TheCoin: Privacy and security considerations within blockchain transactions

Mohamed ikbal, M.I.N, nacer
bournemouth university,
bournemouth, uk

Simant, S.P, Prakoonwit
bournemouth university,
bournemouth, UK

Edmon, E.P, Prakash
Cardiff Metropolitan University,
Cardiff, UK

ABSTRACT

TheChain is a solution to many problems such as monopoly, heavy state transition, and security vulnerabilities. TheChain solves these problems by introducing the intersection of regions as an incentive before allowing validators to nest a client directory. Intersecting their operating territories forces them to keep a watch over each other. The definition of privacy can take many forms, starting from the right to be forgotten beside being away from public attention. Although the pseudonymity of the user within the network can enhance the user's privacy, several pieces of research have studied the techniques to take advantage of the network structure to identify the users of pseudonyms. Moreover, two models have been used to record the updated exchange of values within the blockchain system, which are the unspent transaction output (UTXO) and the balance model. The UTXO suffers from duplication of information and the balance model suffers from having a single point of entry. This paper introduces TheCoin model that defines the protocol of the exchange of valuable datum within TheChain system. The solution has introduced a novel approach of initiating the transaction from the receiver side by taking advantage of mobile agents empowering a topology hiding to the network. Billing within the platform has been introduced to allow advanced contractual logic to be adopted into the system on the information level. Moreover, traceable fuzziness has been used to eliminate duplication. The paper presents an evaluation of the TheCoin model in terms of system security, block size, and search performance.

CCS CONCEPTS

• **Network** → Network protocol; Application layer protocol; peer-to-peer protocol..

KEYWORDS

Blockchain, UTXO, Transaction, Mobile agent, topology hiding

ACM Reference Format:
Mohamed ikbal, M.I.N, nacer, Simant, S.P, Prakoonwit, and Edmon, E.P, Prakash. 2021. TheCoin: Privacy and security considerations within blockchain transactions. In *2021 2nd Asia Service Sciences and Software Engineering Conference (ASSE '21), February 24–26, 2021, Macau, Macao.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3456126.3456127

ASSE '21, February 24–26, 2021, Macau, Macao
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8908-2/21/02…$15.00
https://doi.org/10.1145/3456126.3456127

1 INTRODUCTION

Money laundering, counterfeiting, and theft are results of information manipulation of the recorded financial ledger or the lack of means to trace and verify the authenticity of a claim. The use of electronic payments has eliminated many problems that tangible payments pose [5]. Centralisation has always been an issue due to concerns about privacy and high transfer fees. Consequently, distribution within the blockchain technology has not only eliminated the high costs but also introduced pseudonymous management of funds. However, many techniques have been found to link the real and pseudonomic identities'. It can be concluded that the violation of the right to be forgotten and public exposure are drawbacks of this technology. Money laundering is based on the manipulation of the value of information and that manipulation is helped by the probabilistic finality and the lack of traceability'. Therefore, allthough public transactions are a strong deterrent', it must be embedded with traceability techniques.

Tracking finance is very ancient and can be traced back in time to the Babylonian, Egyptian, and Sabaean civilisations. Today, double-entry accounting is mostly used, and the use of triple-entry accounting is on the rise as well. Recording all information within the blockchain ledger to make it public diminishes the ability of malicious users to manipulate or elude traceability. The adoption of blockchain technology by the financial sector has been a hot topic of discussion and research in recent years [3]. The first proposal of the system was to develop a prototype that exchanges financial information and eliminates double-spend. The bitcoin [12]proposal aimed to create a new type of money and eliminate the trusted party, but this solution is vulnerable to monopoly [11]. The monopolist is turning out to be the new foundation of trust. It has provided a new approach to validate transactions by making a malicious node weak compared to those interested in the ledger validity for their financial benefit. However, it was clear from the first project that the aim is to eliminate banks than targeting the concept.

The ledger is organised as a sequence of transactions nested with owned objects that have been generated during the mining process or through an exterior investment as another type of fiat money. However, within the transaction, there are two types of information: the unspent transaction model (UTXO) and the balance model. These two types of solutions have many disadvantages in a distributed environment. However, many types of transaction initiations can take place depending on the type of wallet used. Paper, hardware, and phone wallets make the owner of the funds the initiator while the web solution assigns a trusted agent to manage the fund. The audit system in banking is based on the use of the internal network run by local servers and mirroring techniques [20]. This kind of systems intend to eliminate any double-spending by fostering membership and the centralisation of decision-making.

TheCoin aims to adopt cash ideology electronically by ensuring the central bank as the issuer of the currency and use of a mobile agent to make a secure way for exchanging keys besides making the receiver of funds the initiator of the transaction. The next section describes the literature on blockchain as a technology, mobile agents, the UTXO model, and Zero-Knowledge proof. The third section will introduce TheCoin architecture and functioning. The fourth section will compare our proposed system with previous works in terms of security evaluation and expected performance.

2 RELATED WORK

Bitcoin [12]or Blackcoin [18]are proposals that have used the Hashcash proof of work (PoW) [2] in the validation of a list of transactions. Each transaction does hold an ensemble of valuable objects. The exchange model is named the unspent transaction output (UTXO). The Ethereum Foundation preferred the use of the balance model as an updated account value over the state transition system. Zerocash proposed the use of Zero-Knowledge proof over the data structure to delink the transaction from the identity. However, it led to the need for a sophisticated wallet that can save all the related proofs. Although there have been many proposals to switch from PoW to Proof of Stake (PoS) [19] or to take advantage of graphs in Tangle IOTA [15], the exchanged datum was always to be chosen from those two models.

The UTXO model is based on the continuous exchange of values described in terms of input and output attached to a transaction. The input can be seen as a list of duplicated coins that have been attached, in which the aggregated values must be equal or superior to the transferred value. It will generate an output that represents new coins. The transaction will state the sender associated with the input coins, and the receiver and the validator of the block will be associated with the output coins, in which the transaction will stand for the total transferred value and fees of validation. The transaction list will be wrapped into a block that generates a network reward for the validator [7]. It can be observed that, at some point, coins will turn to be of no use except for wasting a great amount of memory. Moreover, in the case of the bitcoin platform, it is developed as a knowledge-base that duplicates all the coins attached to their owner identity. The drawback of this solution is the unexpectedly massive growth of the micropayment exchange. The criterion for the validation of a transaction is that the value of input must exceed the value of the output over and above the transaction fees and the new transferred value.

The Balance-Approach is a more natural approach for the management of funds. The adoption of this technique in blockchain technology started with the Ethereum project. The solution models the system as the growth of an updated balance. However, the data structure of the transaction contains the sender and receiver keys beside the transferred value. The updating of the account will be subject to normal number manipulation, but the distributed criteria make it vulnerable to the replay attack [9]. The solution introduced the nonce number that will provide each transaction with a unique identifier and eliminate the threat of replay attack. The Ethereum approach has a strict condition on the nonce number, which leads to the elimination of any parallelism and the approach suffers from one point of entry requirement. TheChain [11] proposed the use of a data structure that contains a balance and UTXO model in which the balance variable is just used to accelerate the decision making. Consequently, it can benefit from privacy and the parallelism of coins besides the easy management of the contract with the balance model. The special aspect of the transaction that follows the UTXO model is its capability of holding many receivers, whereas, in the account approach, it is more appropriate to associate each transaction with a receiver and a nonce number.

The zero-knowledge succinct non-interactive arguments of knowledge (ZK-snark) have been implemented in blockchain technology to provide privacy by unlinking the data from the identity [16]. The ZK-snark simply hides the true value by obfuscating submission to other peers within the network. It is based on the generation of verifier and prover algorithms that are cropped into many small steps. Each step is converted through a Rank-1 constrain system [8] with three matrices that contain, as elements, a simple number of Boolean objects, each of which stands for an existing variable The prover must send relation generated from a witness matrix to the verifier that validates and ensures the knowledge of the solution. The solution is used to ensure the secure exchange of information. It has attracted much attention since the ZeroCash proposal [16]. It aimed to solve many problems, such as tracking of identity online or analytics to understand the different exchanges [6]. However, the user may be a subject of tracking through IP addresses due to the rigid connection with peers that stand for miners through the DNS server that returns a specific peer for each user.

Previous works in the literature discussed the UTXO model that can be used to enforce high privacy, but the model leads to a high and is an expensive search on the ledger or in terms of memory with massive growth in the coin's knowledge base. On the other hand, the balance mode leads to a high validation schema at the price of privacy. TheChain has combined both approaches to introduce a model in which the user can benefit from easy management of funds through a balanced approach and the mixing of public keys to increase privacy. The adoption of the technology on a large scale needs a more convenient approach than entrusting keys to the third party to manage the user's personal wallet. Consequently, this work aims to use mobile agent nested zero-knowledge proof as a way to exchange public keys between the initiators and the receivers of transactions. TheCoin uses a fuzzy reference to associate the spent and unspent coin. Finally, this work asks the following question: "If cash is paper signed by central banks, can it be replaced by data signed by the central bank?"

3 THECOIN

3.1 Data Structure

TheCoin is a model that is proposed to adopt cash ideology within the blockchain system from a fiat perspective based on authenticity criteria. This work is a continuation of the work on TheChain that proposes the adoption of a cash ideology by attaching to each coin a unique identifier. The data structure is the background for a good search algorithm.

The class coin stands for an element of the data structure that holds a unique identifier saved in the identifier variable. The issuer, validator, and owner proof signatures are generated on a different

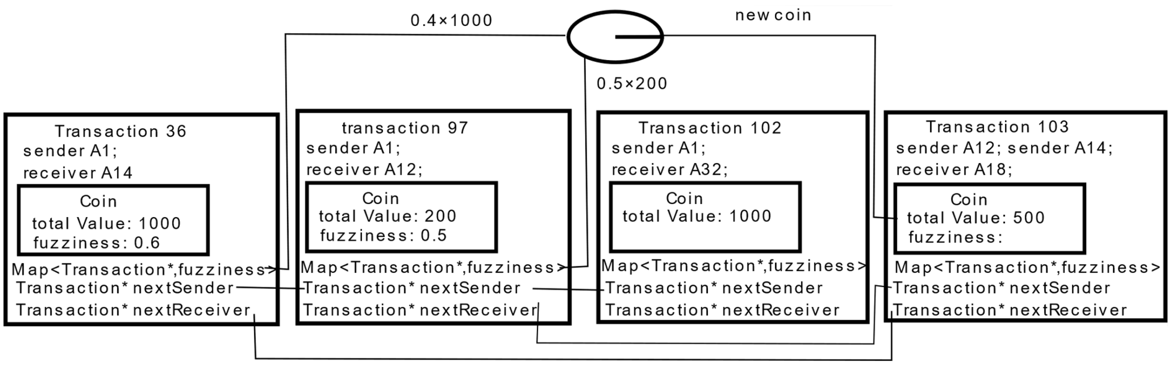

Figure 1: TheCoin Traceability

level. The first coin is generated and signed by the issuer. Therefore, issuerType will be zero, and a sub coin that derives from the main-coin will be signed by the validator. The signature of the new owner is a must in the exchange. Value is the variable that stands for a value of this coin, fuzzinessMap represents a pointer to the next transaction where this coin has been used partially or totally. Layer stands for the potential application of fuzziness over the controlled space described in MAXVALUE and MINVALUE in which each incrementation/decrementation stands for multiplication or a division by hundred. The class transaction is a wrapper of many coins in TheChain. In this system, the user coins will be linked consecutively through the use of pointers. the nextSender and NextReceiver serve the system by providing a total order on the memory reference layer between all the transactions, generating an accurate sequential number for the sender and the receiver. Sender and receiver are public keys that stand for the management of the fund. However, the use of a ring signature provides the user with a high level of privacy. The fundTotransfer is the value transferred to the receiver who will be the initiator of the transaction with the signature. The receiverSignature stands for the signature generated by the receiver. Figure 1 shows the sequence of a transaction and the way each transaction is referred to in the memory reference layer.

```
Class Coin:
    double MAXVALUE;
    double MINVALUE;
    String identifier;
    double value
    String[] parentIdentifier;
    Map<Transaction*, double> fuzzinessMap;
    int layer;
    int issuerType;
    Byte[] IssuerSignature;
    Byte[] validatorSignature;
    Byte[] OwnerProofSignature;
Class Transaction:
    Integer[] sequentialNumberSender;
    Integer[] sequentialNumberReceiver;
    publicKey[] sender
    publicKey[] receiver
    double[] fundTotransfer;
    List<Coin> coins;
    Vector<Transaction*> nextSender;
    Vector<Transaction*> nextReceiver;
    Byte[] receiverSignature;
```

3.2 Information's search

The search for related information can be a very expensive process in an open system. The Ethereum Foundation Balance-Approach can be faster for information retrieval as it is always a subject of addressing last element to be updated, but it lacks many advantages of coin management, such as- privacy and validation parallelism. TheCoin runs over TheChain data structure with the construction of transactions that are initiated. Algorithm 1 below describes the search for related information that is later sent from the validator to the sender to be signed before being injected into the transaction initiated by the receiver.

The search for related information can build a vector from one or many senders. Consequently, the tracker, which stands for an object that saves the reference for the first and the last transaction of unused funds by the owner, is requested to give the memory reference of the attached sender's first transaction. It extracts a list of coins attached to the transaction before entering a loop to calculate the total value of the coins. Some coins are used totally. In the UTXO model, such a case is handled by using input coins on which the output is based. This approach causes duplication of an object that will not be used again. TheCoin proposes the fuzziness with a layer to identify the exact position of the last coin that was partially used. The setCoinfuzziness method takes the coin and the difference as parameters to generate a layer that represents the number of decimal places after a number in the hundreds. For example, a coin that holds 1000 as a value and then the owner pays 999.999, it will generates a new coin for the new value with a unique identifier. The remaining value of 0.001 is set in the fuzziness variable The new coin is created through the generation of the unique identifier and its signing by the validator before being injected in the specific order to secure traceability to the original issuers.

3.3 Transaction Validation

The transfer of money with blockchain technology suffers from unreadability of the public keys leading to its exposure on different

Algorithm 1 Search for Related Information

```
input: listOfsenders, values;
output: vector<Coin>
int fund ⟵0;
boolean done⟵false;
vector<coins> toTransfer;
for(= 0;i< listOfsenders.size();i++):
  done=false;
  Transaction* first ⟵ Tracker.get(listOfsenders.get(i));
   while(!done):
    vector<coin> coins ⟵ first->getcoins();
    for(j=0;j<coins.size();j++):
      fund⟵fund+ coin.get(j).getTotalValue();
      if((fund - allPrevisousValue(i)) > values.get(i)):
        done⟵true;
        if(fund> valueToTransfer):
          double value ⟵ coin.get(j).getTotalValue();
          double difference ⟵ valueToTransfer-fund;
          first->setCoinfuzziness(difference, values.get(i));
          toTransfer.add(new Coin(value - difference))
          break;
      toTransfer.add(coin.get(J));
     If(!done OR (fund - allPrevisousValue(i)) ==values.get(i) ):
        if(first.getsender()== listOfsenders.get(i)):
          first ⟵ first->getNextSender();
        else:
          first⟵ first->getNextReceiver();
 updateFirst(listOfsenders.get(i), first);
```

forums or entrusting the wallet management to a trusted party. However, even if the user is capable of handling the last issue, the receiver may have to deal with the malicious activity of the sender by investing in probabilistic finality that may go through many stages by playing on rules such as the longest chain [21] or network convergence. Moreover, users may use analytical techniques or sniffing to locate the current owner of the fund. TheCoin introduces the concept of mobile agents as the mechanism of exchange of keys and validation of transactions based on solving a zero-knowledge proof puzzle between the two parties. Though the initiator of the transaction may be either the sender or the receiver, by default, that person is set as the receiver, and the mobile agent is dedicated to verifying the proof of transfer. The agent is defined as an extension of the object and enriched by the concept of autonomy [3]. Autonomy is the capability of the agent to seek only its own interest, which is derived from its ability to decide.

The message contains the basic variables that are shared to validate the identity and transfer; it also contains the public keys that are the signature that validates the transferred value generated from the owned coins. The message also generates the prover signature and the shared sentence and the public key for the prover algorithm.

The verifier agent expects an ensemble or a sender to participate in the transaction, in which it will be saved in a senderNickNames vector initiated by the user. The same AgentID helps in identifying the agent to serve in the exchange. The list of messages contains an ensemble of the received messages from the expected senders. The agent will be moved to a specific location that stands for the container ID or platform ID of a validator. The agent will register its service on that page, stating that it will appoint a validator to a transaction with a specific ID number. The agent will enter a loop that is initiated with the size of the expected sender for this transaction. The method, receiveBlocking(), will be blocked till a message is received that the sender will be checked to see if is a member of the senders' list before being added to the list of the messages. Finally, two behaviours, named verifierBehaviour and intiatTransaction, will be added to the agent to collect proofs. Finally, the agent will be back to the receiver with a transaction to be signed before the broadcasting for validation.

The following is a mobile agent that will run on the platform:

```
Class AgentVerifier:
Private vector<String> SenderNickNames;
Private vector<Message> messages;
protected void setup():
  move(location);
   registerInYellowPage()
  for(i =0;i<list.size();i++):
     Message msg = receiveBlocking();
      if( msg!=null):
       if(senderNickName.contain(msg. getNickNames()):
                 messages.add(msg);
    addBehaviour(new VerifierBehaviour());
    addBehaviour(new IntiatTransaction());
    move(receiverLocation);
Class AgentProofProvider:
 private String service;
 protected void setup():
      move(location);
     AgentDescription[]
     Agent=SearchYellowPage(service);
        addBehaviour(new ProvideProof
(Agent.getName()));
Class Message:
   PublicKey sender;
   Byte[] thePublicKey;
   Byte[] signatureProver;
   Byte[] sentence;
Class AgentProver:
  private String service;
  protected void setup():
    move(location);
    AgentDescription[] Agent=SearchYellowPage(service);
    This.addBehaviour(new
proverBehaviour(Agent.getName()));
```

The other two behaviours that are executed perform the same two purposes of collection of proofs and verification of the signature as the validator before initiating the transaction. However, the introduction of the code mobility by Picco [13] coupled with the concept of agent that was introduced by Russell and Norvig [17] can raise many issues and concerns of security relating to the host site. The user may also be subject to tracking through sniffing. However, the authenticity of the transaction lies in the digital signature. The inter-platform transfer of code can lead to rigorous interoperability standards that lower performance. However, the concept has many advantages, such as loose programme modelisation, which leads to easy integration, maintenance, and introspection, and the server host is expected to be well-equipped with security software.

Below is an expected implementation of the prover and the verifier behaviour:

```
Class VerifierBehaviour:
  Private VerifierModel model;
  Private vector<Message> messages;
  Private Transaction transaction;
    Public void action():
      Foreach(Message msg: messages):
          Byte[] public = msg.getPublic();
          Byte[] sentence =msg.getSentence();
          If(model.verify(public,signature,
          sentence):
          transaction.addsignature(msg,
  getSignature();
          transaction.setType(msg. getType());
          transaction.addPublic(msg.
PublicKey ());
          Broadcast(transaction);
Class ProverBehaviour:
  Private ProverModel model;
  private Byte[] thePublicKey;
  private String sentence;
  private String agentID;
    Public void action():
    Byte[] Signature = model(password
,sentence);
    Message msg= MessageFactory(agentID, sender,
thePublicKey, signatureProver ,sentence, Signature)
    sendMessage(msg)
```

3.4 Validator Billing

The transaction initiation within TheChain depends on fees designed for a specific validator securing its public state. The public state is verified through the duplication of the same information with all-region validators besides the intersected regions. The bill is a data structure that contains a map from services to a sequential number or contracts with the associated fee per transaction for each service and the total expected fees.

There are two types of billing, the posterior and the anterior. The anterior is based on the new client buying contract that depends on the number of validations for a specific service, implemented above the graph of the validation layer [11], and will be growing based on two factors: the bought token related to the anterior billing and the current balance. The posterior option is based on a special service given to some users for which they pay after the service has been consumed based on the level of risk. The receiver is obliged to pay the funds or lose the associated balance. The approach follows normal economic behaviour in which the user is subject to paying for retrieving information coming from government-related institutions, from the management of funds to the management of more complex information.

Algorithm 2 is executed before the injection of any block and the bill is associated with each receiver before calculating the risk implied by their profile and their current balances. Each user with a high risk receives billing immediately.

Algorithm 2 Bill Management

```
input: listOfTrs
output: map<
int size = listOfTrs.size();
for(int i= 0; i< size; i++):
    boolean submitted=false;
    Transaction trans = listOfTrs.get(i);
    List<Bill> Bills = getBilltrans);
    Profile = getProfile(trans)
    double risk = 0.0;
    for(int j= 0; j< size; j++):
            risk += calculateRisk(profile, factures);
            if(isRiskHigh(risk)):
                SubmitBill(trans,profile);
                submitted=true;
    if(trans.containContract()):
            updateBill(trans.getContract());
    updateBill(trans);
```

3.5 TheCoin Authenticity

The generation of tangible cash by central banks has many disadvantages, such as being prone to counterfeiting and the lack of traceability that leads to easy manipulation, which can be used for perpetrating frauds or money laundering. The idea of signed data instead of cash can be a solution to full digitalisation that can solve many real-world problems besides securing parallelism in execution. Moreover, it lowers the fees and can be executed under the bars of the expected World Bank goal, which is three percent. The authenticity of the coin can be the same as the authenticity of the transaction. That authenticity is ensured by the digital signature. The first depends on the issuer of the coin, and the latter depends on the validator of the transaction. The model proposes that all first-issued coins from the central banks should be signed. However, the system may later need the issuance of a new coin to finish a transaction, but each new coin is signed by the validator and linked to the previous one to ensure traceability.

4 DISCUSSION

4.1 The centric user measures

The idea of the UTXO model depends on the continuous generation of coins in which the validation of transactions requires the duplication of the previous coin and the generation of a new coin that holds the same value as the input. It states that the element of truth is centred around the transaction and forces the whole network to converge on one version of the ledger, which introduces a high competency between the nodes to converge.

The transaction, as central to the truth, is fed by several UTXO. In bitcoin ideology, the search for related information uses either a brute-force search from the root to the leaf or a bank of coins with a pointer to the attached transactions. The malicious user may invest in such vulnerability by building duplicate transactions with dispersed coins that can cause a delay in convergence and

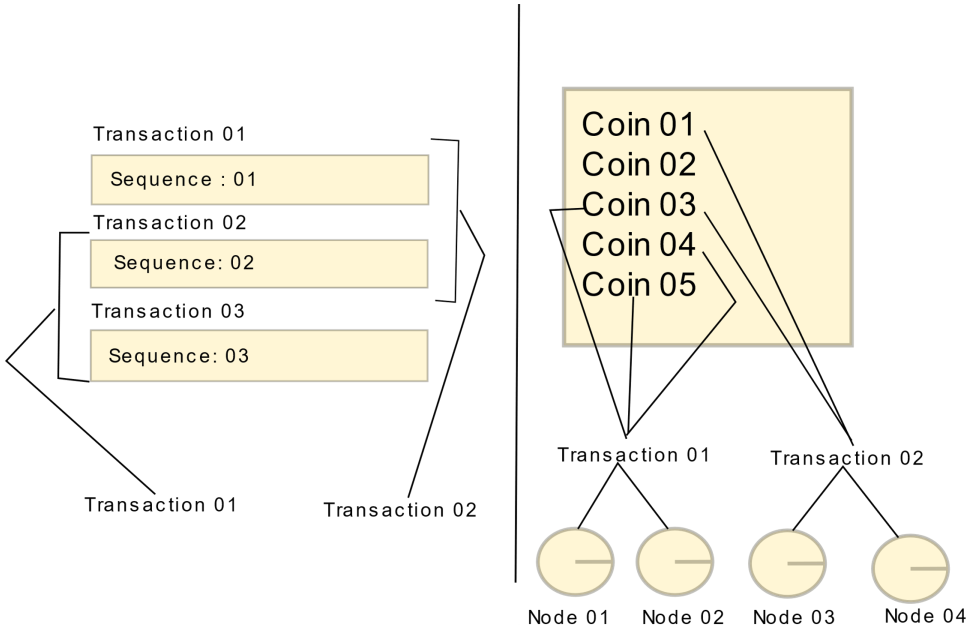

Figure 2: :TheCoin Model (left) vs Bitcoin UTXO model (right)

competency among nodes and by investing probabilistic finality and even double-spending through the generation of TheCoin's coin takes a different approach by using coins in sequential order because it is expected to run over TheChain data structure and reputation-based system that forces the sequential number to follow the standards. Thus, TheCoin choices unleash parallelism executed within the same region comparable to rigid standards for out-of-region execution. Moreover, it allows the ring signature to be used, leading to a higher degree of privacy. Figure 2 depicts the difference, in which the link between the sequential number and the used coins eliminates the huge number of delays introduced by the competency model of PoW. The duplicated coin on the right side, which is number three, has been used on both newly initiated transactions leading the two portions of the network to compete to secure the reward.

The blockchain is a distributed peer-to-peer system that needs bootstrapping mechanisms. The use of a DNS server is one of the solutions. However, the returned peers are the source of truth for the user. This renders the system vulnerable to various kinds of attacks such as RBG hijack. A study has shown that RBG attacks can eliminate 50% of bitcoin hash power by the elimination of less than a hundred gates. The eclipse attack aims to isolate a portion of the network to provide a unique history. The initiation of a transaction by the sender increases the risk of double-spending by investing in providing a partial truth to the receiver. However, TheCoin runs over a protocol that uses a reputation-based-
system in which the validator is subject to continuous validation that may cause the loss of its business in the system. On the other hand, the validation mechanism allows the receiver to be the initiator of the transaction. The capability of the receiver to aggregate proofs of transfer from different senders taking a validator as the host for the exchange of keys and proofs diminishes the possibility of isolation because, unlike previous works, it imposes a structured network with a hidden topology. However, the unstructured building of the network makes users connect to unrelated peers who might be maintainers working on different partial centralisations with low interest in ensuring the validity because of the lack of reward from it or its traceability. Consequently, a peer-selection approach must be adopted.

The users in previous works suffered from poor key management and the sharing of the public key on social media or forums, which violated the privacy standards. Moreover, the lack of understanding of the aim of the blockchain system led many users to trust their keys to third trusted parties, the avoidance of which was the first reason for switching to the blockchain network from normal financial behaviour. However, the protocol proposes the use of a mobile agent as the mechanism for exchanging public keys between the two parties. The users are subject to solving a zero-knowledge proof based on a password and a shared sentence generated and known to both parties.

4.2 Data structure measures

TheCoin model has been implemented over TheChain data structure that has been built over the Petri Network model in which the choice of modelisation offers the opportunity to link all transactions and updated wallets together. TheCoin objects are attached to transactions that are linked sequentially and tracked using memory references.

The elimination of the input values reduces the size of the block by reducing the size of each transaction in it. Table 1 presents 100 blocks in detail in one ledger before the calculation of the size of each object. As can be observed, in the case of micropayment, the mean size can be ten times more on the UTXO than TheCoin. The object size can be bigger if micropayment is lesser than the size used

Table 1: Block Size

	Micro (UTXO)	Not a micro (UTXO)	Micro(TheCoin)	Not a micro(TheCoin)
Transaction Nbr	100	100	100	100
Block depth	100	100	100	100
Mean	2217 kB	276.046 kB	204 kB	221.105 kB
Std	985 kB	7.833 kB	20 kB	4.498 kB
min	262 kB	247.648 kB	3 kB	208.008 kB
25%	1408 kB	271.728 kB	204 kB	217.976 kB
50%	2424 kB	276.568 kB	207 kB	220.824 kB
75%	3068 kB	281.432 kB	209 kB	223.672 kB
Max	3503 kB	291.128 kB	223 kB	239.336 kB

Table 2: Transaction Validation

	ImplementationA speed	Implementation B Speed	Implementation C speed	Implementation A with I/O	Implementation B with I/O
Transaction Nbr	100	100	100	100	100
Block depth	100	100	100	100	100
Mean	2.83 (ms)	4.013911 (ms)	631. (ms)	615 (ms)	722 (ms)
Std	1.02 (ms)	0.765979 (ms)	234 (ms)	290 (ms)	750 (ms)
min	1.88 (ms)	3.298200 (ms)	353 (ms)	292 (ms)	285 (ms)
25%	2.15 (ms)	3.493325 (ms)	455 (ms)	417 (ms)	383.5 (ms)
50%	2.36 (ms)	3.731950 (ms)	563 (ms)	551 (ms)	413.5 (ms)
75%	3.19 (ms)	4.187250 (ms)	775 (ms)	668 (ms)	467 (ms)
Max	6.93(ms)	6.490500(ms)	1249 (ms)	1727 (ms)	2994 (ms)

in our experiment, which was 0.001. In the case of normal payment, which is not micro, the average size of the object in the case of the UTXO model is 20% more than TheCoin. Moreover, it has been observed that since the standard deviation is not significantly large, there is no need to generate many new coins. It can be argued that the size of the block does really depend on the implementation, and the duplication drawbacks can vary among the blocks. However, eliminating the duplication lowers the size, which makes a huge difference, particularly in the case of micropayment.

The execution of TheCoin shows the same negligible performance as expected when all coins addresses have been saved in a separate base of knowledge. Moreover, another implementation that can be very expensive is the brute-force search in which the search for related information can sequentially use coins as done in TheCoin model but it is very expensive because there is no sequential link between the different transactions in previous works. The used machine is a 64-bit Processor Intel(R) Core(TM) i5-8250U, 1.60 GHz, 1.80 GHz, and 8 GB.

Table 2 is a presentation of three techniques. A stands for algorithm 1 injection with TheChain implementation. B stands for bitcoin ideology with a bank of knowledge that saves all coin references. C stands for the brute-force search within the bitcoin ideology, and the same implementation can be found in the work by li in [21]. As can be observed, the execution of 100 blocks in depth leads to the results discovered in which the difference between A and B is negligible with a mean of 2 and 4 milliseconds and with a low standard deviation due to the use of trackers for each element. However, C shows a very expensive search with a mean of 631 milliseconds and cumulative growth that affects massively the speed. The implementation with the use of IO access has demonstrated as well a negligible difference between the implementation A and B. However, in the use of micropayments, it is recommended that a base of knowledge with predictable models be implemented to manage the coins. Although the two approaches A and B perform very similarly on search, our approach eliminates the use of bank knowledge for coins through the use of pointers trackers.

4.3 System measure

The exchanged datum has a great impact on system performance. Table 3 shows the difference in the aspectual contribution that the different implementations can have. TheCoin runs over TheChain. Therefore, it absorbs the different criteria from both approaches before making contributions due to the use of mobile agents.

Parallelism is very high within the UTXO, model but it can lead to greater delay due to competency between different nodes. However, TheCoin used the sequential order to eliminate this. Moreover, the use of the coin objects model gives the system the advantage of the ring signature that can enhance privacy as compared to the balanced approach. TheCoin has facilitated the billing approach that uses contracts over TheChain system. However, it is hard to implement business logic over the UTXO model because it requires the generation of rewards, which does not fit within a fiat protocol. Finally, a consensual delay that can be derived from the valuable datum model can be very high in the UTXO model if it is implemented

Table 3: Conceptual comparison

	UTXO	Balance	TheCoin
Specific user parallelism	High	Not application	High within the same region. It is controlled outside.
Privacy	High (use of ring signature)	Depends on pseudonymity	High (use of ring signature)
Business logic management	Hard to implement	Easy to implement	Easy to implement
Consensual delay	High	Low	Low

over a probabilistic finality with no sequential estimation. However, TheCoin with a regional perspective can control parallelism, which leads to low consensual delay.

5 CONCLUSION

This work has introduced TheCoin protocol. It is proposed to be run over TheChain system due to the risk involved in running randomly referred coins over a regional space. The solution, as has been discussed, has shown an optimal performance in terms of size and the security issues that it mitigates. The paper can be summarised as follows:

1. Use of fuzziness to manage the partial use of sequential use of unduplicated coins.

2. Use of the mobile agent as a method to exchange public keys between the different parties.

3. Introduction of the reverse approach in which the receiver is the initiator of the transaction.

4. The introduction of the concept of the bill within the permissionless blockchain technology.

5. The concept of the coin authenticity.

The next work will address the ledger where the token will be changed to more complex symbolic elements than numbers. It will invest in the concept of coin authenticity by changing the value to information and build logical chaining mixed with a distribution-capturing approach from statistical methods to be applied to algorithm 1 and build a decision within TheChain.

REFERENCES

[1] Stefano Angieri, Alberto García-Martínez, Bingyang Liu, Zhiwei Yan, Chuang Wang, and Marcelo Bagnulo. 2018. An experiment in distributed internet address management using blockchains. arXiv preprint arXiv:1807.10528 (2018).

[2] Adam Back and others. 2002. Hashcash-a denial of service counter-measure. (2002).

[3] Maicon Azevedo da Luz and Kleinner Farias. 2020. TheUse of Blockchain in Financial Area: A SystematicMapping Study. InXVI Brazilian Symposium onInformation Systems. 1–8.

[4] Alex Biryukov, Ivan Pustogarov, Fabrice Thill, and Ralf-Philipp Weinmann. 2014. Content and popularity analysis of Tor hidden services. In 2014 IEEE 34th International Conference on Distributed Computing Systems Workshops (ICDCSW). IEEE, 188–193.

[5] Markus K Brunnermeier, Harold James, and Jean-Pierre Landau. 2019. The digitalization of money. Technical Report. National Bureau of Economic Research.

[6] Sudarshan S Chawathe. 2019. Clustering blockchain data. In Clustering Methods for Big Data Analytics. Springer, 43–72.

[7] Sergi Delgado-Segura, Cristina Pérez-Sola, Guillermo Navarro-Arribas, and Jordi Herrera-Joancomartí. 2018. Analysis of the Bitcoin UTXO set. In International Conference on Financial Cryptography and Data Security. Springer, 78–91.

[8] Max Hoffmann, Michael Klooß, and Andy Rupp. 2019. Efficient zero-knowledge arguments in the discrete log setting, revisited. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. 2093–2110.

[9] Jiliang Zhang, Yaping Lin, and Gang Qu. 2015.Reconfigurable binding against FPGA replay attacks.ACM Transactions on Design Automation of ElectronicSystems (TODAES)20, 2 (2015), 1–20.

[10] Olivier Moindrot and Charles Bournhonesque. 2017. Proof of Stake Made Simple with Casper. ICME, Stanford University (2017).

[11] Mohamed Ikbal Nacer, Simant Prakoonwit, and Ismail Alarab. 2020. TheChain: A Fast, Secure and Parallel Treatment of Transactions. In Proceedings of the 2020 2nd International Electronics Communication Conference. 81–89.

[12] Satoshi Nakamoto. 2019. Bitcoin: A peer-to-peer electronic cash system. Technical Report. Manubot.

[13] Gian Pietro Picco. 2000. Understanding code mobility (tutorial session). In Proceedings of the 22nd international conference on Software engineering. 834.

[14] Ariana Polyviou, Pantelis Velanas, and John Soldatos. 2019. Blockchain Technology: Financial Sector Applications beyond Cryptocurrencies. In Multidisciplinary Digital Publishing Institute Proceedings, Vol. 28. 7.

[15] Serguei Popov. 2016. The tangle. cit. on (2016), 131.

[16] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. 2014. Zerocash: Decentralized anonymous payments from bitcoin. In 2014 IEEE Symposium on Security and Privacy. IEEE, 459–474.

[17] Wenwu Tang and David A Bennett. 2010. Agent-based modeling of animal movement: a review. Geography Compass 4, 7 (2010).

[18] Pavel Vasin. 2014. Blackcoin's proof-of-stake protocol v2. URL: https://blackcoin.co/blackcoin-pos-protocol-v2-whitepaper. pdf 71

[19] Nikos Vlassis. 2007. A concise introduction to multiagent systems and distributed artificial intelligence. Synthesis Lectures on Artificial Intelligence and Machine Learning 1, 1 (2007), 1–71.

[20] Hakim Weatherspoon, Lakshmi Ganesh, Tudor Marian, Mahesh Balakrishnan, and Ken Birman. 2009. Smoke and Mirrors: Reflecting Files at a Geographically Remote Location Without Loss of Performance.. In FAST. 211–224.

[21] Congcong Ye, Guoqiang Li, Hongming Cai, Yonggen Gu, and Akira Fukuda. 2018. Analysis of security in blockchain: Case study in 51%-attack detecting. In 2018 5th International Conference on Dependable Systems and Their Applications (DSA). IEEE, 15–24.

[22] H. Zhou. 2019. Learning Blockchain in Java: A Step-By-step Approach. Independently Published. https://books.google.co.uk/books?id=gzSyygEACAAJ

[23] Maicon Azevedo da Luz and Kleinner Farias. 2020. TheUse of Blockchain in Financial Area: A SystematicMapping Study. InXVI Brazilian Symposium onInformation Systems. 1–8.