Usability of Cryptocurrency Wallets Providing CoinJoin Transactions

Simin Ghesmati
SBA research, Vienna, Austria
Vienna University of technology
sghesmati@sba-research.org

Walid Fdhila
SBA research, Vienna, Austria
University of Vienna, Vienna, Austria
wfdhila@sba-research.org

Edgar Weippl
SBA research, Vienna, Austria
University of Vienna, Vienna, Austria
eweippl@sba-research.org

***Abstract*—Over the past years, the interest in Blockchain technology and its applications has tremendously increased. This increase of interest was however accompanied by serious threats that raised concerns over user data privacy. Prominent examples include transaction traceability and identification of senders, receivers, and transaction amounts. This resulted in a multitude of privacy-preserving techniques that offer different guarantees in terms of trust, decentralization, and traceability. CoinJoin [19] is one of the promising techniques that adopts a decentralized approach to achieve privacy on the Unspent Transaction Output (UTXO) based blockchain. Despite the advantages of such a technique in obfuscating user transaction data, making them usable to common users requires considerable development and integration efforts. This paper provides a comprehensive usability study of three main Bitcoin wallets that integrate the CoinJoin technique, i.e., Joinmarket, Wasabi, and Samourai. The evaluation includes usability and fundamental design criteria to find the ease of use of these wallets based on cognitive walkthrough during coin mixing. The comparison of the wallets with respect to usability and privacy criteria can be used for future evaluation of privacy wallets. The finding of this study can provide better insights for UTXO-based wallet developers.**

I. Introduction

Over the last decade, a lot of attention has been paid to blockchain technology. Beyond the hype, this interest is fueled by its intrinsic properties and unique conceptual design. Since its inception in 2008 by Satoshi Nakamoto [22], and unlike traditional systems that rely on centralized entities, blockchain technology uses a distributed shared ledger to permanently record transactions. In particular, in open blockchains such as Bitcoin, anyone can join, validate, and access the history of all transactions since the genesis block. Although in principle, this is supposed to be one of the key characteristics of blockchain technology, such transparency can put the financial privacy of users at risk. This comes from the fact that all transaction details in Bitcoin are visible to everyone in unencrypted form. Such details include but are not limited to sender and recipient addresses as well as the exchanged amounts.

Despite the use of pseudonymous identities in the form of public keys, it is still possible for an adversary to undermine the privacy of users. While a single transaction reveals very little information, literature [20], [25], [13], [5], [15] has shown that linking multiple transactions together can expose users' actual identities, interactions, and financial data. Having such information exposed can, in turn, lead to undesirable consequences; e.g., attract criminals, motivate extortion or discrimination, and benefit competitors.

To overcome the privacy issue in the Bitcoin blockchain, several mixing techniques have been proposed to mitigate the traceability of the users. One of this promising techniques is CoinJoin, which was integrated into different privacy wallets. In an analysis of the anonymity market, Möser and Böhme [21] found a notable turnover (8 million USD) during eight months from Jun 2015 to Jan 2016 in JoinMarket [14]. Dumplings [23], a platform for CoinJoin statistics, has shown a significant increase in CoinJoin transactions since 2018. The platform indicates bitcoins that get to CoinJoin in Wasabi, Samourai and other CoinJoin applications (including JoinMarket) where most of the CoinJoin transactions are created by other applications and Wasabi has the second rank. Stockinger et al. [27] extracted the CoinJoin transactions created by Wasabi and Samourai. They found that in total, Wasabi and Samourai have been used to mix 177,291.66 BTC and 13,485.45 BTC, respectively. The blockchain analysis [23], [21] indicates that CoinJoin based techniques are the most used privacy-preserving techniques for coin mixing in practice. For this purpose, in this paper, we solely focus on the CoinJoin technique. The platform indicates the increment of CoinJoin transactions created by Wasabi, Samourai, and other CoinJoin wallets including Joinmarket. The mass adoption of such privacy wallets, however, also requires addressing usability issues for both technical and non-technical users. Considering that an unusable system can not attract more users, and therefore can not achieve much anonymity [4]. Ease of use is one of the necessary factors that indicates how users accept sophisticated technologies such as mixing. Moreover, usability of privacy wallets is important, since user errors can cause irreversible privacy compromises.

In this work, we aim to investigate the usability of Bitcoin privacy wallets that support CoinJoin transactions. Therefore, we first review the main Bitcoin CoinJoin wallets (i.e., JoinMarket [14], Wasabi [30], and Samourai [26]). To the best of our knowledge, those are the only ones currently supporting CoinJoin transactions. Note that other wallets that previously supported the CoinJoin technique are not considered in this study as they either disabled CoinJoin transactions or the corresponding projects were completely abandoned [9]. Then, we perform a cognitive walkthrough based on insights from experts in the area of blockchain security and privacy research to evaluate the usability of these wallets. Additionally, we discuss usability issues and important features that should be provided by privacy wallets.

In this paper, we mainly focus on the usability aspects

of privacy wallets. However, a thorough evaluation of mixing techniques (e.g., CoinJoin) from security and privacy perspectives can be found in [9]. The main contributions of this paper are as follows:

- A cognitive walkthrough is conducted to identify usability issues during coin mixing and suggestions for usability improvements are provided. Learnability and errors were adopted as usability criteria [11], and fundamental design criteria [17] were investigated during the learnability walkthrough.
- Three CoinJoin-based Bitcoin privacy wallets are reviewed and compared with respect to nine usability and privacy criteria including portability, multi-wallet support, direct send, untraceability, preventing address reuse, anonymity set, CoinJoin creation time, CoinJoin amount, and CoinJoin fee, which can be used for future evaluation of privacy wallets.

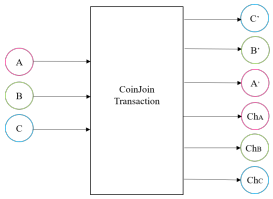
The remainder of the paper is structured as follows: Section II introduces the main concepts and reviews the CoinJoin wallets. Section III discusses the methodology and the evaluation criteria, while Section IV evaluates the usability of the wallets according to predefined criteria. Section V compares the wallets based on the usability and privacy criteria and outlines the discussion. In section VI related works are provided. Section VII concludes the work and summarizes the challenges.

II. Background

Bitcoin. Bitcoin as a peer-to-peer (P2P) electronic cash system was proposed by Satoshi Nakamoto [22] in late 2008 and developed in January 2009. Bitcoin uses asymmetric cryptography through a combination of a public and a private key. In most cases, Bitcoin addresses correspond to the hash of the public keys, and the bitcoins associated with an address can only be unlocked by the corresponding secret key.

In Bitcoin, a transaction is a statement for transferring coins from input addresses to output addresses [2]. The sender uses her unspent transaction output (UTXO) associated with her address as an input to the transaction with the recipient, whose address represents the transaction output. If the sender holds more coins than she wants to spend, she should provide a fresh address called "change address" to get the remaining coins. This change address is also considered as an output of the transaction. To include a transaction in a block, a transaction fee should be paid to a miner. Users utilize Bitcoin wallets to manage the keys and addresses where they can create and sign the transactions [1]. In most wallets, users create addresses to receive the coins and fund the wallets. To spend the coins, users send the coins to their desired addresses by specifying the amount that should be sent.

All transactions are publicly available on the blockchain putting users' anonymity at risk. Anyone can apply specific heuristics and auxiliary information (e.g., address tags) to cluster and identify users and their transactions. This tends to correlate Bitcoin addresses to real identities. One of the prominent heuristics, namely "common input ownership" combines all the input addresses to one user [20], which then effectively works on the address clustering. This heuristic assumes that multiple inputs of the transaction are controlled by the same



Ch_A, Ch_B, Ch_C: Change addresses.

Fig. 1. CoinJoin

user [22], as the coins associated with an address can only be redeemed by providing the corresponding signature. To diminish privacy issues in Bitcoin, and more specifically to break the "common input ownership" heuristic, several mixing techniques have been proposed. These mixing techniques seek to hide the relationships between input and output addresses in the Bitcoin transaction. CoinJoin is one of the first mixing techniques introduced in the Bitcoin forum, to prevent tracking users' transactions.

CoinJoin. CoinJoin [19] is a joint transaction among Bitcoin users to hide the relationship of the sender and recipient addresses. In Bitcoin, each input should be signed by the corresponding key independently from other inputs. This property makes a novel form of transactions in Bitcoin in which users can provide a set of inputs (A, B, and C) and outputs (A', B', and C') to create a transaction. The users are able to provide their change addresses (Ch_A, Ch_B and Ch_C) to get the remainder of the coins back (Fig.1). All the users should spend the same amount of coins, otherwise, the values in inputs and outputs can reveal the relationships. Once the transaction is created, the users separately sign the transaction, and one of them posts the transaction to the network. Fig.1 indicates how a CoinJoin transaction looks like.

CoinJoin wallets. In what follows, we review the CoinJoin wallets, and explain how they implement the CoinJoin technique.

JoinMarket wallet. JoinMarket [14] is a desktop wallet, it applies a taker-maker model to create CoinJoin transactions. A taker broadcasts her willingness to create a CoinJoin transaction on the Internet Relay Chat (IRC) messaging channel (i.e., specifying the amount, the fee, and the number of counterparties [the input peers]). The makers listening to the IRC send their participation confirmations to the taker including fees. The taker creates the transaction with the desired CoinJoin amount and sends it to the makers for signing. Due to insufficient liquidity in JoinMarket, finding a large number of peers to create CoinJoin transactions can be a difficult task. Besides, IRC cannot handle the participation of a big number of makers (e.g., 50) [10]. As the taker is the one who creates the CoinJoin transaction, she can put the desired recipient address among the outputs without the makers knowing to which input the output is related (unless the transaction is created with one counterparty). Thus, in JoinMarket, it is possible to send directly to the desired recipient address. In other wallets, the users first send the mixed coins to their own addresses and then create a new transaction to send the coins to the desired destination address.

Wasabi wallet. Wasabi [30] is a desktop wallet which uses coordinator to create CoinJoin transactions. Chaumian

CoinJoin [8] adopted to blindly signed [3] the outputs such that the coordinator can not map inputs to outputs. In Wasabi, CoinJoin is created in three main phases: (i) input registration, (ii) output registration, and (iii) signing. The users register their inputs by sending the UTXO, the proof of the UTXO ownership, and the change address to get the remainder, and their blinded output to the coordinator to prevent correlating inputs to outputs. Afterwards, the coordinator verifies that the inputs, i.e. the UTXOs, include enough funds and have not yet been spent, signs the blinded output and sends each of the outputs back to the senders. In the output registration phase, the senders unblind and send their outputs to the coordinator. If the coordinator finds his signature on the output, he creates a CoinJoin transaction with all the registered UTXOs as inputs and all the registered outputs and change addresses as the outputs of the transaction. In the signing phase, the coordinator sends the transaction for signing the inputs by the corresponding users, collects all transactions, combines the signatures, and broadcasts the transaction to the network [8].

The Wasabi application has a CoinJoin tab where the user can select the coins to be mixed and register them into the Wasabi pool. At the time of writing, there is only one pool with a pre-specified amount (0.104 BTC on the mainnet). The CoinJoin is created if certain inputs are registered (100 peers) or the time interval is achieved (one hour). Upon broadcasting the CoinJoin transaction, the mixed coins with their associated anonymity set are listed in the "CoinJoin" and "Send" tabs, where the user can spend them.

Samourai wallet. Samourai [26] is a mobile wallet currently released as an Android application. It also creates CoinJoin by a coordinator using Chaumian CoinJoin under the name "Whirlpool". At the time of writing, four pools (0.001 BTC, 0.01 BTC, 0.05 BTC, and 0.5 BTC) can be joined to create CoinJoin transactions. There is a flat fee rate for the pools which are indicated in table I. Users register their coins to one of the pools and wait for the required peers to create a CoinJoin transaction. In Samourai, the coins are first split into the selected pool amount in transaction 0 (TX0). These UTXOs are not mixed yet and are considered as pre-mix UTXO, they are listed in the pre-mix wallet. These UTXOs are registered to a coordinator, which will create the CoinJoin transaction for the selected pool. Once the CoinJoin is created, the mixed UTXO appears in the post-mix wallet, which is different from the main wallet. The Samourai application includes different wallets: main wallet, pre-mix wallet, and post-mix wallet. The user can send the mixed coins to her desired address using the post-mix wallet.

III. Methodology & Evaluation Criteria

Our study uses a cognitive walkthrough following the methods in [31], [6]. The former conducted the walkthrough for the PGP application and then a user study, and the latter performed a cognitive walkthrough with two experts in six Bitcoin clients. In a cognitive walkthrough, the expert evaluates the learnability of the interface and considers how novice users may pass or fail in conducting the tasks while they are using the interface. The experts try to identify possible errors or confusions for novice users [31].

In this research, we evaluate whether the CoinJoin technique implemented by wallets (JoinMarket, Wasabi, and Samourai) can be successfully used by Bitcoin users to achieve effective privacy on the Bitcoin blockchain. To evaluate the wallets we conduct a test scenario in which the experts should mix their coins with the CoinJoin wallets and send the coins to their desired destination addresses. The tasks were performed by two experts from the area of blockchain security and privacy research in different operating systems including Linux and Windows for JoinMarket and Wasabi, and Android for Samourai. The tasks that should be done are as follows:

- T.1 Installing the application.
- T.2 Generating a wallet.
- T.3 Funding the wallet, which includes creating a receive address and checking the balance.
- T.4 Performing a CoinJoin transaction.
- T.5 Transferring CoinJoin coins to the destination address.

The tasks are evaluated based on the following criteria:
Usability criteria. We adopted the following usability criteria from [11]:

- ***Learnability***: The ease of using the system to do a task in the first attempt.
- ***Errors***: The errors that the user makes during doing a task and the ease of recovery from those errors.

Fundamental design criteria. We adopted the following criteria from [17]:

- ***Visibility***: The user can clearly see the things (e.g., buttons, tabs) that she needs to interact with. The visibility of these things helps the user discover and use them.
- ***Feedback***: The user receives feedback whenever an action has been taken (e.g., hitting a button, clicking on a tab). The feedback is clear to prevent user confusion. If a problem is encountered, a clear notification should be provided.
- ***Constraints***: The interaction possibilities are limited to clearly show the user what can be done and prevent user confusion.
- ***Mapping***: The user can clearly understand the relationship between functions (e.g., buttons) and what happens when used. The terminology used in the interface is clear and understandable.
- ***Consistency***: The user can perform similar actions using similar elements to improve the learnability and memorability of the system.

IV. Cognitive walkthrough

The walkthrough on the three wallets was conducted by using the Bitcoin testnet. In the following, the versions and the operating systems which have been tested are provided. Each subsection explains one of the wallet walkthroughs based on the usability criteria.

A. JoinMarket Wallet

We tested JoinMarket version 0.8.2 on Ubuntu 20.04.2 LTS and Windows 10. Here we only focus on the usability of JoinMarket GUI (graphical user interface) also known as JoinMarket QT.

T.1 Installing the application.
Learnability. To install the wallet, the user should follow the instructions on the JoinMarket Github page. The wallet has several dependencies that take significant time to be installed (e.g., Bitcoin Core, Python 3). Selecting the appropriate assets based on OS to download may be confusing for a novice user (fails constraints). On Linux, once the package is downloaded and verified, the user needs to follow a quick start. By running install.sh, the installation starts interactively, following the command provided in the quick start page, and wallet scripts should be run. The user is informed about the Qt GUI, which can be selected during the installation.

The next part on the Github page directs the user to the "usage guide" page if she is new or otherwise to follow the "JoinMarket-QT walkthrough" page. On the usage page, it is stated that running the wallet script should quit with an error, as Bitcoin core configuration is required to use the wallet, which is probably one of the barriers of using this wallet. Configuration for Bitcoin core is provided in the documentation. There is also the "configuring JoinMarket" part in the installation page which then refers to the usage page. We suggest integrating all Bitcoin core configuration guides in one part and refer to that whenever is required. These separate instructions for configuration by referring to different parts are confusing (fails visibility). To use QT, the user should follow the instructions on the walkthrough which is slightly easier for novice users.

Running V.0.8.2 on Windows 10 leaves an error that is related to the problem of finding secp256k1 library. Thus, we had to use QT.exe. If the user downloads the .exe file via chrome, it suggests discarding it. If the user keeps the file and tries to open it, Windows prevents the app from running, which is unpleasant for a user who wants to use it as a wallet. It is better to inform Windows users about this in the installation guide and explain how they can verify the file. When QT runs for the first time, it quits with the Bitcoin core connection failure error. The user should configure Bitcoin core after the first running attempt, which is similar to the Linux configuration.
Errors. There is no categorization in the release page based on different OS, thus confusion about which are the proper files for the user's OS can occur.

T.2 Generating a wallet.

Learnability. In the first run of QT, the user gets informed to load or generate a wallet from the menu (achieves visibility). Hitting the generate button asks the user to enter a two-factor mnemonic recovery passphrase if she knows what it is (which is a bit technical), then the passphrase should be given two times (achieves constraints), and next the wallet name should be given, which has a default name. Then, the recovery words and seed phrase are shown, and the user gets informed to write them down (achieves feedback). A message showing that the wallet is generated informs the user about the task's success (achieves feedback). Once the wallet is generated, a message to restart Bitcoin core in the case of wallet recovery or wallet generation is shown. If the user presses OK, it directs to quit JoinMarket with yes and no options. If the user selects no, the wallet is loaded, while if she selects yes, JoinMarket will be closed. Loading without restarting may be confusing for the user if she considers the message that she previously received (fails mapping).
Errors. The wallet does not inform the user that the order of the recovery words is important, and it does not ask the user to enter the recovery words to be sure that the user has the correct memory of them.

T.3 Funding the wallet.
Learnability. In QT, There is not a "Receive" button similar to other wallets to create an address to receive bitcoin, the addresses are created in "mixdepths" that are not visible to the user, the user should click on the mixdepths to open them and see the addresses (fails visibility), then the user can copy one of the addresses and fund it. Once the address is funded, the new balance is updated in Joinmarket, however, no message is shown to inform the user (fails feedback). The current presentation of addresses according to the mixdepths is too technical for novice users.
Errors. The addresses are always shown on the JoinMarket wallet main page unless they are spent, which can not prevent address reuse. Address reuse is one of the prominent privacy issues in Bitcoin which can effectively relate the transactions belonging to one entity. The only mitigation of address reuse in JoinMarket wallet is that the addresses are indexed (e.g., deposit in red color), which is not a clear indication for the user to not reuse them. It is also possible that the funded address is copied again by mistake and then reused.

T.4 Performing a CoinJoin transaction.
Learnability. Due to the liquidity on the testnet, we tested a single join with one counterparty via JoinMarket. In QT (Fig.2), a user should open the "Coinjoins" tab (achieves visibility), and then the recipient address, number of counterparties, mixdepth, and the amount should be filled out. The mixdepth concept is a bit technical for novice users and in the current presentation in QT, the user does not get informed that she is not able to spend the coins from different mixdepths in one transaction. Hence, a clear guide would be helpful. While the CoinJoin transaction is broadcasted to IRC, the details of what is running are shown in a box at the bottom of QT. Some technical messages in the box cannot be easily understood by novice users (fails feedback).

If a user chooses to spend all the amount of a mixdepth, the value zero should be entered as the amount, which is not clear in QT (fails visibility). A maximum button that automatically fills out the amount with the maximum amount can help in this regard. Once the CoinJoin is created and broadcasted, the details can be found in the "TX History" tab.[1]
Errors. If the user chooses to spend the coins which have less than five confirmations, the transaction is aborted. The user has to read a long message which lists three reasons for aborting the transactions and the problem is not clearly specified (fails feedback). A clear error message can be helpful. However, it is better to check this condition before broadcasting the

[1]MultiJoin and taking a maker role to earn money to create CoinJoin transactions are also offered by JoinMarket, which are out of the scope of our test.

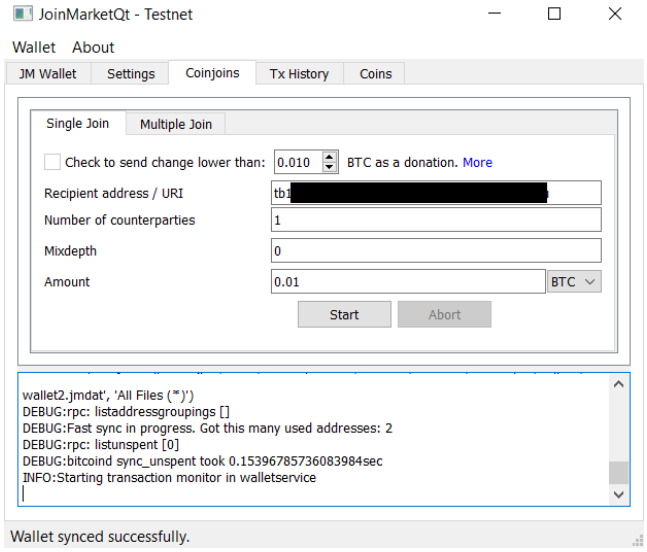


Fig. 2. JoinMarket CoinJoin

transaction to IRC to prevent the user from getting confused by the "Transaction is aborted" error.

In our first attempt, creating a CoinJoin failed with "error pushing = -26 min relay fee not met" which was not clear (fails feedback). Searching on the Internet, we found that increasing the transaction fee in configuration can solve the error. As JoinMarket does not provide a clear suggestion to solve the error, the user may fail to create a CoinJoin if she encounters such an error. Once a maker is found, JoinMarket asks the user to confirm performing the transaction which shows fees and the transaction details. If the user is not available during this time, she may eventually miss the CoinJoin creation. We suggest automatically confirming creating the transaction rather than asking the user to confirm it. Currently, transaction history in running QT on Windows does not contain the incoming transactions and it only lists the CoinJoin transactions that are created by the wallet, which may cause confusion in finding incoming transaction details (fails mapping).

T.5 Transferring CoinJoin coins. As a result of the direct send possibility, T.5 could be done during T.4.

B. Wasabi Wallet

We tested Wasabi wallet version 1.1.12.5 on Ubuntu 18.04.5 LTS, and Windows 10.

T.1 Installing the application.
Learnability. The download button is clearly visible on the website (achieves visibility) and the user can choose the package based on the OS (achieves constraints). A guide is provided, which indicates a step-by-step installation. The package is signed and verified on Windows and for other operating systems, the PGP should be verified.
Errors. The installation steps are quite clear and prevent critical errors by the user.

T.2 Generating a wallet.
Learnability. Wallet generation is opened when Wasabi is run for the first time (achieves constraints). The wallet can be generated by filling out a name and a password (achieves visibility). The user is warned that she is not able to recover her wallet without this password. The "show character" option helps the user see what she entered (leaving the password empty is also acceptable). On the next page, the twelve recovery words are shown. The user can generate the wallet by confirming that she has written the recovery words and password. Once the generate button is clicked, the page including the wallet name is shown (achieves mapping). Loading the wallet requires typing the credentials. The password box is located at the bottom of the page, if the user does not see the box and double-clicks on the wallet to load it the "Wrong password" message appears at the right bottom (achieves feedback), which can be replaced by "Enter the password".

A log is also available. Easy access to the folder containing all the files is provided (achieves feedback). The interface is simple and not overloaded with functionalities (achieves constraints), and the feature names are self-explanatory (achieves mapping). Moreover, notifications are highlighted with different colors, green for success and red for eventual problems (achieves feedback).
Errors. The user is informed that the wallet can be recovered by "your Recovery Words AND your Password" in a bullet format. We suggest adding "BOTH" before these two items to prevent any wrong interpretation of "AND" for non-technical users. A confirmation that the user has written down the recovery words and password is required to generate the wallet, however, we suggest asking the user to enter the recovery keys on the next page to be sure that she has a correct backup of recovery words. We also suggest to inform the user that the order of the recovery words is important. Currently, the wallet shows twelve recovery words in three columns, each column involves four words, and the order is based on the columns (the first four words are in the first column), while in some other wallets the order is based on the row (first three words are in the first row) which may get a careless user in trouble. If a user writes the words according to the rows and without paying attention to the numbering, then the wallet cannot be recovered.

T.3 Funding the wallet.
Learnability. In the first attempt to load the wallet, the user is forwarded to the "Receive" tab (achieves constraints), where she can generate an address by labeling it and then hitting the "Generate receive address" button (achieves visibility). By putting the cursor on the label box, "Who knows the address is yours?" E.g.: "Max, BitPay" is shown, it is not clear if this labeling is related to the party that sends the coins to this address (fails mapping). Therefore, a clear message is suggested. The created address is shown with its label. Double-clicks on the address copy the latter and show the message "Copied" (achieves feedback). By clicking the small triangle on the left, the QR code, public key, and key path appear. If the user clicks on the address or its label, these items are not shown (fails visibility). We suggest adding a new button "More info" to make it easier to find the address QR code and additional information. The QR code can also be indicated along with the address which makes it visible.

Once the address is funded, it disappears from the receive tab to prevent address reuse, and a message is shown at the bottom of the page (achieves feedback). The received coins can be seen in the history tab including the time of the transactions,

the amount, transaction ID, and specified label. Double-clicks on the row open a new tab that only adds the confirmation status and the block height to the information provided in the history list. The address that got funded is not shown in the transaction details.

In the current format, if the user funds several addresses, she has to copy the transaction ID and then use one of the blockchain explorers to see her address as the input or output of the transaction. When the user clicks on the transaction ID to copy it, the selected part contains only the characters that are located before the cursor, and the entire ID is not selected by simple double-clicks (fails consistency). We suggest copying the ID by double-clicking on that. Checking the incoming transaction can be performed via the history tab where the incoming transactions are shown in green and the outgoing are shown in red (achieves mapping).

Errors. Public key and the key path which are shown in the drop-down menu of the created address are too technical for novice users. We suggest adding "Address" and "Address QR code" tags to make it clear to prevent getting confused by the public key. The address disappears once it is funded, however, the user is not informed that she can check the transaction's status in history (fails feedback), and she may think that she lost her funds. An informing message on this page would be helpful. To check the transaction confirmations, the user has to click on the transaction in the history tab, then a new tab will be opened showing the transaction details, however, it is not updating. While this transaction history tab is opened, each time that the user clicks on the transaction in the history tab, she gets jumped to the previously opened transaction details with the previous information, thus the confirmation is outdated. The user should first close this tab and then go to the "History" tab and click again on the transaction to open the transaction detail. We suggest automatically updating the transaction details page.

T.4 Performing a CoinJoin transaction.

Learnability. CoinJoin transactions can be created via the "CoinJoin" tab (achieves visibility and mapping) (Fig.3), the user can see a list of coins with their labels and their associated privacy. The associated privacy of the coin is shown in different colors (red, yellow, or green), by putting the cursor on the dedicated privacy color, the anonymity set of the coin (the set that the coin is mixed and unidentifiable among that set) is shown (achieves feedback). The user should select the coins that she prefers to perform CoinJoin with and then enqueue the selected coins. This activity referred to input registration in a CoinJoin transaction. The user can specify the desired anonymity set by clicking on the target button (achieves visibility). Currently, three anonymity sets are shown as default (2, 21, or 50) which can be edited in the setting, however, the user does not get informed that she is able to change them. We suggest showing a message (e.g., when the user puts the cursor in the target button) informing her that the anonymity set can be changed in the setting.

To enqueue the coins the wallet's password should be entered and the "Enqueue Selected Coins" should be pressed (achieves mapping). By enqueueing the coin, a status column is added and shows "queued" in front of the selected coin (achieves feedback). Once the coin (transaction input) is registered, the status is changed to "registered". The user is able to see the number of registered peers at the bottom right of the page as well as the remaining time for input registration (achieves visibility). However, she can not get informed that the CoinJoin is created only if one of these conditions (minimum peers or minimum time) is achieved (fails mapping). The user should wait and leave the wallet open until the end of the CoinJoin rounds. When the required peers are registered, the status is changed to "Connection confirmed", "Output registered", "Signed", respectively (achieves feedback). Once a CoinJoin has created the mixed coins and the changes are listed in the "CoinJoin" tab. These coins are also listed in the "Send" tab where the user can transfer her coins to the desired address. Privacy (the anonymity set) associated with the coin and the cluster (labels) are shown in front of the coins in the "CoinJoin" and "Send" tabs (achieves visibility). Cluster shows how the coin can be traced in the blockchain by the labels that the user has provided, however, the concept of clustering is too technical for novice users.

At the time of writing, the CoinJoin amount in Wasabi has set to 0.104 BTC on mainnet and 0.0001 on testnet. If a user has a large number of coins or if she selects the larger anonymity set, she has to wait more to repeatedly create CoinJoin transactions by the wallet. This will be done automatically resulting in significant delays for large amounts or large anonymity sets. The user should not only wait for at least one confirmation for each transaction (almost ten minutes), which is clearly shown by a label in front of the coins (achieves feedback) but also for the minimum of peers that are required to create the next CoinJoin and if one of the peers leaves the wallet, the delay will be increased until enough peers have joined. If the user's internet or Tor connection are lost, or the user shuts down her computer during creating CoinJoin, the coin is banned for a specific time [29], which adds up to the delay. The user should wait for the expiration of the ban. The current ban message does not provide any specific reason for the user, and the user may get confused about the "banned" status meaning. The message only specifies that "The coordinator banned this coin from participation until specified time " (fails feedback). The time in the message also does not contain the time zone, which is suggested to be added. We also suggest providing the details of banning the coin to make it clear for the users.

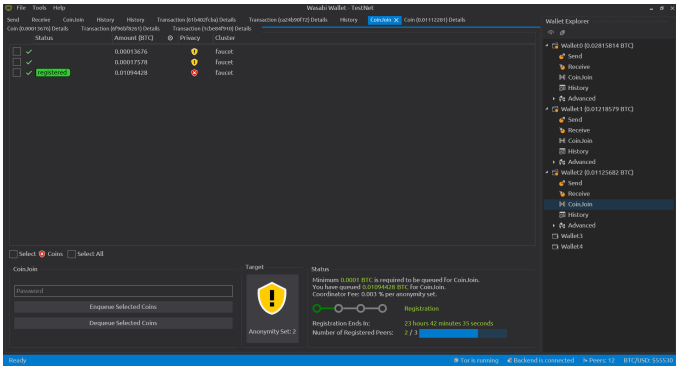


Fig. 3. Wasabi CoinJoin

Errors. The user can close the wallet during multiple rounds of CoinJoin (when a first-round CoinJoin is created and the next round is waiting for the transaction confirmation to start the

next round), which results in loss of the CoinJoin participation in the next rounds (fails feedback). Even if a user closes the wallet by mistake, no warning is shown. We suggest warning the user when she attempts to close the wallet during multiple rounds of CoinJoin. Currently, the user gets a warning if she closes the wallet after input registration and before signing the CoinJoin. In this case, the wallet asks her to be patient to finish the created CoinJoin transaction, and the user does not have any option to leave and close the wallet in this specific situation (achieves feedback).

T.5 Transferring CoinJoin coins.
Learnability. All coins including CoinJoin coins and non-CoinJoin coins are listed in the "Send" tab (achieves visibility). The user can select the coin that she wants to spend, enter the destination address, amount, label, and wallet password, and hit the "Send Transaction" button, which is completely easy to follow (achieves constraints and mapping). The "Max" button which shows the amount that can be spent considering the deduction of the transaction fee is really helpful, preventing the user from calculating the amount that should be entered if she wants to spend the entire amount of the selected coins. The user should fill out the label field which is related to the destination address, an informative message is suggested when the cursor is placed in the label field. The user gets informed once the transaction is broadcasted (achieves feedback).
Errors. If a user selects CoinJoin coins and non-CoinJoin coins for spending at the same time, the wallet warns "Merging unmixed coins with mixed coins undoes the mixes". The user can always ignore the warning and merge these coins.

If a user selects all her CoinJoin coins as inputs of a transaction, she can merge all of these coins in one transaction without any warning. Merging CoinJoin coins in one transaction results in losing privacy by "common input ownership" heuristic. A warning and a confirmation by the user are required in this scenario (fails feedback).

C. Samourai Wallet

We tested Samourai .apk package version 0.99.96f on Android 5.1.1, Android 10, and Bluestack.

T.1 Installing the application.
Learnability. The wallet is only developed on Android and can be installed via Android .apk package, Google Play, and F-Droid. All the installation packages are accessible in the downloads tab of Samourai website (achieves visibility). Currently, installation via .apk provides a choice of mainnet or testnet, and installing the wallet from Google Play only provides the wallet on mainnet without the possibility to change the network. The installation is simple, and the user just needs to hit the install button on Google Play, or download the .apk and install the package.
Errors. If the user installs the wallet via .apk file in the first attempt, she should select "testnet" or "mainnet", which may result in some problems for novice users who do not know the difference between testnet and mainnet (fails constraints). In the worst case, she could also send a testnet address to a malicious seller to fund her wallets. We suggest setting the default network to mainnet and provide changing the network to testnet via advanced options in the menu and warn the user that she is using the testnet.

T.2 Generating a wallet.
Learnability. To generate a wallet, the user should hit the create wallet button that is shown when the wallet is opened for the first time (achieves visibility and constraints). Then, a passphrase should be filled in two times. We suggest adding a "show character" icon to prevent any type errors. On the next page, the user should create a PIN code and then confirm it by re-entering the PIN. The last page indicates twelve recovery words, informing the user to write them down and keep them in a safe place. The user should confirm that she has already written down these recovery words and the passphrase to generate the wallet. We suggest adding the need for a passphrase for wallet recovery on the first page where the user should provide a passphrase. It is a little too late to inform the user that she also needs the passphrase for wallet recovery. Once the wallet is generated, the wallet main page appears (achieves mapping).
Errors. We suggest asking the user to enter twelve recovery words to be sure that the user has the correct recovery words. It would also be better to inform the user that the order of these recovery words is important. The current version may lead to critical problems for novice or careless users who may lose their funds forever.

T.3 Funding the wallet.
Learnability. To fund the wallet, the user should hit the plus button to see the wallet functions including "Receive" at the bottom right, which is not clearly visible on the main page (fails visibility). We suggest showing the functions in the plus button in the first attempt to make it easier for the user to find them. By hitting the "Receive" button, a page showing the address as a text and a QR code is shown (achieves mapping). Pressing the advance button enables the user to specify the requested amount, change the address type and leads to information about the key path. This solution is usable, since putting the information in the advanced section prevents novice users from getting confused by these advanced settings. To copy the address a message alerting the user that "If the address is copied, it may be visible to other applications" is shown and the user should hit "yes" to copy the address (achieves feedback). However, the message does not contain any solution for this alert. It could be mentioned that "you can use QR code scanning instead".

Once the address is funded, the balance is updated and the amount of incoming transactions is shown on the wallet main page (achieves feedback), and indicated in green color, which helps the user to figure out that this is an incoming transaction (the outgoing transactions are indicated in white color). Clicking on the amount shows the transaction details, including date, time, status (the number of confirmation), miner fee rate, miner fee paid, and transaction ID. Clicking on the icon on the top right directs the user to the Blockstream website where the user can check the transaction in the block explorer. The presentation of block explorer is not clear unless the user hits the icon (fails mapping). We suggest adding this along with other items to the transaction details with a clear tag such as "Checking transaction status".

To check the latest status of the transaction, the wallet main page should be refreshed by pulling down the page, however, the user does not get informed about this feature (fails visibility), a clearly visible refresh icon would help.

Errors. At the bottom of the transaction detail page, there is a "Boost transaction fee" button, by which the user can increase the fee to speed up the transaction confirmation, however, if the user stays on this page and then hits the button while the transaction got the confirmation, the error is returned "No value for address" which is not clear for the user (fails feedback). If the user refreshes the page, the button disappears and the confirmation status is shown.
The status in the transaction details shows the confirmations out of 3, however, if 3/3 is reached the status is still unconfirmed. 3/3 is confusing if four confirmations are required to consider a transaction as a confirmed one (fails mapping).

T.4 Performing a CoinJoin transaction.
Learnability. To create a CoinJoin transaction, the user should hit the plus button on the main page and select "Whirlpool". The name differs from what is currently used for the protocol which is called "CoinJoin". Therefore, it is not clear for the user if this item is used to create CoinJoin transactions (fails mapping). By selecting Whirlpool, a new page is opened, the user should again hit the Whirlpool icon on the bottom right. Two options are shown on the next page "Mix UTXOs" and "Spend Mixed UTXOs" (fails consistency). The term "UTXO" is also technical for novice users, and therefore should be replaced by "coins or bitcoin". The word "Mix" is the third terminology for one concept, considering the protocol name "CoinJoin", and the service name "Whirlpool". Avoiding different terminology for the same concept would help a lot. It is highly suggested to follow the terminology which has been adopted by the community for the protocols to make it easier for the users to understand the wallet functions.

By selecting "Mix UTXOs", the user is forwarded to a new page (Fig.4) where she can select the coins that she is preferring to do CoinJoin with (achieves constraints). On the next page, the cycle priority is shown in three options "low", "normal", and "high". "Cycle" is again a new term where it remains unanswered what it refers to (fails mapping). The user should select one of the listed pools (achieves visibility) (Fig. 4). The pools are enabled according to the amount that the user previously selected (achieves constraints). Thus, it is not possible to enter pools that are larger than the selected amount. The pool fee, miner fee, and the total fee are shown, and by pressing "Review Cycle", the details of the CoinJoin transaction are shown (achieves mapping). There are still some items that may be not clear for novice users (fails mapping), including "UTXOs created", which here means the number of new UTXOs or generally new coins (e.g., if a user selects 0.8 bitcoin and enters 0.1 pool, she receives 8 new UTXOs each of them contains 0.1 bitcoin). The other items are "Deterministic links", "Combinations", and "Entropy" which are technical terms without any further explanation. In the following, the fees, the change, and the amount to Whirlpool are shown and the user should hit the "begin cycle" button to join the pool (achieves visibility). The user is asked about "Doxxic change", she can choose the change as non-spendable to prevent being tagged. A message informs the user that even if she makes the change non-spendable, she can find the change in the list of unspent, however, it does not give any information where this unspent list is located (fails feedback), which is currently in the top-right menu in the wallet main page. By selecting yes and then refreshing the Whirlpool page, all the UTXOs are listed as "Unmixed" by showing the amount and "Mix 1/5-Queued" in front of each UTXO.

A new transaction is created on the wallet main page from which the amount selected to be mixed plus the fees are deducted from the wallet as an outgoing transaction. The first UTXO's status in the Whirlpool page changes to "Mix 1/5-Joined a mix" once it is joined to the pool. After six days on testnet, the status never changed, without any feedback on what the problem is (fails feedback). We tried to create other wallets to join the same pool in different devices, however, in all the wallets the status remained "Mix 1/5-Joined a mix". Therefore, once the coins are entered in the Whirlpool, the amount is transferred into the Whirlpool balance and is deducted from the main wallet balance (the same happens when the mixing is finished, the amount is transferred from Whirlpool balance to post-mix wallet balance). Checking different balances in different wallets may be confusing for the user, as she can not see her total balance (fails visibility), we suggest clearly showing all the different balances according to their wallets on the main page (e.g., main wallet balance, whirlpool balance, post-mix wallet balance, ...). Moreover, switching between the wallets is confusing (fails visibility). The Whirlpool can be reached from the bottom right plus button, and Post-mix can be reached by hitting the Samourai icon on the top left, which is not clear for the user (fails consistency). A clear way to access these wallets is suggested.

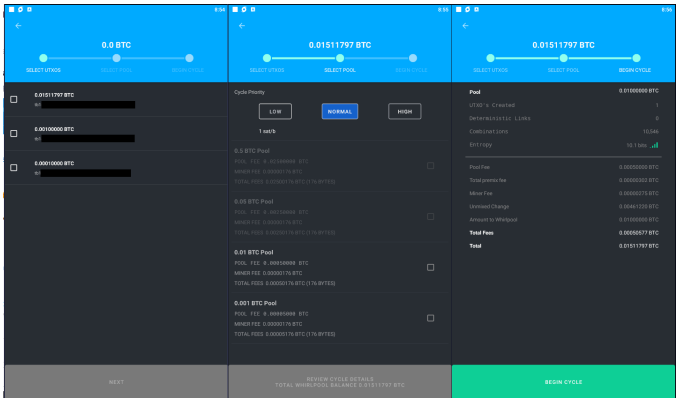


Fig. 4. Samourai CoinJoin

Errors. In our walkthrough on the testnet, selecting different cycle priorities did not change the amount which was shown under this option, selecting all the priorities showed 1 sat/b (fails mapping). Note that after six days the wallet did not list one of the UTXOs in the unmixed list, while the Whirlpool balance and Pre-mix balance showed the sum of the coins which included the hidden UTXO's amount. The bug should be fixed. A critical problem with Samourai was that we were not able to abort the CoinJoin and use our coins.

T.5 Transferring CoinJoin coins.
Learnability. This task could not be fulfilled as we could not receive CoinJoin coins. To spend the mixed coins, the user should go to the post-mix wallet by hitting the Samourai icon on the main page or the Whirlpool icon on the Whirlpool page and select "Spending mixed UTXOs" to be directed to the post-mix wallet, where the CoinJoin coins are received. Both options are not clearly visible (fails visibility). Then, the user can fill out the destination address, the amount and hit the "Review the transaction" button (achieves mapping).

The following is the description of sending the coins from Samourai main wallet, which is similar to spending the coins from the post-mix wallet (achieves consistency). When the user hits the transaction review and then taps the send button (achieves mapping). The transaction is created, signed, and broadcasted which are shown on the page (achieves feedback).[2]

Errors. On the send page, the user can select all the coins as the amount of transaction. Transaction fees are not deducted at this stage. On the next page, the fee is deducted based on the user-selected fee rate and the true amount that would be sent to the destination is shown as a message. If the user does not read the message carefully, she may think the sent amount is what was entered on the first page. We suggest deducting the minimum fee from the maximum amount in the first step.

V. Evaluation and Discussion

Table I compares the wallets over nine usability and privacy criteria.

Portability evaluates the possibility to use the wallet in different operating systems. JoinMarket and Wasabi provide support for different operating systems, while Samourai only supports Android.

Multi-wallet support evaluates whether the user can generate multiple wallets by installing the application on her device. JoinMarket and Wasabi are desktop wallets and the user is able to create as many wallets as she wants and save the wallets on her device, while in Samourai the user can generate only one wallet.

Direct send evaluates whether the user is able to directly send her UTXO in a CoinJoin transaction to the destination address. In Samourai and Wasabi, The UTXOs are mixed in a CoinJoin transaction and sent to the user's address and then the user can send mixed coins to her desired destination address, while in Joinmarket the user can directly send the UTXO to the destination address in the CoinJoin transaction.

Untraceability indicates that the relation of the inputs and the outputs of the transaction cannot be traced by other users. Considering external traceability in which the transaction is investigated by blockchain analysts, equal-size output CoinJoin cannot be easily traced. The number of peers and the multiple rounds of CoinJoin has a significant role in the external untraceability of the CoinJoin transactions, and are provided by the wallets. In terms of internal traceability between input peers, as Wasabi and Samourai wallets use Chaumian CoinJoin, the coordinator who creates the CoinJoin transaction is unable to trace the inputs and outputs. In JoinMarket, the CoinJoin transaction is created by the taker who pays the CoinJoin fee, thus the inputs and outputs are traceable by the taker and there is no privacy for the makers in that sense.

Preventing address reuse evaluates whether wallets prevent address reuse, which leads to transaction linkability. In Wasabi, the receive address disappears from the receive tab, once it got funded to prevent address reuse. Samourai shows only one receive address on the receive page and creates a new address whenever an address gets funded. In JoinMarket, the user can see all the addresses in their mixdepths. The wallets index addresses by "deposit" in red color to prevent address reuse.

Anonymity set per CoinJoin transaction evaluates the set of peers that are registered as input peers in a CoinJoin transaction. Wasabi can provide large anonymity sets because of the liquidity in its network (at the time of writing, up to 100). Currently, Samourai creates the CoinJoin pools with 5 peers. JoinMarket anonymity set can be set by the users, although it is confined by the liquidity on the network and IRC channel message handling.

CoinJoin creation time evaluates the minimum time in which one round CoinJoin can be created. Creating CoinJoin in JoinMarket and Samourai depends on the availability of other peers in the network., while Wasabi creates CoinJoin if the number of registered peers reaches 100 or the waiting time is achieved. Thus, the user is sure that the CoinJoin is created in 24 hours at the latest on Bitcoin mainnet.

CoinJoin amount evaluates the amount a user can register for CoinJoin. As can be seen in the table, there is no restriction on the amount in JoinMarket, and the user is not confined by a specified number of input peers, which can be set by the user. In Samourai there are specific pools with the corresponding amounts and in Wasabi only one pool is available with a specified amount.

CoinJoin fee evaluates the fee of a CoinJoin transaction. JoinMarket uses random fees to pay as CoinJoin fee to the makers, which was relatively small on the testnet (0.001% of the transaction amount). Wasabi takes 0.003% of the transaction per anonymity set. Samourai has a flat fee rate for its pool and the pool fees do not depend on the user UTXO amount, however, the transaction fee for transaction 0 should be paid beforehand to join the pool.

From the usability perspective, Wasabi has easy installation and is well documented. The documentation is structured with two different explanation levels: (i) for beginners and (ii) advanced. All steps and workflows are well described from the installation to the use of the features including intermediary steps, and best practices. The interface is user-friendly in comparison with the other wallets. The transaction can be created with quite large input peers (up to 100) and the user gets informed that she has the chance to create a CoinJoin transaction in a one-hour time frame. However, Wasabi creates too many small coins by creating CoinJoin transactions since the pool amount is set to a small amount and can not be changed by users. If the user wants to send large amounts to a destination address, she should either merge all the small coins which creates privacy problems by the so-called "common input ownership" heuristic or spend the coins one by one which requires creating too many transactions separately. One of the problems with Wasabi and Samourai is that if the change is less than the minimum pool amount, it is left in the wallet and should be merged with other coins to be eligible for a CoinJoin pool, while in JoinMarket the user is able to CoinJoin the entire amount.

JoinMarket's configuration is not easy for non-technical users, and creating CoinJoin cannot be easily done without reading the documentation and searching on the internet when an error occurs during the CoinJoin. Some errors do not give a clear indication of what should be done to be handled. However, it has some features that can not be found in Wasabi

[2]Privacy add-ons including "Ricochet: additional hops between wallet and destination", and "Cahoot: create on-demand CoinJoin" can be enabled while sending the coins. Each of them contains a description of its functionalities. In Cahoot's explanation, CoinJoin terminology is used, while previously the wallet used "Whirlpool" and "mix UTXOs" for creating CoinJoin, but no further information if all of them are using CoinJoin protocol (fails mapping). Different names make it unclear if they are applying the same protocol. The investigation of add-ons is out of the scope of the task.

Wallet	Portability	Multi wallets	Direct send	Untraceability	Preventing address reuse	Network	Anonymity set ⋆⋆	CJ† creation time	CJ amount	CJ fee
JoinMarket [14]	✓	✓	✓	✓±	††	testnet/ mainnet	Set by user (Current default: 9)	X±±	Set by user	Set by user (Random fees ∼0.001%)
Wasabi [30]	✓	✓		✓	✓×	testnet	3 peers	24 hours	0.0001 BTC	Coordination fee 0.003%⋆
						mainnet	100 peers	1 hour	∼0.104 BTC	Coordination fee 0.003%⋆
Samourai [26]				✓	✓×	testnet/ mainnet	5 peers	X±±	0.001 BTC	TX0 fee+Pool fee 0.00005BTC
									0.01 BTC	TX0 fee+Pool fee 0.0005BTC
									0.05 BTC	TX0 fee+Pool fee 0.0025BTC
									0.5 BTC	TX0 fee+Pool fee 0.025BTC

⋆⋆ Per CoinJoin transaction. † CoinJoin. ± Internal traceability by taker. †† Just indexing. × By disappearing. ⋆ Per anonymity set. ±± Depends on the liquidity.

TABLE I. EVALUATION OF COINJOIN WALLETS

and Samourai. It lets users modify the setting for the fees and the number of counterparties. Moreover, there are two important features in performing CoinJoin via JoinMarket; (i) the first one is the ability to specify the amount by the user without any need to enter a specific pool and be confined with the pool amount. Note that performing a CoinJoin for a large amount in JoinMarket is possible, which represents an advantage over the other wallets, although for large amounts there should be market makers accepting to create a CoinJoin with that amount. (ii) The second feature is to directly send the mixed coins to the destination address, instead of sending it to the user's own address and then creating another transaction to send the CoinJoin coins to the destination. Thus, creating CoinJoin with Joinmarket requires one transaction less in comparison to the other two wallets and consequently one transaction fee less.

Samourai provides a simple installation and using it as a normal wallet is satisfying. However, the wallet is only released for Android which affects portability. The wallet interface lacks visibility of the functions, and the function names differ from the terms commonly used in the community. Creating CoinJoin with Samourai is a little bit difficult and the user is not informed about the reason if the CoinJoin be stuck. It is also not satisfactory if the user cannot abort the CoinJoin and spend the coins in different transactions.

The main objective of the research was to evaluate the usability of CoinJoin wallets. The evaluation includes all steps required to use the wallet from the installation till the mix and transfer of the coins. For example, the complexity of installing JoinMarket can considerably decrease the latter adoption by novice users despite the unique features it provides for performing CoinJoin transactions. The results also show that despite the utilities offered by such CoinJoin wallets, it can be cumbersome for a novice user to correctly use their mixing services. Indeed, it is not only required that users have to be, to a certain extent, familiar with the protocol for creating CoinJoin transactions, but also cautious about undoing the mix by spending the CoinJoin UTXOs as the inputs of one transaction.

VI. RELATED WORK

Blockchain privacy from the user perspective has been studied in [16], [7], [18]. The studies indicate the lack of users' knowledge in privacy issues in blockchain and consequently, the users are not well informed why and how they should use privacy techniques to mitigate the risk of de-anonymization in the blockchain. Krombholz et al. [16] conducted a user study on Bitcoin security and privacy and found a serious misconception between users in privacy and being anonymous in the Bitcoin network. Fabian et al. [7] performed research on the user's perspective of Bitcoin anonymity. They found that almost 18% of users were not aware of the risk of deanonymizing the Blockchain, half of them were aware and concerned in some way, and the rest were aware of the risk but were not concerned. They also investigated the awareness of the user in mixing services, their result shows that half of the participants are not familiar with CoinJoin technique. Apart from the need to improve user's knowledge, the usability of implemented privacy techniques has a significant role in their adoptions in practice.

The usability research in key management [31], [6] performed a clear methodology to the usability study of a system where they defined specific tasks and conducted a cognitive walkthrough by experts to evaluate the learnability of the interface. Eskandari et al. [6] performed usability research in Bitcoin key management. They defined an evaluation framework and then performed a cognitive walkthrough to compare different key management approaches to specify whether they achieve or fail the usability criteria. Ljunggren [17] defined criteria on evaluating the top five Ethereum mobile wallets which are inspired by Norman [24] and conducted a user study to evaluate the wallets and then provided an application structure to improve the wallets based on their findings. The usability of the Zcash wallet was studied in [12]. It found that most of the users failed to purchase a real item using the wallet due to the complexity of the installation and integration of the wallet with the network-level protection tools. In [28], an analysis of the top five mobile cryptocurrency wallet reviews shows that UX shortcomings and users' misconceptions may cause serious errors and loss of funds. To our knowledge, this is the first study on usability of Bitcoin privacy wallets. Our cognitive walkthrough follows the methodology of [31], [6] and the usability criteria defined in [11], [17].

VII. CONCLUSION

Bitcoin is a publicly available database in which the users and their transactions can be deanonymized. Privacy-preserving techniques such as mixing protocols have been proposed to mitigate privacy issues in the blockchain.

This paper provided a cognitive walkthrough to evaluate the usability of three main CoinJoin wallets. Our results show that further improvements are required to make these wallets usable by common users. In particular, users who do not understand at least the main concepts of the CoinJoin technique might find some difficulty in mixing the coins via the applications and dealing with the error messages. They should also be aware to not undo mixing by merging them with other UTXOs. Designing user-friendly interfaces and providing proper notifications will help the user easily execute CoinJoin transactions. Future work consists of a user study that includes both technical and non-technical users.

Acknowledgments: This research is based upon work partially supported by (1) SBA Research (SBA-K1); SBA Research is a COMET Center within the COMET – Competence Centers for Excellent Technologies Programme and funded by BMK, BMDW, and the federal state of Vienna. The COMET Programme is managed by FFG. (2) the FFG ICT of the Future project 874019 dIdentity & dApps. (3) the FFG Basisprogramm Kleinprojekt 39019756 Decentralised Marketplace for Digital Identity.

REFERENCES

[1] A. M. Antonopoulos, *Mastering Bitcoin: Programming the open blockchain*. " O'Reilly Media, Inc.", 2017.

[2] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "Sok: Research perspectives and challenges for bitcoin and cryptocurrencies," in *2015 IEEE symposium on security and privacy*. IEEE, 2015, pp. 104–121.

[3] D. Chaum, "Blind signatures for untraceable payments," in *Advances in cryptology*. Springer, 1983, pp. 199–203.

[4] R. Dingledine, N. Mathewson, and P. Syverson, "Challenges in deploying low-latency anonymity," *NRL CHACS Report*, pp. 5540–625, 2005.

[5] D. Ermilov, M. Panov, and Y. Yanovich, "Automatic bitcoin address clustering," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2017, pp. 461–466.

[6] S. Eskandari, J. Clark, D. Barrera, and E. Stobert, "A first look at the usability of bitcoin key management," *arXiv preprint arXiv:1802.04351*, 2018.

[7] B. Fabian, T. Ermakova, and U. Sander, "Anonymity in bitcoin?–the users' perspective," 2016.

[8] A. Ficsor, "Zerolink: The bitcoin fungibility framework," *URL: https: //github.com/ nopara73/ZeroLink*, 2017.

[9] S. Ghesmati, W. Fdhila, and E. Weippl, "Sok: How private is bitcoin? classification and evaluation of bitcoin mixing techniques," *Cryptology ePrint Archive*, 2021.

[10] A. Gibson, "Joinmarket update for oct 2020," *URL: https: //joinmarket.me /blog /blog/oct-2020-update/*, 2020.

[11] N. N. Group, "Usability 101: Introduction to usability," *URL: https: //www.nngroup.com/a rticles/ usability-101-introduction-to-usability/*, 2012.

[12] H. Halpin, "Holistic privacy and usability of a cryptocurrency wallet," *arXiv preprint arXiv:2105.02793*, 2021.

[13] M. Harrigan and C. Fretter, "The unreasonable effectiveness of address clustering," in *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ ATC/ ScalCom/ CBDCom/ IoP/ SmartWorld)*. IEEE, 2016, pp. 368–373.

[14] Joinmarket, "Joinmarket," *URL: https://github.com/JoinMarket-Org/ joinmarket-clientserver*, 2015.

[15] M. Jourdan, S. Blandin, L. Wynter, and P. Deshpande, "Characterizing entities in the bitcoin blockchain," in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2018, pp. 55–62.

[16] K. Krombholz, A. Judmayer, M. Gusenbauer, and E. Weippl, "The other side of the coin: User experiences with bitcoin security and privacy," in *International conference on financial cryptography and data security*. Springer, 2016, pp. 555–580.

[17] N. Ljunggren, "Improving the usability of secure information storing within blockchain applications," 2019.

[18] A. Mai, K. Pfeffer, M. Gusenbauer, E. Weippl, and K. Krombholz, "User mental models of cryptocurrency systems—-a grounded theory approach," 2020.

[19] G. Maxwell, "Coinjoin: Bitcoin privacy for the real world, 2013," *URL: https:// bitcointalk. org/ index. php*, 2013.

[20] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, "A fistful of bitcoins: characterizing payments among men with no names," in *Proceedings of the 2013 conference on Internet measurement conference*, 2013, pp. 127–140.

[21] M. Möser and R. Böhme, "Join me on a market for anonymity," in *Workshop on Privacy in the Electronic Society*, 2016.

[22] S. Nakamoto, "A peer-to-peer electronic cash system," *URL: https:// bitcoin. org/ bitcoin. pdf*, 2008.

[23] Nopara73, "Dumplings," *URL: https://github.com/nopara73/Dumplings*, Last access 12 May 2021.

[24] D. Norman, "Psychopathology of everyday things," 2013.

[25] F. Reid and M. Harrigan, "An analysis of anonymity in the bitcoin system," in *Security and privacy in social networks*. Springer, 2013, pp. 197–223.

[26] Samourai, "Samourai wallet," *URL: https:// samouraiwallet.com/ whirlpool*, 2015.

[27] J. Stockinger, B. Haslhofer, P. Moreno-Sanchez, and M. Maffei, "Pinpointing and measuring wasabi and samourai coinjoins in the bitcoin ecosystem," *arXiv preprint arXiv:2109.10229*, 2021.

[28] A. Voskobojnikov, O. Wiese, M. Mehrabi Koushki, V. Roth, and K. Beznosov, "The u in crypto stands for usable: An empirical study of user experience with mobile cryptocurrency wallets," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–14.

[29] Wasabi, "Use of wasabi," *URL: https://docs.wasabiwallet.io/ FAQ/FAQ-UseWasabi.html how-can-i-mix-large-amounts*, 2018.

[30] ——, "Wasabiwallet," *URL: https://wasabiwallet.io/*, 2018.

[31] A. Whitten and J. D. Tygar, "Why johnny can't encrypt: A usability evaluation of pgp 5.0." in *USENIX Security Symposium*, vol. 348, 1999, pp. 169–184.