EARMARKED UTXO FOR ESCROW SERVICES AND TWO-FACTOR AUTHENTICATION ON THE BLOCKCHAIN

Jisha Pillai
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_projects

Part of the Information Security Commons, and the Other Computer Sciences Commons

Recommended Citation
DOI: https://doi.org/10.31979/etd.fcb5-rs6t
https://scholarworks.sjsu.edu/etd_projects/675

This Master's Project is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Projects by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.
EARMARKED UTXO FOR ESCROW SERVICES AND TWO-FACTOR AUTHENTICATION ON THE BLOCKCHAIN

A Project
Presented to
The Faculty of the Department of Computer Science
San José State University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by
Jisha Pillai
May 2019
The Designated Project Committee Approves the Project Titled

EARMARKED UTXO FOR ESCROW SERVICES AND TWO-FACTOR AUTHENTICATION ON THE BLOCKCHAIN

by

Jisha Pillai

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

May 2019

Dr. Thomas Austin    Department of Computer Science
Dr. Jon Pearce       Department of Computer Science
Dr. Philip Heller    Department of Computer Science
ABSTRACT

EARMARKED UTXO FOR ESCROW SERVICES AND TWO-FACTOR AUTHENTICATION ON THE BLOCKCHAIN

by Jisha Pillai

The security of accounts on the blockchain relies on securing private keys, but they are often lost or compromised due to loopholes in key management strategies or due to human error. With an increasing number of thefts in the last few years due to compromised wallets, the security of digital currency has become a significant concern, and no matter how sophisticated and secure mechanisms are put in place to avoid the security risks, it is impossible to achieve a 100% human compliance.

This project introduces a novel concept of Earmarked Unspent Transaction Outputs (EUTXOs). EUTXOs enable every user on the blockchain to lock their funds to be spendable only to a designated set of users, even if the private key gets compromised.

We validate the utility of EUTXOs by using it to implement an Escrow service in the blockchain to overcome the limitations introduced by traditional Escrow services. We also implement decentralized two-factor authentication (2FA) on the blockchain using EUTXOs and discuss the tradeoffs of this design.
TABLE OF CONTENTS

CHAPTER

1 Introduction ................................................. 1

2 Background .................................................. 5
   2.1 What is a Blockchain? ............................. 5
      2.1.1 Peer-to-Peer (P2P) ......................... 5
      2.1.2 Proof-of-Work .............................. 6
   2.2 Cryptography on Blockchain ..................... 7
      2.2.1 Hashing .................................. 7
      2.2.2 Digital Signature ........................ 8
   2.3 Unspent Transaction Outputs (UTXOs) .......... 9
   2.4 Transaction Fees ................................ 10
   2.5 Blockchain Risks ................................ 11

3 Escrow service .............................................. 12
   3.1 What is Escrow? ................................ 12
   3.2 Escrow On Blockchain ............................ 13
   3.3 Limitations ...................................... 15

4 Two-Factor Authentication with the Escrow service as a TTP 16
   4.1 How Two-Factor Authentication Works? ........ 16
   4.2 Centralized Two-Factor Authentication .......... 18

5 Earmarked UTXOs ........................................... 21

6 Restricted Escrow .......................................... 23
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Spartan Gold (SG) Transaction</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Digital Signature</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>Communication Using An Escrow Service</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>Transaction from Alice to the Escrow service</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td>Transaction from the Escrow service to Bob</td>
<td>14</td>
</tr>
<tr>
<td>6</td>
<td>Two Factor Authentication</td>
<td>17</td>
</tr>
<tr>
<td>7</td>
<td>2FA with Escrow Service on the blockchain</td>
<td>19</td>
</tr>
<tr>
<td>8</td>
<td>A sample transaction published to blockchain</td>
<td>19</td>
</tr>
<tr>
<td>9</td>
<td>Escrow To Bob Transaction</td>
<td>20</td>
</tr>
<tr>
<td>10</td>
<td>Earmarked Transaction</td>
<td>21</td>
</tr>
<tr>
<td>11</td>
<td>Earmarked Escrow</td>
<td>24</td>
</tr>
<tr>
<td>12</td>
<td>Initial UTXO</td>
<td>25</td>
</tr>
<tr>
<td>13</td>
<td>Transaction From Alice To Escrow Service</td>
<td>25</td>
</tr>
<tr>
<td>14</td>
<td>Transaction From Escrow Service To Alice’s Paper Wallet</td>
<td>26</td>
</tr>
<tr>
<td>15</td>
<td>UTXO After Transaction</td>
<td>26</td>
</tr>
<tr>
<td>16</td>
<td>Enforced Escrow</td>
<td>28</td>
</tr>
<tr>
<td>17</td>
<td>Initial UTXO</td>
<td>28</td>
</tr>
<tr>
<td>18</td>
<td>Alice’s Transaction</td>
<td>29</td>
</tr>
<tr>
<td>19</td>
<td>Bob’s Transaction</td>
<td>29</td>
</tr>
<tr>
<td>20</td>
<td>UTXO After Alice To Bob Transactions</td>
<td>30</td>
</tr>
<tr>
<td>21</td>
<td>Decentralized 2FA Flow Diagram</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>---</td>
<td>-----------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>22</td>
<td>Earmarked Transaction</td>
<td>33</td>
</tr>
<tr>
<td>23</td>
<td>Transaction To Earmark SG Coins</td>
<td>34</td>
</tr>
<tr>
<td>24</td>
<td>Alice To Alice’s Mobile Transaction</td>
<td>35</td>
</tr>
<tr>
<td>25</td>
<td>Alice’s Mobile To Bob Transaction</td>
<td>35</td>
</tr>
</tbody>
</table>
CHAPTER 1

Introduction

There has been tremendous growth in the cryptocurrency market in the past decade. The primary reason behind the growth is its use of an ingenious technology called the blockchain, which is used to manage the digital currency in a distributed, decentralized, and immutable manner [7]. The blockchain maintains an incorruptible public ledger that is available to every user in the blockchain network. As there is no third-party or centralized authority involved in storing and maintaining the data, there is no single point of failure. Thus, it is difficult for hackers to tamper with or corrupt the data. The key features of the blockchain technology that led to its success are the peer-to-peer (P2P) (Section 2.1.1) technology used to maintain a shared distributed public ledger and the cryptographic techniques such as hashing and public-key cryptography (Section 2.2).

Although the blockchain technology can ward off many attacks, its security depends on users protecting their private keys. When the keys are lost, currency may get stolen. 5.1 million dollars worth of bitcoins were stolen from BITSTAMP on January 2015 [21], on August 2016 a 72 million dollars worth of bitcoins were stolen from BITFINEX [2]. 7 million dollars worth of Ethereum was stolen from COINDASH on July 2017 [23]. 16 million dollars worth of Ethereum and ERC20 were stolen from a New Zealand based cryptocurrency exchange Cryptopia on January 2019, compromising around 17,000 wallets [6].

The major drawback of using public-key cryptography is the inability to store and manage the cryptographic keys efficiently [3][4][5]. Inefficient key management leading to endpoint vulnerabilities (Section 2.5) is one of the primary reasons for most of the attacks on the cryptocurrency exchanges. Endpoints are nodes on the blockchain network where mining systems, PCs, or mobile phones reside. Nodes are
the basic building blocks of the blockchain network, as they are responsible for storing and managing wallets, initiating transactions, finding proofs, reaching consensus on the validity of the transactions, etc. Ownership of an account in the blockchain network is the same as owning the digital wallet. So if a wallet representing a node is compromised, there is no way for the blockchain network to validate the authenticity of the user [8]. Thus, nodes or endpoints are the most vulnerable points in the blockchain network.

As we all know, "To Err Is Human" and no matter how sophisticated key management strategies are put in place to avoid endpoint security risks, it is impossible to avoid the loopholes that come with each strategy and to achieve a 100% human compliance [11]. The goal of my project is to come up with security mechanisms that will guard the cryptocurrencies against theft occurring due to compromising of wallets and exchanges.

To achieve this goal, I introduce *earmarked UTXOs* (EUTXOs), with EUTXOs funds can only be sent to a set of accounts specified at the time the EUTXO is created.

Specifically, the approaches implemented in this work are shown below. Steps 1 and 2 are implemented and used for comparison by the later approaches to discuss how the limitations introduced by the first two approaches are overcome by approaches 3, 4, and 5.

1. Implement the Escrow service as a trusted-third-party on the blockchain and discuss its limitations.

2. Implement two-factor authentication on the blockchain using the Escrow service as a trusted third party and discuss its limitations.

3. Restricted Escrow introduces a novel concept of immutable *earmarked UTXOs* (EUTXOs) field at the transaction level to earmark/lock the funds to be spendable
to a list of recipients specified by the user. The approach combines the concepts of Escrow service as a TTP and EUTXOs; here, the Escrow service’s possible actions are restricted due to earmarked transactions.

4. Blockchain as Escrow with EUTXOs implements a decentralized Escrow service where each user acts as its own Escrow on the blockchain.

5. Decentralized 2FA implements two-factor authentication on the blockchain using EUTXOs.

A simplified blockchain based cryptocurrency called Spartan Gold (SG) [20] developed by Dr. Thomas Austin is used as a foundation to implement and experiment with different approaches incorporating EUTXOs. A simple Spartan Gold (SG) transaction is shown in the Figure 1. The input field shows the address of the user initiating the transaction and the output field contains the address of the receivers and the amount of SG coins assigned to them. The transaction is signed with the user’s private key and is included in the sig field; every transaction also consists of a timestamp.

```json
{ txDetails:
  { output:
    { '+dz4ctwTEXszyte91YabGW+i3NIn56IEqTQ3+v90ms=': 100,
      'ib1000ZF+Hccorreup5Fkd09x81bdY0N0G2sSHkDqU=': 33 },
    input: '+dz4ctwTEXszyte91YabGW+i3NIn56IEqTQ3+v90ms='
  },
  timestamp: 1556312367206,
  sig: '8727ff7806b4b3bd1cd6077468fa7a583037dbf9df228366412712dbd6f01ada84b79dd0d142eac3d45a21b81a613337ef2b32d848f166a8a1ab446f086cf838b35b9e4f48a0cc85f7dd4f6056f73ec711c1c604c238a54d4be6ef07f042aba337b9ca16e7d0f9fa14292b915c180b26cde80a99ab2a1860391ba0043ec3fcf25383b30a3f22ead89a1d523effc7deafa9d0e6a894c17292265f9fc3a1edc27dc86a30034ea7db8f244e5651002a0e30216196642ba93621357023ce806baf4836aed4c4a67d5d3fcb1b7e212c95db33109af771b8f1ad3ba2b05a4bd72a8ddf96586ba0f1a062a8e062d107b25e12b4ab9b8cb3bb73b41f9916b4c1cad' } }
```

Figure 1: Spartan Gold (SG) Transaction

Spartan Gold implements the UTXOs at the block level, whenever a miner finds the proof of a block and adds it to the blockchain, the UTXOs gets updated for accounts involved in the transactions included in the block. Every user initiating the
transaction should account for all the SG coins he/she owns, such that any leftover change must be assigned back to the user initiating the transaction. Failing to assign back the difference will result in the change being awarded to the miner who finds the proof of the block containing the transaction.
CHAPTER 2

Background

2.1 What is a Blockchain?

A blockchain [1] is a decentralized, distributed ledger available to everyone on the blockchain network. Each block in the blockchain contains the data and the hash of the previous block. The type of data in a block depends on the type of transactions being carried out in the blockchain. The data field of a block in the blockchain that does bitcoin transactions contains fields for the sender’s address and receiver’s address, and the amount of bitcoins being transacted.

A hash of a block is used to uniquely identify a block. This feature of the blockchain makes it difficult for attackers to tamper with the data in a block since any modifications to the block results in the change of the hash value associated with the block. Each block’s hash is computed using the hash of the previous block and the data in the current block, thus forming a chain of blocks. So any modifications to a single block in the blockchain will invalidate all the blocks that follow it. But with the computational power available these days, an attacker could easily re-calculate the hash of all the blocks that follow the block that was modified resulting in a valid chain.

To overcome this drawback, the blockchain uses something known as Proof-of-Work (PoW) discussed in Section 2.1.2. The advantage of PoW is that it delays the time taken to create a new block.

2.1.1 Peer-to-Peer (P2P)

One of the most critical factors which make the blockchain secure and decentralized is its use of a peer-to-peer (P2P) [1] network. Each peer is also known as a node. In the blockchain technology, each node on the P2P network has a copy of the distributed ledger. Since each node has a full copy of the blockchain, it becomes
impossible to destroy or tamper with the data. If someone wants to tamper with the
data, they have to tamper with the replication of data stored on every node in the
blockchain network, thus making it impossible to achieve. This decentralized nature
of the blockchain overcomes the limitations of using a centralized authority for storing
and accessing data, which will result in a single point for failure or loss of data, as all
the data is stored and managed at a single location (e.g., Banks).

Nodes on the blockchain network act as clients who make transactions and accept
new blocks added to the blockchain; they also act as miners. Miners on the blockchain
are responsible for validating the transactions. They verify the signature and the
correctness of data for every transaction. Miners are responsible for consolidating a
set of transactions into a block. Each miner competes with other miners to find the
Proof-of-Work (PoW) (Section 2.1.2).

Once the miner finds the proof of a block, the peers on the network validate the
proof. The block which is agreed upon by more than 50 percent of the peers gets
added to the blockchain. After the block is added to the blockchain, the miner with
the proof of that block gets rewarded. For Bitcoin, a new block gets added to the
blockchain network every 10 minutes on average, and the miners are rewarded with
both transaction fees and newly minted bitcoins.

2.1.2 Proof-of-Work

Mining is carried out in Bitcoin using a Proof-of-Work (PoW) \[9][15] system.
Miners compete with each other over the blockchain network to compute the proof
for a block. PoW is a complex mathematical problem that the miner has to solve
using a cryptographic hash algorithm. It is used to prove that a certain amount of
computation is performed before a block is added into the blockchain. It requires
the hash of each block to start with a certain number of 0’s. Since the process of
generating the hash which meets the required criteria is a stochastic process, it requires
a lot of computation to calculate the hash, thus delaying the process of adding a block
to the blockchain.

In the case of Bitcoin, it requires 10 minutes on average to find a PoW for a new
block and add it to the blockchain. Even though a miner finds the PoW for a block,
the block is not necessarily accepted into the blockchain. The reason behind that is
that the blockchain is decentralized, so every user on the blockchain has their own
copy of the ledger available. When a miner finds the PoW for a block, it is published
into the network, and a copy of it is sent to all the users over the network. Once
the peers validate the proof of the block on the network and the block is added to
the blockchain, the miner who found the proof is rewarded with new bitcoins or with
the leftover change from the transaction in the form of a transaction fee. Currently,
12.5 bitcoins are generated per block. The number of bitcoins generated per block is
reduced by half for every 210,000 blocks generated [10].

2.2 Cryptography on Blockchain

The Blockchain uses hashing and digital signatures. Hashing is used by the
users on the blockchain network to reach an agreement on the current state of the
blockchain. Whereas, digital signatures are used to verify that the owners of the
transactions are who they claim to be.

2.2.1 Hashing

Hashing is a powerful technique used these days as fingerprints for files, documents,
and transactions. The hashes of the documents or transactions are computed to ensure
that the data is not tampered with in an unintended way. Hashing is the process of
taking input of any arbitrary length and converting it into an output of fixed length.
For example, the hashing algorithm Secure Hash Algorithm-256 (SHA-256) takes any
length of data as input and converts it into a 256-bit output. This feature of hashing algorithms makes it easier to verify and handle a large amount of data. The key properties of the hashing algorithm that makes it ideal for use in blockchain are:

- Deterministic: Hash of a particular input always results in the same output
- Efficient: Able to compute hash in a small amount of time irrespective of the size of the input data.
- One-way: It is impossible to determine the input data from its hash value.
- A small change in input changes the hash value

Blockchain, as we discussed in Section 2.1, is a chain of blocks. Each block consists of a set of transactions. Hashing is used as a mechanism to agree upon the validity of the present state of the blockchain. The cryptographic hash is first computed for the genesis block. For every block that follows afterward, the hash of the previous block and the set of transactions in the current block is used to compute the hash. With this property, changing of the input value of any block changes the hash value of that block and all the blocks that follow it. Also, as the blockchain uses a P2P network, each user on the network has a copy of the blockchain. Thus, making it computationally infeasible to tamper with any of the transactions present on the blockchain. Hashing acts as a backbone of the blockchain where the hash of the last block acts as proof of the boundless state of the blockchain.

2.2.2 Digital Signature

Blockchain uses a digital signature based on public key cryptography to provide authentication. Each user in the P2P network has a key pair (public key, private key) associated with it. The public key is published over the network when a transaction is initiated by the user and acts as an identity of the user on the network. The private key, as the name suggests, is stored privately in a digital wallet.
Whenever a transaction is being published to the blockchain. The sender digitally signs the transaction with his/her private key. The public key and the signature of the sender is used by the users on the network to verify the authenticity of the user. Figure 2 depicts how digital signatures work in the blockchain network.

![Figure 2: Digital Signature](image)

Figure 2 shows Alice sending a transaction to the blockchain network. She signs the transaction using her private key and publishes the transaction into the network. Bob who is also part of the network validates the authenticity of the transaction by verifying the message with Alice’s public key. Since the blockchain implements a peer-to-peer (P2P) network. The transaction needs to validated by every user on the network as done by Bob.

### 2.3 Unspent Transaction Outputs (UTXOs)

Unspent transaction outputs (UTXOs) is the output of a transaction that has not been spent. After every transaction, the input is deleted, and the output is updated
to the UTXOs. Every user refers to the UTXOs available to him/her from previous transactions to carry out future transactions. Analogous to the account balance in Banks, UTXOs are the unspent crypto coins left for each public key/user on the blockchain. However, unlike Banks which acts as a central authority in storing and maintaining the balance, a copy of UTXOs of every public key on the blockchain is available to every node on the network, thus, avoiding the single point of failure.

Each transaction on the blockchain can only use the unspent transaction output (UTXO) from previous transactions. The reason for enforcing this mechanism in cryptocurrencies is to avoid the problem of double-spending [16].

Double Spending can be defined as duplication of digital currency in order to spend it multiple times. UTXOs keeps track of the unspent amount for each transaction. If an attacker publishes multiple duplicate transactions simultaneously to the blockchain, only one of the transactions will be accepted and added into the blockchain with the help of proof-of-work, and the remaining transactions will be invalidated.

2.4 Transaction Fees

Transaction fees [18] are the crypto coins paid by the blockchain wallet in order to add their transaction into the blockchain. Transactions are validated and added into the block and eventually to the blockchain by miners through the process of mining. Initially, when the transactions are published into the blockchain network, they reside in a memory pool; from there miners pick up the transactions that they wish to include in their block. The way the miners prioritize the transactions is based on multiple factors such as transaction size and transaction fees.

For example, Bitcoin can add a block with a max size of 1 MB, so miners prioritize transactions with a smaller size as they are easier to validate. Also, picking smaller transactions enables them to include a large number of transactions into their block,
hence obtaining a larger transaction fee. In order to include transactions of larger size, the wallet has to pay a larger transaction fee.

Transaction fees are dynamically calculated by the wallet based on the current state of the network, the size of the transaction, etc. However, it also provides an option to customize our transaction fees.

2.5 Blockchain Risks

The most vulnerable area of the blockchain technology is actually outside the blockchain network. It is the endpoints or the nodes where the risks start, as nodes act as an interface for humans to access the blockchain network. Also, endpoints are used for mining as well as for proofreading the transaction on the blockchain. The most basic requirement for accessing a blockchain network is to have a key pair (public key, private key). While the public key is published into the network and used as a means to identify the user over the network, the private key needs to be stored privately in a wallet and needs to be used by the user to sign his/her transactions.

The private keys are mostly stored in digital wallets at the endpoints/computers in the network. If a hacker hacks into the computer and gains access to the wallet, there is no way for the blockchain to validate the user. As each account on the blockchain is identified by the key pair (private key, public key), whoever owns the private key has complete ownership of the account. Blockchain has no means of validating the change of ownership. One of the most important causes of the majority of the blockchain attacks is inefficient security and management of wallet keys [11].
CHAPTER 3

Escrow service

This chapter focusses on implementing a simple Escrow service on the blockchain and pointing out the limitations introduced by this approach. The approaches implemented in the later chapters overcomes the limitations introduced by this approach.

3.1 What is Escrow?

An Escrow service can be considered as a contract between the consumer and the provider. A trusted third-party (TTP) serves as an Escrow service to ensure that both the consumer and the provider fulfill the terms of their service. Suppose Alice demands a particular service from Bob within a particular period of time, and Bob charges a particular amount of money for the service. The Escrow service here acts as a middle man between the two parties. Alice and Bob register with the Escrow service. Alice transfers the service charge to the Escrow service along with the amount of time within which the service needs to be completed.

Once Bob completes the service, he notifies the Escrow service as well as Alice. The Escrow service then checks if the service was completed within a specified time; if yes, the money is transferred to Bob.

Figure 3 shows the communication between Alice (the consumer) and Bob (the provider) using an escrow service.

1. Alice transfers the service charge intended for Bob to the Escrow service.

2. (a) Bob notifies Alice regarding completion of the requested service.

   (b) Bob also notifies the Escrow service regarding the completion of the requested service.

3. Escrow service releases the funds sent by Alice to Bob.
3.2 Escrow On Blockchain

This approach implements the Escrow service as a trusted third party (TTP) on the blockchain. The Escrow service is implemented as a user in the blockchain network who listens for transactions involving an Escrow service on the blockchain network. In order to identify a transaction involving an Escrow service, a comment field is introduced at the transaction level. The comment field starts with the keyword Escrow and contains additional information such as the address and the amount of money that needs to be sent to the receiver along with the information regarding the transaction fee for the Escrow service.

Let’s see how a transaction between Alice (consumer) and Bob (provider) takes place with an Escrow service in the picture. Alice publishes a transaction addressing the Escrow service as shown in the Figure 4.

Alice has 133 SG coins. Figure 8 shows that Alice publishes a transaction to the Escrow service assigning 100 SG coins to the Escrow service and the remaining 33 SG...
Figure 4: Transaction from Alice to the Escrow service
coins to herself. The comment field indicates that this transaction involves an Escrow
service and provides additional details to the Escrow service that it should send 99
SG coins to Bob and can keep 1 SG coin to itself as the transaction fee. Once the
escrow service receives the transaction published by Alice, it holds the funds until it
receives a notification from Bob (provider) regarding the completion of the requested
service by Alice. The Escrow service releases the fund once the requested service is
completed by Bob as shown in the Figure 5.

Figure 5: Transaction from the Escrow service to Bob
Figure 5 shows that the Escrow service sends 99 SG coins to Bob and assigns 1
SG coin to itself as the transaction fee.

Some of the implicit ideas incorporated while implementing the above approach are:

- The Escrow service always charges a transaction fee of 1 SG coin per transaction.
• The change that is not accounted for in the transaction is always assigned to the miner who finds the proof the block containing the transaction.

• The unspent money in the transaction is always assigned back to the sender who is initiating the transaction.

• The Escrow creates a new key-pair for each user it is transacting with and stores it in its database to provide an additional level of security.

3.3 Limitations

The limitations of this approach are:

• The Escrow service is responsible for actively regulating the funds between Alice and Bob. So, if the Escrow service defrauds from its service, there is no way to ensure that the funds get transferred to Bob.

• The Escrow service could demand a higher transaction fees from Alice to transfer funds to Bob.
CHAPTER 4

Two-Factor Authentication with the Escrow service as a TTP

This chapter extends the approach discussed in the previous chapter to implement a centralized two-factor authentication (2FA) on the blockchain using Escrow service as a trusted third party and discuss the flaws introduced by this approach. The approaches implemented in the later chapters overcomes the limitations introduced by this approach.

4.1 How Two-Factor Authentication Works?

Passwords were used as a primary method of authentication before two-factor authentication (2FA) came into place. Passwords posed many security problems due to the inability to store and manage them efficiently. Some of the commonly known problems are the use of easily predictable, similar passwords for multiple accounts. Storing passwords as plaintext on Databases, PCs or mobile, etc. Thus making passwords susceptible to attacks such as Dictionary Attacks, Phishing, Shoulder Surfing, Spidering, etc. 2FA [12][13] is used these days to overcome the drawbacks with passwords. Two-factor authentication is used to provide an additional level of security by combining two of the below three combinations:

- Something you know
- Something you have
- Something you are

Figure 6 represents a scenario which combines something you know (username/password) and something you have (phone). The 2FA depicted in Figure 6 can be explained as:

1. The user tries to log in to an application using his/her username and password (e.g., banking application).
2. The application sends a request to Trusted Third Party (TTP) 2FA provider to generate a PIN.

3. A copy of the PIN is sent to the application.

4. A second copy of the PIN is sent to the user’s phone.

5. The user then enters the received PIN which is then validated by the application in order to provide access.

Two-factor authentication can be carried out in various ways. Some of the most commonly used ones are:

- SMS: The user receives a One Time Password (OTP) as a message after entering the username and password into an application. The application authenticates the user after entering the OTP. As explained in Figure 6.
• Authenticator app: Uses a Time-based One Time Password (TOTP) [13] for 2FA. Where the app will generate an OTP or a QR code that needs to be entered/scanned within the specified time window (e.g., Google Authenticator).

• Push-based 2FA: Notifies the user when a login attempt is made on his/her account/device. The user can then accept/deny the request. Apple often uses this approach with their devices.

4.2 Centralized Two-Factor Authentication

This approach focuses on implementing a centralized two-factor authentication (2FA) service on the blockchain by implementing an Escrow service as a trusted third party. The initial step is to implement the Escrow as a user in the blockchain, who listens to the blockchain for any transaction that involves the escrow service.

A flow diagram representing how two-factor authentication (2FA) works on the blockchain network with the Escrow service in the picture is shown in Figure 7.

The way the above communication works is, suppose Alice wants to send some money to Bob. Alice decides to use the Escrow service for the transaction; she registers herself with the Escrow service. She publishes her transaction to the mining network. The transaction contains the amount of money that Alice wants to send to the Escrow account, which should eventually go to Bob. Alice has included a comment field in her transaction indicating that she is using the Escrow service and the amount of money that she wishes to send to Bob along with information regarding the transaction fee for the Escrow service as shown in Figure 8:

The above transaction shows that Alice is sending 100 spartan gold (SG) coins to the Escrow and assigning the remaining coins that she is left with, i.e., 33 SG coins to herself. The Escrow keeps listening to the mining network for any transaction involving an Escrow service. If it finds one that Alice published, it sends a time-based
one-time password (TOTP) to Alice’s phone to make sure Alice indeed made the transaction. She enters the TOTP.

Once the Escrow account verifies the PIN, it creates a transaction transferring the funds to Bob and publishing it to the mining network as shown in the Figure 9. The transaction published by the escrow account to the mining network is shown in
the Figure 9:

```json
{ txDetails:
  { output:  
    { 'fTN7oS1Dct0befqNBV9fkr6RNhxsc0pAsy8ZXnUN0ac=': 1,
      '+P0x30vCqHq8r3WT1Gv5q1j+Qlfl48IV0uHfFwEqz28=': 99 },
   input:  
    "fTN7oS1Dct0befqNBV9fkr6RNhxsc0pAsy8ZXnUN0ac=' },
  timestamp: 1556347421240,
  sig:  
   "3f2496a4c9b1f3179f0a6b9bb772a95bde1e022d1de69aa384c971b09045f14c5cfc4db4493a4b65600d333504b6ebcb5fa7e19a8e844adbf095c4cf84e4c3081eae3bef5a03baa44626c6b66de2b0eacab370f7735cceb1593583335066e5fdded43c590c31b7f57e0affba0199782ef24df7f6941cb7df252b3e459d482991487dbf8b258987c8cc9d885b0f696919966bfe5b6f14e219c08bbbf5f0d462cc5910cad2a8e9cccb155e493fbb6290c287382bd5499f443ef9e765721c147d53a9c108ee8a38bad54b46030ed94c3aeef82a646a7eecd34dd08173e2edb54ff757345f086040c1354ede4b2911dc9218a38e775ec6c57429f4ca73d56beaca0d' 
} }
```

Figure 9: Escrow To Bob Transaction

Figure 9 shows that Escrow transferred 99 SG coins to Bob and assigned 1 SG coin to itself as the transaction fee.

A limitation that is still not overcome with the current approach is, even though escrow can verify Alice’s identity after a transaction is made from Alice to Escrow. However, if Alice’s credential (private key) gets stolen, there is no way of ensuring the money get transferred to the Escrow service. In order to ensure that we need to have a new mechanism of ensuring that the coins can be sent only to Escrow even if Alice’s account is compromised. We will be discussing our approach for overcoming this limitation in the following chapters.
CHAPTER 5

Earmarked UTXOs

We introduce a novel concept of earmarked UTXOs (EUTXOs). Earmark as defined by Merriam-Webster is, to designate (something, such as funds) for a specific user or owner. This feature enables the user/public key to lock his/her unspent transaction outputs (UTXOs) to be spendable only to a designated set of users.

An additional field called earmarked is implemented at the transaction level along with other fields such as input, output, timestamp, signature, and comment. Using this feature the user will be able to spend his/her EUTXOs only to the list of earmarked users.

Figure 22 represents a transaction involving an earmarked field. It shows a special case of EUTXOs discussed in more detail in Chapter 8.

```json
{ txDetails:
  { output:
    { '0aQ11RmzZACJ/vd2ae1Pd8w)pNo9vOQeS+D3EmR4gm=': 0,
    'J+23GWL/50+27C+RwbB/lusBzJwAXcUH/DtBA5JwpVcVu=': 400 },
  input: '0aQ11RmzZACJ/vd2ae1Pd8w)pNo9vOQeS+D3EmR4gm=',
  comment: '100 spartan gold fromj+23GWL/50+27C+RwbB/lusBzJwAXcUH/DtBA5JwpVcVu=toMBnIA0InW+9yl1cVotgeD61nrgF5WvBdEItCaJx/K/t4=',
  earmarked:
    [ 'p1vR640RaQ7+tBb03MeLdcdW18b1jmC30550Txj0F9I=',
      'Um9TQBlyCgoVwYyU8PVMwwQdBdR5sbUMVFWqJXy/PA=' ],
  timestamp: 1556357505692,
  sig: '8da81d9b6ed80a6b69e8f9e077940e41dc7f8b111a5a5c98a5e9e53f6f1b18175dc68663ee7665dc
0a4f8d805229ef4196e63f6a0438d42545444a636f735551946a06734cd342598283fb653234675052a666
8ba31c18396618f4d416f3e688006e636f26d03a9c13736e35a917d0220076e920f9fdd5f5a72eef1c7f02d16
5fc6f766f96a41fc0096a05f70eaa5f6bdfbfccf13d9cb453f89df99d9a679427ef04332a414649dc38517e9
5a5890f1852a1e5efdb93bb8a8b6c96dc8b87e56c459ab37713db3d4a4f444c98e3a6caef921b28b73070f75
fee832c4926d112a502ff1094533e117d392c5fbcccc0b76eda24eb777a4f95f28d778104' }
```

Figure 10: Earmarked Transaction

Figure 22 shows that Alice’s computer which owns two key pairs is making a transaction from one public key/address to the other. She is assigning the 400 SG coins that she owns from her temporary key to the primary key, earmarking her transaction to be spendable only to a designated set of public keys/users. Publishing this transaction to the blockchain enables Alice to lock her UTXOs to be spendable
only to the list of users specified in the **earmarked** field. The advantage with EUTXOs is that, even if Alice’s key gets compromised, the only way to spend the 400 SG coins is by sending it to the earmarked users. Thus, preventing the attackers from stealing the SG coins.
CHAPTER 6

Restricted Escrow

This approach implements the Escrow service as a trusted third party (TTP) on the blockchain using EUTXOs to overcome the limitations introduced by traditional Escrow service on the blockchain.

6.1 Restricted Escrow Approach

EUTXOs are introduced with the Escrow service on the blockchain to overcome the drawbacks of traditional Escrow services as discussed in Chapter 3. The approach discussed in this chapter is represented in the flow diagram Figure 11.

Suppose, Alice is the consumer and Bob is the provider. Alice has 400 SG coins and the key pair \( K_A \). The service charge that Alice needs to pay Bob once the required service is completed is 100 SG coins. Alice and Bob register themselves with the Escrow service. Figure 11 depicts a scenario where Bob was unable to complete the required service. The earmarked field introduced at the transaction level contains the plausible recipients of the 100 SG coins, such as Bob, Alice’s paper wallet.

The scenario represented in the Figure 11 can be explained as:

1. Alice publishes 100 SG coins to the escrow service containing the service charge that she needs to send Bob. Along with the comment field that contains the address of Bob (needed by the Escrow service) and the earmarked field containing the plausible recipients.

2. Escrow service receives the transaction.

3. Bob notifies the Escrow service that the requested service was not completed.

4. Escrow service scans through the Earmarked field for other plausible recipients other than Bob. In this case, the Escrow service sends the 100 SG coins to Alice’s Paper Wallet specified in the Earmarked field.
5. Alice’s paper wallet receives the transaction.

Let’s discuss the above scenario from the implementation perspective. Highlighted fields in the Figure 17 shows the initial UTXOs for Alice (133 SG coins), Bob (49 SG coins), and Alice’s Paper Wallet (0 SG coins).

Alice (customer) needs to send 100 SG coins to Bob (provider) for the service. Both Alice and Bob registers with the Escrow service. Alice creates a transaction assigning 101 SG coins to the Escrow service and 32 SG coins to herself as shown in the Figure 13. The pre-decided transaction fee for using the Escrow service in this case is 1 SG coin. The earmarked field in the transaction shows that the money is meant to be sent only to Bob or to Alice’s Paper Wallet.

If Bob (provider) notifies the Escrow service that the service was not completed. The Escrow service transfers the funds to Alice’s paper wallet address present in the
earmarked field as shown in the Figure 14. The escrow service assigns 1 SG coin to itself as the transaction fees and refunds 100 SG coins back to Alice’s Paper Wallet.

Figure 15 shows the updated UTXOs after the transactions. Alice’s current UTXO is 32 SG coins, Bob has a UTXO of 49 SG coins as before, Alice’s Paper Wallet has the refunded 100 SG coins as the current UTXO, and the Escrow service has a UTXO of 1 SG coin (transaction fee).
6.2 Advantages

The advantages of the current approach are:

- Even though the Escrow service acts as a third-party/mediator of the transactions, it does not have full control over the SG coins sent to it. Every transaction sent to it will involve an **earmarked** field, stating the SG coins can be spendable only to the users in the **earmarked** field. Even though the Escrow service acts as a TTP, its possible actions are restricted.

- If Alice’s private key gets compromised in the scenario depicted in Figure 11. She could still obtain the refund from the Escrow service as it is sent to her paper wallet.
CHAPTER 7
Blockchain As Escrow

Decentralized Escrow service is implemented using EUTXOs. In this approach, each user acts as their own Escrow.

7.1 Enforced Escrow Approach

This approach implements a decentralized version of the Escrow service on the blockchain. Each user acts as their own Escrow service. Each user has the main key and temporary key dynamically created each time a user wants to transact with another user on the blockchain. Suppose, Alice (customer) wants to send 100 SG coins to Bob (provider) for his service. Alice has a main key pair $K_{A\_Main}$, she generates a new key pair $K_{AB}$ for the transaction from Alice to Bob. Alice publishes the transaction of 100 SG coins to $K_{AB}$. The transaction is earmarked to Bob and signed with Alice’s main key $K_{A\_Main}$. Once Bob notifies Alice regarding the requested service completion, she sends the private key $K_{AB}$ to Bob, who can now retrieve the funds from the blockchain. Figure 16 shows the scenario discussed above.

Figure 16 provides a high-level flow diagram of the scenario discussed above and can be explained as:

1. Alice published 100 SG coins to the temporary key $K_{AB}$, earmarked for Bob. The transaction is signed with her main key $K_{A\_Main}$.
2. Bob (provider) notifies Alice (customer) that the requested service is completed.
3. Alice sends the private key $K_{AB}$ to Bob.
4. Bob retrieves the funds published to the blockchain using the key $K_{AB}$.

Let’s see how the transaction between Alice and Bob works at the implementation level. Figure 17 shows the current UTXO of Alice is 133 SG coins and for Bob it’s 49
Alice needs to send 100 SG coins to Bob. So, she generates a temporary key and publishes 100 SG coins to the newly created key and assigning the remaining balance of 33 SG coins to herself as shown in the Figure 18. The **earmarked** field contains Bob’s address, indicating that the 100 SG coins are only spendable to Bob. Once Bob completes the service that he needs to provide Alice, he notifies Alice of the completion of the service. Alice now sends the private key of the generated temporary key pair.

---

**Figure 16: Enforced Escrow**

**Figure 17: Initial UTXO**

---

```
Initial balances
{
  '7syTHdyxpxlPkV9w570VVeafhx8IspeB1YoFeTWC/0': 73,
  'TAsunp0gEurh2HQN7bbxjPPPKQYQzoUs/1g+mWr/l/pw': 133,
  'CuF9XPfQs952Z4fSzu4rP8vzIMj2HaesD+jKB8gms=': 0,
  't/a51hMuDu3d74uvLwDX7N++ouk8EntdMNPcJoG+59M=': 0,
  'QGdqbrLC0cZ1kaRMBahXEnjeaAQG5dWun+++ee2wp1=': 49,
  'BG6352CIvpb+0Q3XmMqRIt0qqY857L/KZo06G08M=': 16,
  'iSVv8wY/jxtcsY1Axy4vlBKnGfJvOtaUK4WxqIMHGE=': 5,
  '7fukj6jPcx/-zBTs4PZf8lZftBtBk4tBlyA/cN4=': 12,
  'KMRmwcIHHR7TvUWWh/BBq6Dv6ayCjLyRPPyM10GJE=': 0
}
```
Bob can now retrieve the service charge using the temporary key. As Bob is also the owner of the temporary private key, he publishes a transaction to the blockchain to assign the 100 SG coins to his main key as shown in the Figure 19. Once the 100 SG coins are assigned to Bob’s account (main key), he can discard the temporary key. The updated UTXO after Alice’s and Bob’s transaction as shown in the Figure 18 and Figure 19 is shown in the Figure 20. Alice’s current UTXO is 33 SG coins and Bob’s current UTXO is 149 SG coins.

7.2 Advantages

The advantages of this approach are:

- Every user acts as their own Escrow. As there is no TTP involved in the
transaction. The customer has full control over locking and releasing the funds.

- Since the transaction published by the customer to the blockchain is earmarked with the address of the provider. Customer cannot misuse the funds once it is published to the blockchain.

### 7.3 Limitations

The limitations of this approach are:

- The provider could provide lesser service than the service promised, once the customer publishes the transaction to the blockchain. As the fund published to the blockchain is earmarked for the provider.

- The customer could publish lesser service charge than the amount promised to the provider.
In this chapter, we will be discussing a novel approach to implementing a decentralized 2FA for blockchain. This approach implements EUTXOs to achieve decentralized 2FA on the blockchain. A simple flow diagram depicting the working of this approach is shown in the Figure 21.

![Figure 21: Decentralized 2FA Flow Diagram](image)

In the flow diagram, Alice’s computer is a node in the blockchain network, with 10,000 SG coins. She has two keys, a one-time temporary key to lock her funds in the blockchain and the main key to make all her future transactions. Alice locks/signs all her funds with the temporary key and publishes it to the blockchain addressing Alice’s main key. She creates an *earmarked* field as part of the transaction, earmarking Alice’s mobile and paper wallet. The *earmarked* field is immutable and cannot be changed once created. Once the transaction is in the blockchain, she discards the
temporary private key. Now, Alice’s main key owns the 10,000 SG coins, with an exception that the funds are spendable only to Alice’s Paper wallet or her mobile. In, this case, no matter how many transactions Alice’s computer makes in the future, it should always only go to Alice’s mobile and/or her Paper Wallet. Alice’s computer makes transactions to Alice’s Mobile as and when needed, specifying whom the coins should go to in the comment field.

The approach discussed is represented in the flow diagram shown in the Figure 21 and can be explained as below:

1. Alice’s computer has the key pairs, $K_{Temp}, K_{M,PW}$. She publishes a transaction to the blockchain assigning all her funds to $K_{M,PW}$. The transaction contains earmarked field with the address of Alice’s mobile and her paper wallet. The transaction is signed with the key $K_{Temp}$. The key $K_{Temp}$ is discarded after the transaction is published to the blockchain, hence locking the earmarked funds to be spendable only to Alice’s mobile and/or her paper wallet.

2. $K_{M,PW}$ receives the transaction.

3. $K_{M,PW}$ sends 100 SG coins to Alice’s mobile. Along with the transaction, there is an earmarked field specifying that the transaction should only go to Bob or Alice’s paper wallet. Also, there is a comment field specifying whom the money should go to once Alice’s mobile receives it.

4. Alice’s mobile ($K_M$) receives the transaction published to the blockchain.

5. Alice’s mobile confirms the transaction has been received. This communication happens off-chain (not on the blockchain network).

6. Alice’s mobile now sends 100 SG coins to Bob.
Figure 22 represents a transaction with the **earmarked** field.

```json
{ txDetails:
  output:
    'RGtBPIC/XEftYMEwsLMNKnxUxKtHNCN0DT1nn7pV6Co=': 33,
    '3zYzjBE90y7pu/gjCAYDBUohw/jSSpsGICuw6VhVY=': 100 },
  input: 'RGtBPIC/XEftYMEwsLMNKnxUxKtHNCN0DT1nn7pV6Co=',
  comment: '100 spartan gold fromRGtBPIC/XEftYMEwsLMNKnxUxKtHNCN0DT1nn7pV6Co=toDTRqL/uGyDPeSQX37kwp/HJVCe5/APJ3iUb8xnM+JYttkhk=',
  earmarked:
    ['3zYzjBE90y7pu/gjCAYDBUohw/jSSpsGICuw6VhVY=','/Odm/5/cuZq/K4tf6G6jojKpj2XTCpnhSPMEPZok4sAA='],
  timestamp: '5548tf791i3914,',
  sig: '6e54ba931ce912cede812d59a5达f94ef00c7833c9a4ed58a8c53ceee6f2f2c7c8e69a541691d600e164d82e605b3d65879c2fＨｂf66b1b5f1de9ee2ed7c23f033722a5f518a545f68d5a3300db53080ba5e239c10a7ca54e49a2c05adb1fa9a74a0df2f891517c98e59713618886f6387a938ce49d6571691e049a1ba2e3bd18a8cc87ccc890bee9abf6666b363093d8bfc2cc30a0b0933d561f7dced425acad39d95bea9e9e608f8c2a1f336328c73c14c439f4fdd41d61e96e6696006b27c133019e3de014598772c280e55174f4eeb7ea8eb3d319a27c54177dd7a99c1cc00aca6d30ba8ec141a469dfac4147f41d390b77770bc3939ad'}
```

Figure 22: Earmarked Transaction

Figure 22 represents a transaction, where the **input** field represents the address of the sender. The **output** field contains the address of the receiver along with the amount of SG coins that need to be transacted to the address. The **timestamp** field is used to timestamp the transaction. The **sig** fields show the transactions digitally signed by the sender with his/her private key. The **comment** field in the transaction is used to convey any additional information that needs to be sent to the receiver. The **earmarked** field is immutable once created and represents a list of public key/users that the sender can transact with.

Decentralized 2FA is introduced in this approach by introducing **earmarked** field (something you know) and multiple keys (something you know) for the user. Let’s examine how an end to end transaction between Alice to Bob works. Alice has a total of 400 SG coins. She needs to send 100 SG coins to Bob. Alice first publishes a transaction to the blockchain assigning all her SG coins to her main key as shown in Figure 23. The **earmarked** field in the transaction indicates that the SG coins are spendable only to Alice’s mobile or her paper wallet. The transaction is signed with the temporary key and discarded once the transaction is published into the blockchain.

Figure 24 shows a transaction from Alice’s computer to her mobile, where she
sent 100 SG coins to her mobile and assigns the remaining 300 SG coins to herself. She earmarks the transaction to be sent to Bob or her paper wallet. The transaction is signed with Alice’s main key. The input field shows Alice’s computer public key/address and the output field shows that she is assigning 100 SG coins to her mobile public key/address and the remaining 300 SG coins to herself. In our implementation, the remaining change or UTXO is always assigned back to the sender. The change that is not accounted for is assigned to the miner who finds the proof of the block containing the transaction.

The transaction contains an earmarked field showing the list of users Alice’s computer can transact with. In this case, two public keys (Bob and paper wallet). As the earmarked field is immutable once created, she can spend her SG coins only to the two earmarked public keys. The comment field specifies the address of Bob and the amount of money that needs to be sent to him. Alice’s mobile uses these details to create a transaction from Alice’s mobile to Bob as shown in Figure 25.

In the current approach, decentralized two-factor authentication. The two-factor authentication is achieved using the two of the something you have features such as:

- Your phone
Your computer

8.1 Advantages

- If Trudy (the attacker) compromises Alice’s computer (private key), the only way she could spend the SG coins is by either sending it to Alice’s mobile or her paper wallet. As the earmarked field is immutable once it’s created.
- Alice’s mobile always receives only the exact amount of SG coins that need to be transferred. If the Trudy (attacker) compromises Alice’s mobile (private key), Trudy could only send the funds to either Bob or Alice’s paper wallet.
- In case, Alice’s mobile (private key) gets compromised. Alice’s computer could send her money to the paper wallet address specified in the earmarked field.
8.2 Disadvantages

- If Trudy (the attacker) steals Alice’s main key and either her paper wallet or mobile key, then Trudy could get away with Alice’s SG coins.
- As Trudy cannot spend the SG coins even if she compromises Alice’s computer (private key), she could decide to do nothing with the money.
CHAPTER 9

Current Approach vs Key management

One of the major reasons for most of the cryptocurrency thefts [19] is inefficient key management, leading to private keys getting compromised. Effective management, control, and storage of private keys is a complex task with no clear solution to it being found. There are several key management strategies that have been proposed in several papers. The research in this area paced up after the Mt. Gox [17] incident in the year 2014, where 370 million dollar’s worth of bitcoin was stolen by compromising the private keys of their customers. Eskandari et al. [11] evaluates several key management strategies, in order to explain how these complex strategies, lack the true ability to achieve effective key management. A brief overview of the pros and cons of some of the strategies are as described below:

**Software wallets:** Storing the private keys in a PC or mobile, inside a file/wallet is one of the most common approaches used by Bitcoin. The major advantage of this technique is user convenience, as the user doesn’t have to go through the hassle of loading the private keys, digitally signing the transaction, etc. As it is performed internally by the software itself. However, this approach comes with a set of disadvantages as well. The keys stored locally on a device connected to the network is susceptible to attack by Malware specifically designed to hack into the system and compromise wallet keys, in order to gain access to the user’s digital currency. The device containing the wallet keys could get stolen, depriving the user of his/her funds, or the system may get corrupted or crash. Bitcoin users can also assign their UTXO (unspent transaction output) to a new key every time a transaction is being made, requiring an additional mechanism to store and manage newly created keys. Also, users could have a key pool with a set of keys, where the change is assigned to a key picked from the pool.
**Password-protected wallet:** In this approach, bitcoin users can encrypt their wallets with a key derived from the password of their choice. The approach wards off keys getting stolen due to loss of the device. But, the attacker could still compromise the wallet keys digitally through malware, keystroke logging, etc. Also, if the user forgets his/her password, the digital currencies associated with his wallet is unrecoverable.

**Offline key storage:** In this approach, the bitcoin users can store their keys offline on a Paper Wallet (e.g. QR Code) or on a USB drive. Keys are stored physically with no access to the network, making it safe from malware attacks. However, every time a transaction is made by the user, the private key should be read back into the device, using a barcode scanner or by connecting the USB drive to the system. Thus, making the keys again vulnerable to attacks.

**Air-gapped device for key storage and signature:** This is an extended form of offline key storage, where the offline secondary device is used not only for storing the private keys but also for creating and signing the transaction. Once the transaction is being generated, it can be imported to the device connected to the network to publish the transaction. An advantage of this approach is that the storage of keys and signing of transaction takes place on a secondary device. Thus, not exposing the private keys to the network. However, in order to import the transaction into the device connected to the network, there needs to be a mechanism that connects the two devices. Thus, making the secondary device susceptible to malware attack.

**Hosted wallet:** In this approach, a trusted-third-party (TTP) is used to manage the user’s private keys/account. The TTP will host a set of web services exposing its transactional functionalities, which can be available to the users using authentication mechanisms such as a password or two-factor authentication (2FA). TTP uses the transactional details provided by the user to create and sign the transaction with the
user’s private key. With TTP there comes the risk of a single point of failure as it is susceptible to attacks (analogous to banks).

We could conclude from our discussion above, that no matter how complex strategies are in place for key management, there can always be a backdoor available for the attacker to gain access to the private key. Moreover, it takes just a simple human error to make these strategies vulnerable to attacks. Thus, instead of finding a novel effective key management strategy, these approaches aim at formulating a novel strategy that bypasses the loopholes of key management to some extent by introducing the novel approach of *earmarked UTXOs* (Section 2.3).

The EUTXOs are immutable records published to the blockchain along with transactions. These immutable records consist of a list of users that the owner of the account can transact with. So, even if the private key of the user gets stolen, the only way the attacker could spend the stolen digital currency is by sending it to any of the addresses specified in the **earmarked** record for that user.
CHAPTER 10

Conclusion and Future Work

In this project, we introduced a novel approach of *Earmarked UTXOs* (EUTXOs) to ward off the cryptocurrency thefts occurring due to compromised private keys/wallets. EUTXOs provided a mechanism for users on the blockchain to lock their UTXOs to be spendable only to a designated set of accounts chosen by the user. The EUTXOs introduced in this project is validated by using them to implement three different services as shown below:

- **Restricted Escrow with EUTXOs** - Overcomes the limitations of traditional Escrow service.
- **Blockchain as Escrow with EUTXO’s** - Decentralized Escrow service where each user acts as their own Escrow.
- **EUTXO Based 2FA** - Implements decentralized 2FA on the blockchain using EUTXOs.

The first approach implements EUTXOs with the Escrow service as a TTP on the blockchain. This approach enables EUTXOs to restrict the amount of control the TTP have on funds of its clients. Escrow services can only have limited control over the funds between its clients as the transactions are always earmarked to be spendable to a particular list of accounts. Thus, preventing Escrow service from defrauding from its service, it also prevents the funds from getting stolen by compromising of TTP.

The second approach implements an Enforced Escrow, which can be considered as a decentralized Escrow. In this approach, the sender generates a temporary key pair for every new transaction and publishes the funds addressing the temporary public key to the blockchain network, earmarking the transaction with the account of the intended receiver. Once the transaction is published into the blockchain network, the sender who initiated the transaction transfers the temporary private key to the
receiver in order to retrieve the transaction from the blockchain. This approach enables every user on the blockchain network to act as its own Escrow as the sender could release the key to the intended receiver only when the receiver completes the requested service. Another advantage of this approach is that the sender cannot compromise the funds intended for the receiver once it is published to the blockchain as it is earmarked for the receiver.

The third approach implements a decentralized two-factor authentication (2FA) on the blockchain with EUTXOs. It achieves 2FA with EUTXOs and multiple keys. This approach allows the user to lock his/her entire funds by publishing a transaction to the blockchain assigning all their funds to the public key of a newly generated key pair, earmarking the transaction to be spendable only to the user’s mobile or user’s paper wallet. The user only transfers the required amount of money that needs to be sent to the receiver to the user’s mobile earmarking the transaction to be spendable only to the receiver’s account.

One possible direction for this project in the future is to replace the Escrow services used in one of the approaches with smart contracts by enforcing the conditions introduced by the Escrow service into a set of rules pre-defined in the smart contracts. The traceability and transparency of smart contracts make it a better approach than the Escrow service. Also, the rules and conditions enforced by the EUTXOs in the approaches can be encoded as part of smart contracts in an effort to make the approaches more modularized.
LIST OF REFERENCES

   https://bitcoin.org/bitcoin.pdf

   Shutdowns (February 27, 2018). Available at SSRN:
   https://ssrn.com/abstract=3131702 or http://dx.doi.org/10.2139/ssrn.3131702

[3] Xiaoqi Li, Peng Jiang, Ting Chen, Xiapu Luo, Qiaoyan Wen, A survey on the
   security of blockchain systems, Future Generation Computer Systems, 2017, ISSN
   0167-739X,

   "How to make secure email easier to use", in CHI, 2005.

   Adoption criteria in encrypted email", in CHI, 2006.

[6] Analysis: Hackers Liquidated 3.2 Million in Tokens From Cryptopia Hack,

   on Decentralized Digital Currencies," in IEEE Communications Surveys &
   Tutorials, vol. 18, no. 3, pp. 2084-2123, thirdquarter 2016. doi:
   10.1109/COMST.2016.2535718

   "SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies,"
   doi: 10.1109/SP.2015.14

[9] Licheng Wang, Xiaoying Shen, Jing Li, Jun Shao, Yixian Yang, Cryptographic
   primitives in blockchains, Journal of Network and Computer Applications,
   2018.11.003.

[10] Blockchain: What is Mining?,
    https://dev.to/damcosset/blockchain-what-is-mining-2eod


