VALUE TRANSFER PRIVACY REVIEW THROUGH CRYPTOCURRENCIES

Functional Specification

by
Tomas Coleman

Student ID: C00218923
Institute of Technology Carlow
Supervised by: Richard Butler
15th November 2019

Introduction

This manual will go through what functionality the wallet will provide, what it will look like, and how the users will interact with it.

I will be creating a cryptocurrency wallet that will allow users to send and receive value on the Monero blockchain. For the purpose of this project, it will only operate within the Monero Testnet, and not the Main net. The wallet's core function will be sending and receiving Monero value.

Project scope

The scope of the project is to create a Monero wallet that allows users to send and receive Monero value. If time allows, it will allow users to view the blockchain and download it so that they can run their own node. They can send and receive value to other users of the cryptocurrency. The deadline for this project is the 20th of April 2020 at 23:59.

The user group for this project is both technical and non-technical users that want to have control of their money and ensure that it is as private as possible.

Requirements

Coding language

The program language that I will be using is Python as it is open source and has parts of the RPC implemented on Git-hub, so this will allow this project to be completed in the given time scope (GitHub. (2019).

Operating System

The operating system I will be using to code the program is both Linux mint for persistent storage and tails for non-persistent. Tails and Whonix will be used to run the program. This is a Linux virtual machine running inside a hypervisor that is also running a Linux distribution.

Device

For the purpose of this project, the device that will be used to demonstrate the program in use will be a desktop running Linux Mint. I understand that for security and privacy best practice, I should be using Whonix. However, my hardware does not allow this.

Monero Blockchain

I need to make use of the blockchain as this is used to send and receive value. If time allows, I will create a part of the program that will download a copy of the blockchain so that the users can run their own node.

Libraries

The libraries that I will be using are on GitHub. RPC calls are included in the Monero eco-system (GitHub. (2019). There is a website that has documentation for Monero-python (Monero-python.readthedocs.io, 2019).

Deliverables

When the project is complete, users will be able to register and authenticate with the program. Only upon successful authentication will a user be able to send and receive value. Users will also be able to download a copy of the Monero blockchain so that they can run their own node to help ensure the security of the Monero blockchain.

Assumptions

- That Monero is still a used cryptocurrency when the wallet is completed.
- That Monero technology will not change massively or in a way that will require changes to the code.
- The wallet will run on the Monero testnet during development and testing. That way we will not lose any coins of value.

Risks

There may be undiscovered bugs that will allow people to gain access to a user's wallet.

Solution Overview

To create a wallet that allows users to send and to receive value, and if time allows, to allow user to create a node on the Monero blockchain.

Requirement Specifications

- Users will be able to register with the program.
- Users will be able to log in.
- Users will be able to log out.
- The product will read value at an address of the users choosing on the Monero testnet if the user has access to that address.
- The product will transfer value from one address to another of the users choosing.
- The product will set the fee value for the transfer of the users choosing.

System configurations

The steps for setting up the programs will consist of:

- Instructions for the setting up of Whonix OS.
- A how-to/read-me manual that will provide detailed instructions for users about how to launch and use the program.

Non-functional Requirements

This refers to functionality that the application may be able to do if time allows, but is not critical to the workings of the application. This includes;

- A pleasant looking graphical user interface (GUI).

Secondary Functionality

This refers to functionality that can be added to the application if time allows. This includes;

- Providing password reset capability to users.

Exception Handling

The user will be informed if they do any of there

- If the wrong address has been entered the user will get a warning and it will fail to send the value to that address.
- Register with a name that has already been registered.
- Use a password that is deemed too weak.
- Login in with the wrong info.

Metrics

FURPS

Functionality
The project will be capable of sending value and receiving value on the Monero testnet. The wallet will require a login using a username and a password. The wallet will only be for one user. It will not be reusable for two people. They will need to run it on a different machine so that the programs do not interact with each other.

Usability
I want the program to have a GUI. This means that the user does not have to have an understanding of how to use a command line interface (CLI). I would like the program to a have nice looking GUI but this is not a functional requirement. The program will be consistent with the language of Monero. The documentation will be handled by a readme file that will be supplied along with the program.

Reliability
The program will need to have reasonable failure resilience as with any program, but primarily, the program will need to fail to send value if an invalid address is entered.

Performance
The wallet does not have to be super-fast as the Monero blockchain will be the limiting factor, but the wallet should not take up a large amount of resources in order to allow it to be used by people that are not able to invest in new and expensive hardware.

Supportability
The program will follow set out programming designs so that other people are able to edit it. This will allow the program to be adaptable and will increase the repair speed should any issues arise.

Precedent for this project

I have discovered that there are existing libraries to create the wallet. I discovered these during the Research Manual after I had decided that this is the logical route I wanted this project to go down. This official project has been ongoing since 13th of Feb 2018. I am unable to find a wallet that is written in Python. The majority of wallets I have found are web wallets and work using JavaScript.

Bibliography

GitHub. (2019). *Monero-ecosystem/Monero-python*. [online] Available at: https://github.com/Monero-ecosystem/Monero-python/releases?after=v0.4.2 [Accessed 13 Nov. 2019].

Monero-python.readthedocs.io. (2019). *Python module for Monero — Monero Python module 0.6.1 documentation*. [online] Available at: https://Monero-python.readthedocs.io/en/latest/ [Accessed 14 Nov. 2019].