Transaction Verification Model over Double Spending for Peer-to-Peer Digital Currency Transactions based on Blockchain Architecture

Iresha Dilhani Rubasinghe
Instructor, Computer Science Dept.
University of Colombo School of Computing
Sri Lanka

T. N. K. De Zoysa, PhD
Senior Lecturer, Computer Science Dept.
University of Colombo School of Computing
Sri Lanka

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. The 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, Double Spending is identified as a key security vulnerability.

The purpose of this study is to investigate the means of addressing the said security issue 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 acknowledgments 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.

General Terms
Security, Double Spending, Blockchain Architecture, Peer-to-Peer Network

Keywords
Cryptocurrency, Blockchain Architecture, Peer-to-Peer, Online Transactions, Digital Payment, Double Spending Attack

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. The bank and merchant service providers are disrupted through mainstream acceptance of cryptocurrency payment services for peer-to-peer payments [1]. 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 [2]. 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 accuracy of destinations but also confirming the atomicity of each transaction [3]. Therefore from a small scale payment system to a larger digital payment platform a proper transaction verification model is identified as mandatory.

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 [4]. 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. One of the widely known such threats is double spending problem. The most popular bitcoin has strongly addressed these security related issues for the computation power based digital currencies [1]. But still, the problematic security threats exist for non-computational powers based currency systems such as service-oriented digital currency platforms. 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.

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

2. RELATED WORK
One way to safeguard from double spending attack is to maintain a third party authorized person to verify transactions within the network [5]. However, it is not applicable in p2p

networked payment systems subsequently there is no centralized control over the network [2]. 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 [6]. 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 [1]. Google Wallet has addressed relay attack and has overcome this by installing secure element applets [7] [8].

The CAFE Consortium had applied cryptographic techniques and had produced a secure open system for consumer payments using electronic money which consists of a 'CAFE infrared wallet' and a card [9]. This is developed as a public key system for electronic wallets. The 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 [1]. 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. Therefore bitcoin's transaction verification mechanism is somewhat simplified though cannot apply for mobile based payment systems due to the heaviness of its blockchain [10]. Here 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 [11] [12]. 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. The 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) [13]. Though it is symmetric it has got the advantage of not relying on the time-consuming modular arithmetic. Authenigraph is another option to provide security against a variety of attacks known within the online transaction environment. In bitcoin users execute the payments by digitally signing the own transactions. They have 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.

3. METHODOLOGY

In this section present the design of proposed approach, the detailed design of the secure transaction handling to overcome 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 earned rewards on services. The overall architecture of the target system is illustrated in the below fig 1.

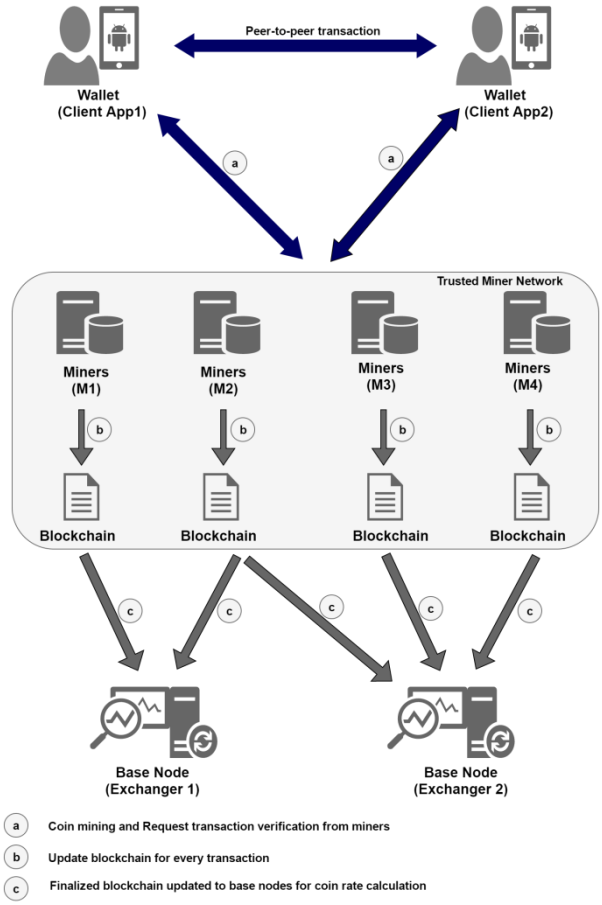

Fig 1: Social Currency Payment Platform Architecture

3.1 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 strong asymmetric algorithm is designed to apply on top of the hashing and obtains a signature to integrate with each protocol. Asymmetric over symmetric is selected since digital currency is a core asset in the system [14]. Digital signature is designed to use for verifying each protocol integrated data at each peer destination when sent over the network. A transaction can either contain a coin or details of a payment. By analyzing 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, which is also designated by a public key [5]. 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 is dropped and canceled. 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 [9]. 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.

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.

3.1.1 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 types of protocol designs are identified as essential [15]. The sender, receiver, time stamp and the particular digital signature are included in all five protocol designs mandatorily in order to verify a transaction. The designs differ by the set of integrated parameters and by the particular 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 fig 2. The S_ID and S_PARA denote the service id and service specific further details such as a carpooling ride distance or a shopping bill id respectively. The S_LOCATION represents the coin miner's location and PROP_VALUE stores the probability value associated with the trusted network concept which 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.

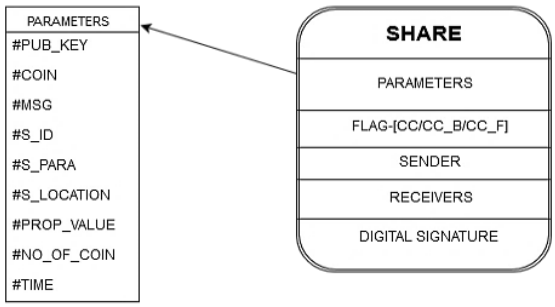

Fig 2: Protocol Design: 'SHARE' Protocol

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. Similarly, 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. Also, 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. Furthermore, the '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.

3.2 Transaction Verification Scenarios

The major components involved in transactions of such a payment platform are Miners and Application nodes as identified by the analysis of target system architecture in fig 1. All the transactions are categorized under three main scenarios based on the involved parties in each transaction and are differentiated in the following fig 3.

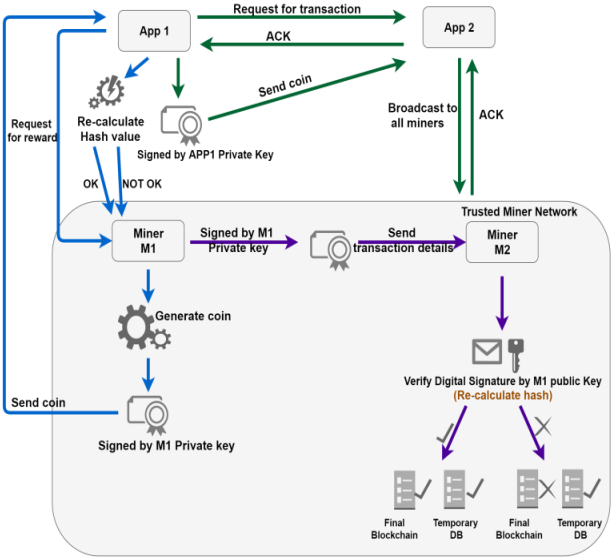

Fig 3: Transactions Overview in a Common Payment Platform

3.2.1 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. The design solution for this scenario is illustrated in below fig 4.

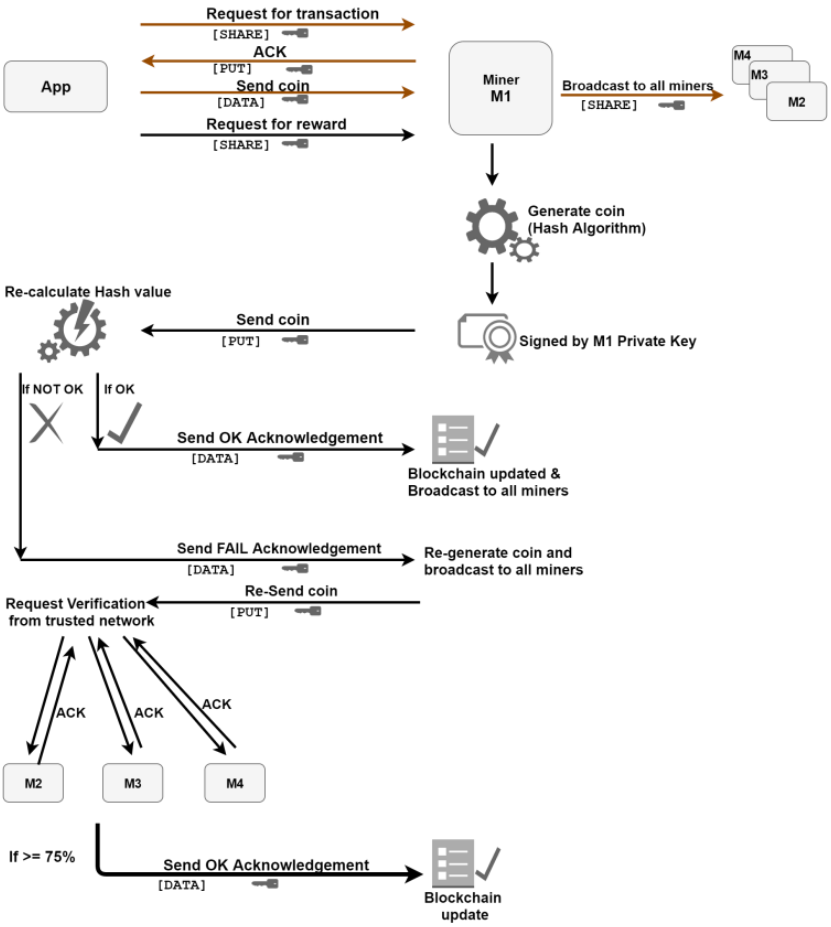

Fig 4: Scenario A: Miner-Application Transaction Verification Model

The 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 to Application (User) Transaction. Else a 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) to Miner Transaction.

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 destination when a particular miner retrieves the request, the miner checks the protocol's sender, receiver details and primarily verifies 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. 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 the hash.

If the verification is failed: the application node should send an ACK back to miner notifying the failure by digitally signing. 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.

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 [16]. 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 varies upon the p2p network conditions.

3.2.2 *Scenario B: Miner to Miner Transaction Verification Model*

This scenario is regarding transactions among service-oriented currency miners as illustrated in fig 5. All the miners are considered as in a trusted network. Two possible requirements are identified to initiate a p2p transaction among service-oriented miners.

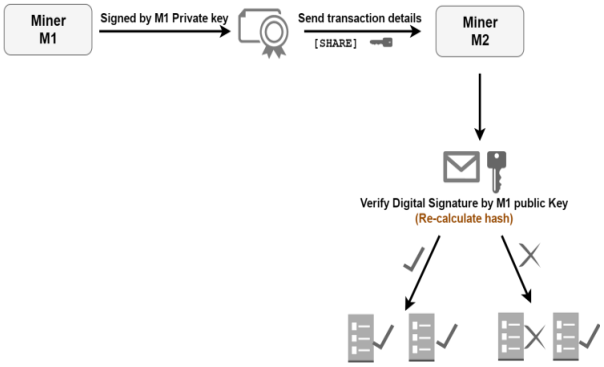

Fig 5: Scenario B: Miner-Miner Transaction Verification Model

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. 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.

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.

3.2.3 *Scenario C: App to App Transaction Verification Model*

This scenario is regarding transactions among user applications and the design is illustrated in fig 6. In this scenario, double spending is identified as crucial. Therefore the design solution is considered in prohibiting a user sending the same coin to more than one application users.

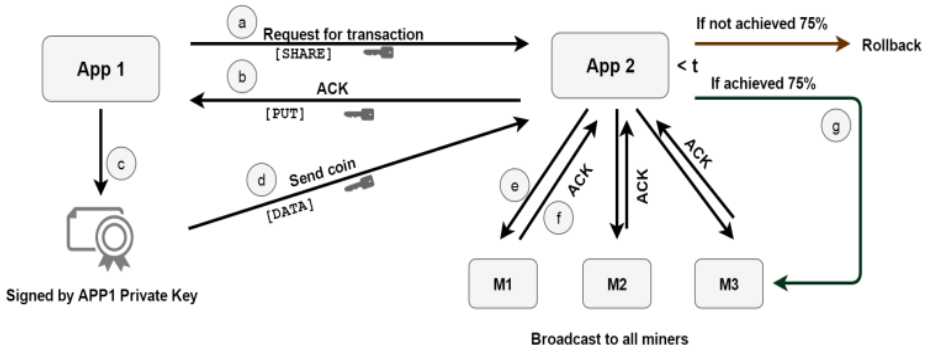

Fig 6: Scenario C: App-App Transaction Verification Model

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. 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

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 [17]. 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 [18]. 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 [19]. 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.

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 [20]. 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 [21]. 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.

5. EVALUATION

The implemented transaction verification model is based on the foundation of the blockchain architecture that consists of two abstract levels of verification. The primary verification concepts applied are the RSA asymmetric digital signature mechanism along with SHA-256 one-way hashing. 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.

In the identified service-oriented common currency platform the double spending is related to two scenarios among the all three scenarios of transactions. The scenario of p2p

transactions among a miner and a user application and transactions among user applications 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. 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/ 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.

As discussed in the conceptual solution, the trusted network of miners with 75% of probability level is focused on the challenge of double spending prevention [5]. The trusted network of miners relies on the adapted blockchain architecture which is supposed to maintain the history of transactions [16]. 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.

In considering the scenario A: 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: 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 that all three involved user applications are made available online continuously. In the scenario A, 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 spends the coin for a service as a payment or exchanges 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: transactions among user applications are illustrated in the following fig 7.

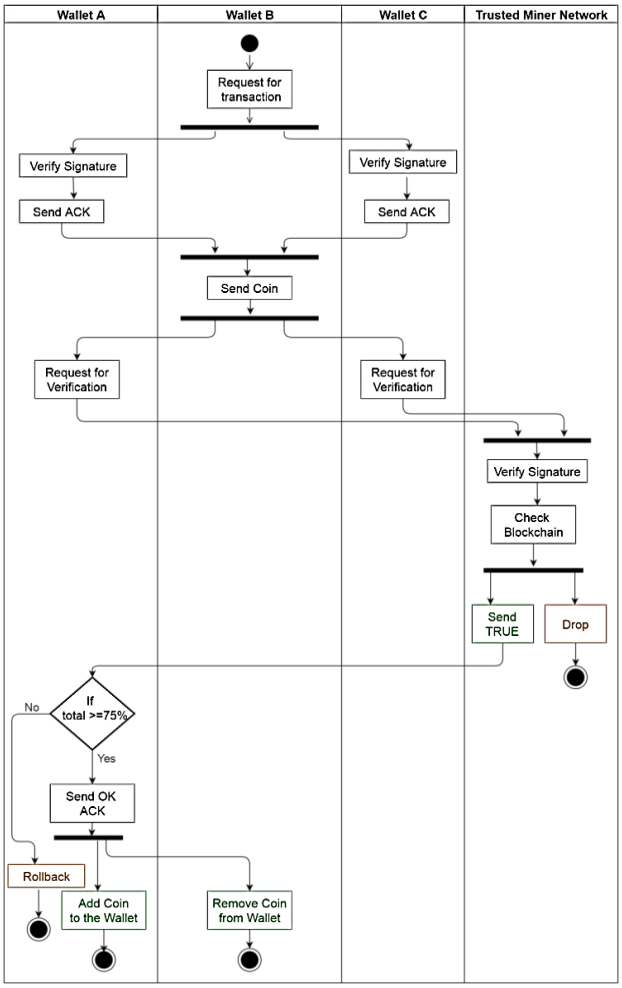

Fig 7: Double Spending Evaluation

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, a 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 50 times, the average 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.

6. CONCLUSION AND FUTURE WORK

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.

Considering different aspects and possibilities there are several future directions that can be suggested for the work of this paper. One is 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. And a reputation-based model for building the trusted network of currency miners to optimize the performance in transaction verification is another future work: 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. Also, a peer-to-peer transactions verification model for offline transactions is another extendible area: 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.

7. ACKNOWLEDGMENTS

I would like to show my gratitude to the supervisor Dr. T. N. K. De Zoysa and the evaluator Dr. C. I. Keppetiyagama at University of Colombo School of Computing for their guidance.

8. REFERENCES

[1] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2012. [Online]. Available: https://bitcoin.org/bitcoin.pdf. [Accessed 2 May 2016].

[2] M. Bawa, et al., “Peer-to-Peer Research at Stanford,” Stanford, 2007.

[3] E. Nordstrom, “Personal Clouds: Concedo,” 2015.

[4] J. Wells, et al., “Enhanced Security for Preventing Man-in-the-Middle Attacks in Authentication, Data Entry and Transaction Verification,” in Australian Information Security Management, Perth, Western Australia, 2008.

[5] Bitcoin.org, “GitHub-Bitcoin,” GitHub.Inc, 04 January 2017. [Online]. Available: https://github.com/bitcoin. [Accessed 24 May 2016].

[6] M. Katina, in Innovative Automatic Identification and Location-Based Services, IGI Global, 2009, pp. 25-233.

[7] R. Handa, et al., Google Wallet - A Glimpse into the future of mobile payments, 2011.

[8] M. Roland, “Applying recent secure element relay attack scenarios to the real world: Google Wallet Relay Attack,” Hagenberg, Austria, 2013.

[9] R. Tripathi and S. Agrawal, “Critical Analysis of RSA Public Key Cryptosystem,” International Journal of Advanced Research in Computer Science and Software Engineering, vol. IV, no. 7, 2014.

[10] M. J. Casey and P. Vigna, “Bitcoin and the Digital-Currency Revolution,” The Wall Street Journal, 23 January 2015. [Online]. Available: http://www.wsj.com/articles/the-revolutionary-power-of-digital-currency-1422035061. [Accessed 2 April 2016].

[11] P. McCorry, et al., “Refund attacks on Bitcoin’s Payment Protocol,” UK, 2016.

[12] M. Lei, “Exploiting Bitcoin’s Topology for Double-spend Attacks,” Zurich, 2015.

[13] M. Karpinsky and Y. Kinakh, “Reliability of RSA Algorithm and Its Computational Complexity,” International Scientific Journal of Computing, vol. II, no. 3, 2003.

[14] R. Tripathi and S. Agrawal, “Comparative Study of Symmetric and Asymmetric,” International Journal of Advance Foundation and Research in Computer (IJAFRC), vol. I, no. 6, 2014.

[15] D. Roio, et al., “Design of Social Digital Currency,” 2015.

[16] K. Croman, et al., “On Scaling Decentralized Blockchains,” 2015.

[17] M. Stevens, “Attacks on Hash Functions and Applications,” 2012.

[18] S. Aggarwal, et al., “A review of Comparative Study of MD5 and SHA Security Algorithm,” International Journal of Computer Applications (0975 – 8887), vol. CIV, 2014.

[19] P. D. Harish, “Towards Designing Energy-Efficient Secure Hashes,” Florida, 2015.

[20] IBM, “Learn How to Choose the Right Database for the Job,” IBM, 2016.

[21] “GitHub:senzprojects/udp-switch,” 20 August 2016. [Online]. Available: https://github.com/senzprojects/udp-switch. [Accessed 30 April 2016].