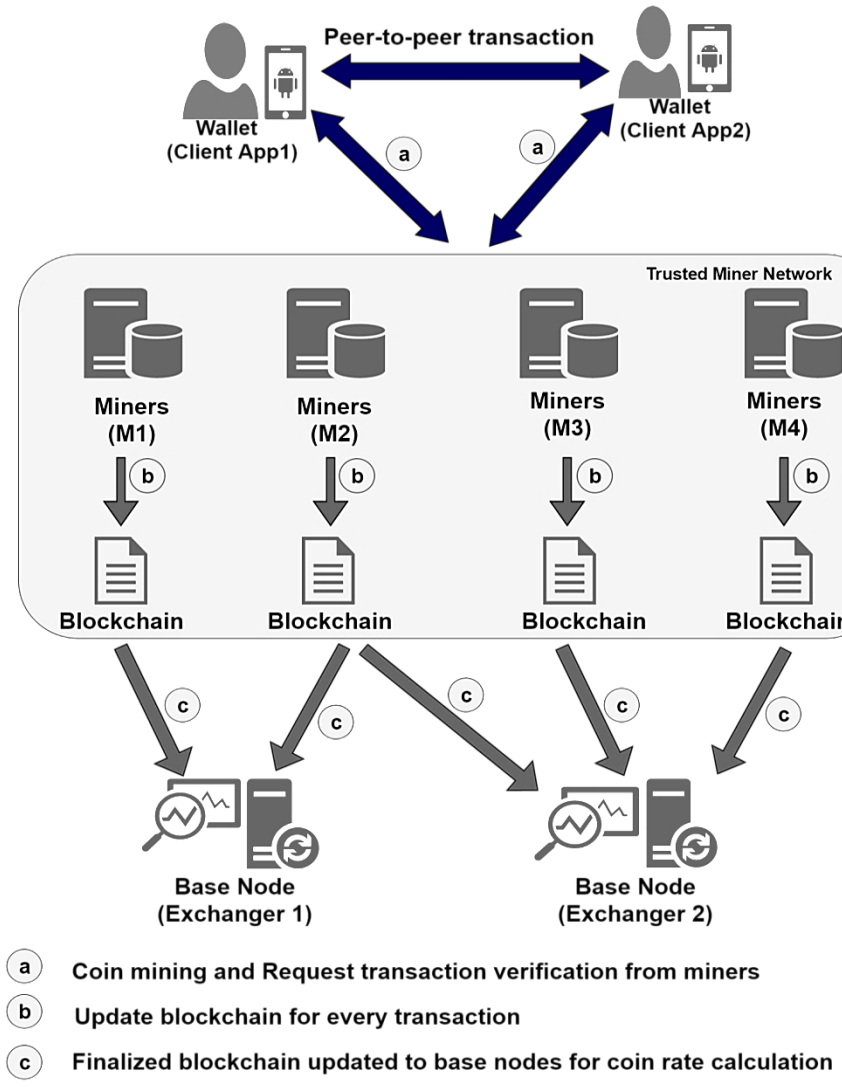

Figure A. 1 Social Currency Payment Platform Architecture

The highlighted (a) section denotes the majorly addressed transactions domain of the system in this research work.

Appendix B – Coin Wise Blockchain Architecture

The concept of blockchain architecture is a tamper-proof and a shared digital ledger that holds transaction records in a public or private peer-to-peer network. It is distributed to all nodes in the network and the ledger permanently records in blocks the history of transactions take place between the peers in the network. The below *Figure B. 1 Coin Wise Blockchain Design* illustrates a customized coin wise maintained blockchain design compatible with a digital payment platform. The transactions are recorded in each coin's identity wise in order to enhance accessing patterns. This design is identified to be useful for transaction verification purpose where coin miners in the trusted network of miners can access the history recorded in blockchain and contribute in the transaction verification process.

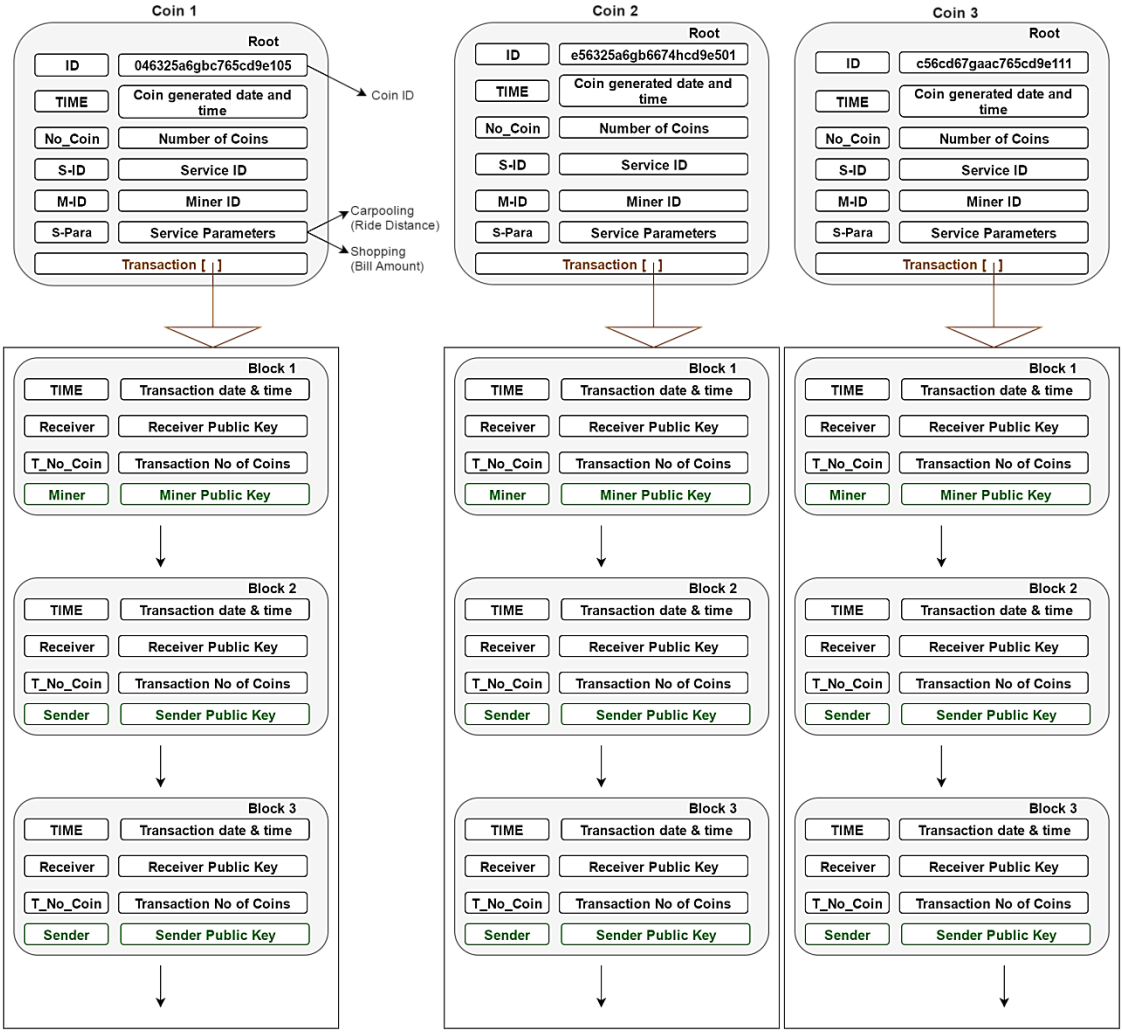

Figure B. 1 Coin Wise Blockchain Design

Appendix C – Bitcoin Blockchain Wallet Users

Among the existing digital wallets the open source *bitcoin* blockchain wallet is identified to be the world's most popular instance by contributing to over 60 million of digital transactions daily. The following *Figure C. 1 Bitcoin Wallet User Growth* reports the rapid growth of *bitcoin* wallet usage among users over last two years by depicting the demand for digital wallets. It illustrates an increasing of approximately 8 millions of users since 2015 to January 2017 (BLOCKCHAIN info, 2017).

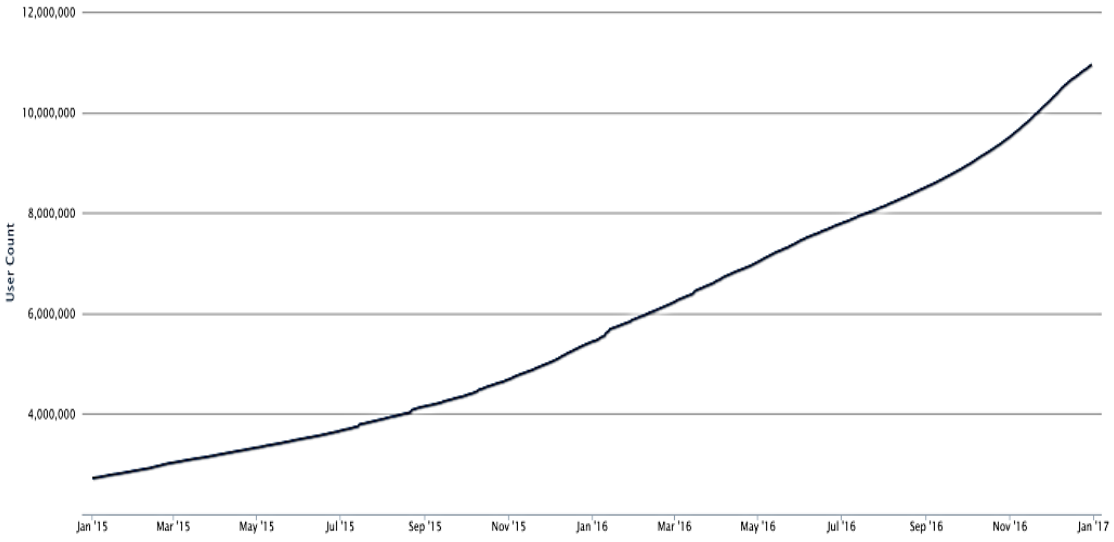

Figure C. 1 Bitcoin Wallet User Growth **(BLOCKCHAIN info, 2017)**

It is powered with capabilities as such sending and receiving *bitcoin* instantly among anyone in the world with high security and privacy including PIN protection. The hierarchical deterministic address architecture, server-side entropy for maximum randomness and the client side encryption-decryption are identified key technical aspects.