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 [22] 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. A cognitive walkthrough was conducted in order to evaluate the ease of use of these wallets using usability and fundamental design criteria. The study findings will enable privacy wallet developers to gain valuable insights into a better user experience.**

***Keywords*—Blockchain, privacy, Bitcoin, mixing, usability, wallet, Coinjoin, anonymity**

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 [26], 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 [24], [29], [15], [6], [18] has shown that linking multiple transactions together with off-chain information (e.g., forums, social networks) 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 concerns in Bitcoin, and mitigate user traceability, several mixing methods [22], [3], [23], [16], [35] were proposed. CoinJoin [22] stands out as one of the first promising techniques that were adopted and integrated within different privacy wallets. Indeed, previous blockchain analysis [27], [25], [31] showed that, in practice, CoinJoin based techniques are among the most used privacy-preserving methods for coin mixing. However, expanding further the adoption of such privacy wallets by both technical and non-technical users requires particular attention to the usability aspects and user experience. More specifically, in this context where the number of users is extremely important to achieve the desired level of anonymity [5], having an unusable system design may become an obstacle to this endeavor. Additionally, having an understandable, informative, intuitive, and user-friendly privacy wallet often ensures a better user journey and prevents actions associated with risks which may yield undesired and irreversible outcomes. Therefore, studying the ease of use of such wallets may be indicative of how users accept sophisticated technologies such as mixing.

In this work, we conduct a usability study on Bitcoin privacy wallets that support CoinJoin transactions. Specifically, we focus on three main Bitcoin wallets providing CoinJoin (i.e., JoinMarket [17], Wasabi [37], and Samourai [30]). At the time of writing, the latter are the main ones currently supporting CoinJoin transactions [10]. Other wallets that used to support CoinJoin are excluded from this study because either they no longer offer CoinJoin transactions or the corresponding projects were completely abandoned [10]. Next, we provide a cognitive walkthrough based on the opinions of two authors with expertise in blockchain security and privacy research. Additionally, we discuss usability issues and important features that should be provided by privacy wallets. We also conducted a small-size user study n=2 with two computer science experts to evaluate the task success and task completion time.

While a thorough evaluation of mixing techniques (e.g., CoinJoin) from security and privacy perspectives can be found in [10], this paper focuses on the usability of Bitcoin wallets, which support CoinJoin as follows:

- Three wallets are selected, reviewed, and compared based on wallet features (e.g., anonymity set and CoinJoin creation time).
- A cognitive walkthrough and a small-size n=2 user study

are conducted to identify usability issues in coin mixing and suggest improvements. The usability criteria include learnability, errors, and efficiency (only in the user study) [13], while the learnability walkthrough includes fundamental design criteria [20].

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

II. Background and Wallet Designs

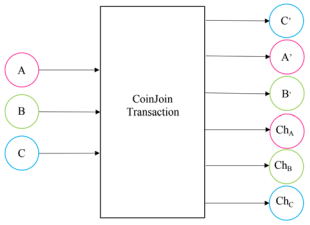
A. Basic Concepts

Bitcoin. Bitcoin as a peer-to-peer (P2P) electronic cash system was proposed by Satoshi Nakamoto [26] in late 2008 and developed in 2009. Bitcoin uses asymmetric cryptography through a combination of a public and a private key. Often, 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 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 used UTXOs correspond to more coins than what she wants to spend, a "change address" must be specified to receive the remaining coins. The latter is also considered as an output of the transaction. A transaction may include a miner fee as a reward to the miner.

Bitcoin wallet. A Bitcoin wallet, whether hardware or software, enables users to manage their cryptographic keys and addresses, and interact with blockchain to create and sign transactions [1]. Among other things, a wallet facilitates sending and receiving bitcoins to and from other users. A privacy wallet provides additional features (privacy-enhancing techniques) that improve user privacy.

Bitcoin Privacy. Bitcoin transactions are publicly available. Anyone can use specific heuristics or auxiliary information (e.g., address tags) to cluster transactions or possibly associate addresses with real identities. For example, one prominent heuristic called "common input ownership" associates all input addresses of a transaction with one user [24] assuming that they are controlled by the same user [26]. Thus, in order to hide the relationships between input and output addresses, several mixing techniques have been proposed. CoinJoin [22] is one of the first mixing techniques introduced in the Bitcoin forum to prevent tracking users' transactions.

CoinJoin. CoinJoin is a joint transaction involving multiple users intending to hide the relationships between 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 (cf. 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 sign the transaction separately, and one posts the transaction to the network..



Ch_A, Ch_B, Ch_C: Change addresses.

Fig. 1. CoinJoin

B. Wallet Selection and Basic Design

As aforementioned, we have selected the three leading wallets, which support CoinJoin transactions. we describe each design and summarize the main basic properties in the following.

JoinMarket wallet. JoinMarket [17] is a desktop wallet that 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 to sign. 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 significant number of makers (e.g., 50) [12]. 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 which input corresponds to the output (unless the transaction is created with one counterparty). Thus, in JoinMarket, it is possible to send the mixed coins directly to the desired recipient address. In other wallets, 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 [37] is a desktop wallet that uses a coordinator to create CoinJoin transactions. By Chaumian CoinJoin [9] the outputs are blindly signed [4] 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, the change address to get the remainder, and their blinded output to the coordinator to prevent correlating inputs to outputs. Then, the latter 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 step ii), the senders unblind and send their outputs to the coordinator. Suppose the latter finds his signature on the output. In that case, he creates a CoinJoin transaction with all the registered UTXOs as inputs and all the registered outputs and change addresses as the transaction's outputs. In step iii), the coordinator sends the transaction for signing the inputs by

TABLE I. OUR RESEARCH WALLETS FEATURES.

Wallet	Platform Support	Network	Anonymity set **	CJ† creation time	CJ amount	CJ fee
JoinMarket [17]	Linux, MacOS, Windows, RaspiBlitz, RaspiBolt, Qubes+Whonix	testnet/ mainnet	Set by user (Current default: 9)	X±±	Set by user	Set by user (Random fees ∼0.001%)
Wasabi [37]	MacOS 10.13+, Windows 10, Debian / Ubuntu, and Other Linux systems	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 [30]	Android	testnet/ mainnet	5 peers	X±±	0.001 BTC 0.01 BTC 0.05 BTC 0.5 BTC	TX0 fee+Pool fee 0.00005BTC TX0 fee+Pool fee 0.0005BTC TX0 fee+Pool fee 0.0025BTC TX0 fee+Pool fee 0.025BTC

** Per CoinJoin transaction. † CoinJoin. ±± Depends on the liquidity. ⋆ Per anonymity set.

the corresponding users, collects all transactions, combines the signatures, and broadcasts the transaction to the network [9].

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 a certain number of 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 [30] 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, there exist four pools (0.001 BTC, 0.01 BTC, 0.05 BTC, and 0.5 BTC) to create CoinJoin transactions with a flat fee rate (cf. 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 UTXOs; 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 UTXOs appear in the post-mix wallet. The Samourai application includes different wallets: main, pre-mix, and post-mix wallets. The user can send the mixed coins to the desired address using the post-mix wallet.

C. Wallet Basic Properties

In the following, we describe the basic properties of each wallet, such as platform support, CoinJoin transaction fees, and anonymity set. Mainly, Table I summarizes the wallet properties on both mainnet and testnet.

Platform support. Wasabi and JoinMarket as desktop wallets support most of the operating systems, while Samourai as a mobile wallet only supports Android.

Anonymity set per CoinJoin transaction defines 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 five 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 describes 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 requires that the number of registered peers reach 100 or the waiting time is achieved (i.e., the CoinJoin is created in one hour at the latest on the Bitcoin mainnet).

CoinJoin amount defines the amount of coins 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. JoinMarket uses random fees to the makers, which amounts to 0.001% of the transaction amount on the testnet. 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 be able to join the pool.

III. METHODOLOGY & EVALUATION CRITERIA

Our study uses a cognitive walkthrough following the methods in [34], [7]. A cognitive walkthrough is a technique, that mainly examines the expressiveness of application tasks and features, e.g., by answering how well first-time users do the tasks without formal training. In the context of this work, we followed the process described in Figure 2. After stating the problem and identifying the research questions (e.g., how practical, intuitive, and easy-to-use is a privacy-aware wallet?), we selected the wallets candidates for the evaluation, which support CoinJoin technique. Then, we defined the tasks to be performed by experts or users. The experts will evaluate each wallet's learnability by measuring how novice users may pass or fail the tasks and identifying possible errors or issues they may face [34]. While the walkthrough was conducted by two authors, the user study was carried out with a sample size n=2 of two computer science experts in Information technology security and privacy, who are very familiar with cryptocurrency wallets. Finally, we captured and analyzed the results, and proposed improvements.

Tasks Definition. Below is a description of the tasks that need to be performed by the participants:

- T.1 Installing the application.
- T.2 Generating a wallet.
- T.3 Funding the wallet.
- T.4 Performing a CoinJoin transaction.
- T.5 Transferring CoinJoin coins to the destination address.

Evaluation Criteria. The tasks are evaluated based on usability and design criteria adopted from [13], [20].

1) Usability criteria. Adopted from [13]:

- *Learnability*: The ease of using the system to do a task in the first attempt.

Fig. 2. Methodology Process

- *Errors*: The errors the user makes during doing a task and the ease of recovery from those errors.
- *Efficiency*: The time the user spends to perform a task (evaluated only in the user study).

2) Fundamental design criteria. Adopted from [20]:

- *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. An explicit notification should be provide in case of a problem.
- *Constraints*: The interaction possibilities are limited to clearly showing the user what can be done and preventing user confusion.
- *Mapping*: The user can clearly understand the relationship between functions (e.g., buttons) and associated actions. The interface terminology 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 was conducted using the Bitcoin testnet.

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., Python 3, Bitcoin Core). Selecting the appropriate assets based on the OS to download may confuse 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 to use this wallet. Configuration for Bitcoin core is provided in the documentation. There is also the "configuring JoinMarket" part on the installation page. We suggest integrating all Bitcoin core configuration guides in one part and referring to that whenever 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 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, similar to the Linux configuration.
Errors. There is no categorization on 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 indicating 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 confuse 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 to the transactions belonging to one entity. The only mitigation of address reuse in the 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.3), 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. Once the CoinJoin is created and broadcasted, the details can be found in the "TX History" tab.[1]

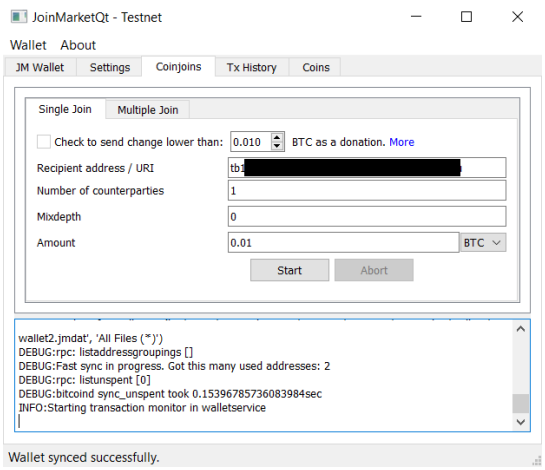


Fig. 3. JoinMarket CoinJoin

Errors. If the user chooses to spend the coins whit 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 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). By 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. It only lists the CoinJoin transactions 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

[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.

(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 ensure that she has a correct backup of recovery words. We also suggest informing 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 without paying attention to the numbering, 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 address 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 "Copied" message (achieves feedback). The QR code, public key, and key path appear, by clicking the small triangle on the left. But 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 transaction time, 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.

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). 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. We suggest automatically directing the user to one of the explorer. 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 double-clicks (fails consistency). We suggest copying the ID by double-clicking on that.
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 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.4), 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 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 on 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). A clear message can help. 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", and "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 been 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 for more to create CoinJoin transactions by the wallet repeatedly. 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. 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 [36], 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 clarify it to the users.

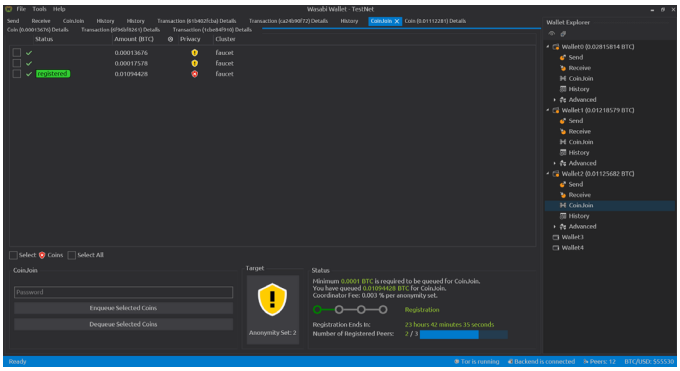


Fig. 4. Wasabi CoinJoin

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 she wants to spend, enter the destination address, amount, label, and wallet password, and hit the "Send Transaction" button, which is 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 beneficial, 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 related to the destination address. An informative message is suggested when the cursor is placed on 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 the "common input ownership" heuristic. We suggest a warning and a confirmation by the user 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 the Android .apk package, Google Play, and F-Droid. All the installation packages are accessible in the downloads tab of the 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. Changing the network to testnet can be offered via advanced options in the menu. The wallet should also warn the user when she is using the testnet.

T.2 Generating a wallet.
Learnability. To generate a wallet, the user should hit the create wallet button 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 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 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 essential. The current version may lead to critical problems for novice or careless users who may lose their funds forever since they cannot recover their wallets.

T.3 Funding the wallet.
Learnability. To fund the wallet, the user should hit the plus button at the bottom right to see the wallet functions, including "Receive", 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 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 transaction is shown on the wallet main page (achieves feedback). The incoming transactions are indicated in green while the outgoing transactions are shown in white. Clicking on the amount shows the transaction details, including date, time, status (the number of confirmations), miner fee rate, miner fee paid, and transaction ID. Clicking on the amount shows the transaction details, including date, time, status (the number of confirmations), miner fee rate, miner fee paid, and transaction ID. Clicking on the icon on the top right directs the user to the Blockstream.com website where the user can check the transaction in 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 an explicit tag such as "Checking transaction status".

The wallet main page is refreshed by pulling down the page to check the latest status of the transaction. However, the user does not get informed about this feature (fails visibility), a 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 called "CoinJoin". Therefore, it is unclear to the user if this item is used to create CoinJoin transactions (fails mapping). A new page is opened, by selecting Whirlpool; 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). We suggest offering these options similar to the functions in the wallet main page with appropriate icons (e.g., a plus button that includes these items). The term "UTXO" is also technical for novice users and 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. We highly suggest following the terminology that the community has adopted 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.5) where she can choose the coins that she prefers 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. 5). The pools are enabled according to the user's previously selected amount (achieves constraints); thus, entering larger pools is impossible. 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 list. However, it does not give any information where this unspent list is located (fails feedback), which is currently in the top-right menu on the wallet main page. By selecting yes and refreshing the Whirlpool page, all the UTXOs are listed as "Unmixed". The amount and "Mix 1/5-Queued" tag are indicated 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 on 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 the problem (fails feedback). We tried to create other wallets to join the same pool on different devices. However, in all the wallets the status remained "Mix 1/5-Joined a mix".This bug on testnet was also reported in [32]. Once the coins are entered into 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 the Whirlpool balance to the post-mix wallet balance). Checking different balances

in different wallets may confuse the user, as she can not see her total balance (fails visibility). We suggest clearly showing all the other 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 straightforward way to access these wallets is suggested.

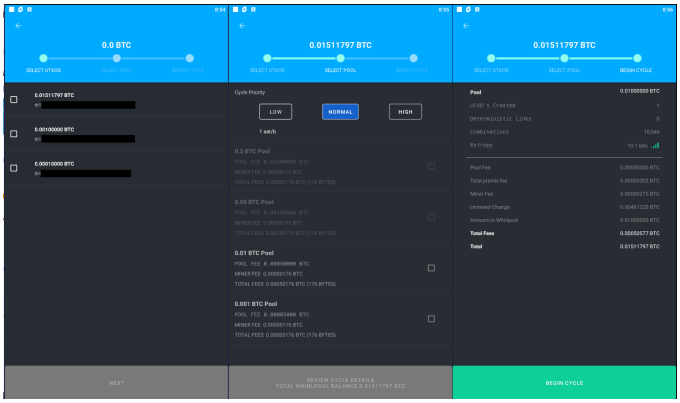


Fig. 5. Samourai CoinJoin

Errors. In our walkthrough on the testnet, selecting different cycle priorities did not change the amount 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 could not 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 and 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 the 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 actual 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.

[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.

V. Small-size user study

To evaluate the success and time on tasks, we conducted a small-size user study n=2 with two users from Information technology security and privacy who are familiar with cryptocurrency wallets. Table II illustrates the results. Both users successfully performed the tasks with the Wasabi wallet after the second attempt in T.4. In contrast, only one user (U1) completed all tasks with JoinMarket after the second attempt in T.4. None of the users could complete the tasks with the Samourai wallet after being stuck in CoinJoin in the testnet.

In the Wasabi wallet, the main reason that both users failed to complete the Coinjoin in T.4 in their first attempt was the ambiguous interface and information about the requirement for joining participants to create a CoinJoin. U1 stated, "*I find the CoinJoin interface a bit confusing.*" U2 pointed out "*I was too impatient for the CoinJoin to have enough peers*", the user continued "*the waiting for confirmation status in combination with the information that the registration ends is confusing (status at the bottom of page).*" In JoinMarket, U1, who succeeded in the second attempt for completing CoinJoin in T.4, specified different errors (e.g., error pushing (trx fees) and reducing the number of counterparties) as the main issues for failing the task in the first attempt. U2 did not complete the tasks in JoinMarket as Windows 10 detected the .exe file as a malicious one. In the Samourai testnet, both users reported being stuck in "Mix 1/5-Joined a mix" after several days and failing to complete T.4. As a result, they could not complete the task and conduct the T.5.

VI. Discussion

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 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. Therefore, Wasabi can be identified as the most user-friendly wallet. 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. Suppose the user wants to send large amounts to a destination address. In this case, 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

TABLE II. USERS' TASK SUCCESS AND TIME ON TASK IN MINUTES (M)

	Wasabi		**JoinMarket**		**Samourai**	
	U1	U2	U1	U2	U1	U2
OS	Ubuntu 18.04.5 LTS	Windows 10	Ubuntu 18.04.5 LTS	Windows 10	Android 10	Android 8
T.1 Installing the application.	✓ 4m	✓ 6m	✓ 13m	X	✓ 2m	✓ 5m
T.2 Generating a Wallet.	✓ 1m	✓ 2m	✓ 3m	X	✓ 2m	✓ 5m
T.3 Funding the wallet.	✓ 8m	✓ 4m	✓ 3m	X	✓ 2m	✓ 3m
T.4 Performing a CoinJoin transaction.	✓✓ several hours	✓✓ 90m	✓✓ 30m	X	X	X
T.5 Transferring CoinJoin coins to destination.	✓ 1m<	✓ 5m	*	*	X	X

✓: Success in the first attempt ✓✓: Success in the second attempt X: No success *T.5 in JoinMarket can be conducted in T.4.

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. Some errors do not clearly indicate what should be done to be handled. However, it has some features that can not be found in Wasabi and Samourai. It lets users modify the setting for the fees and the number of counterparties. Moreover, there are two essential 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 to the pool amount. Note that performing a CoinJoin for a large amount in JoinMarket is possible, which represents an advantage over the other wallets. However, 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 them to the user's address and then creating another transaction to send the CoinJoin coins to the destination. Thus, creating CoinJoin with Joinmarket requires one transaction less than the other two wallets and, consequently one transaction fee less. JoinMarket can be identified as the most expert-accommodating wallet.

Samourai provides a simple installation, and using it as a regular wallet is satisfying. However, the wallet is only released for Android. 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 is 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 use their mixing services correctly. 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. This misconception can also be found in one of the wallets' chat support [3].

[3]https://t.me/WasabiWallet/65300

VII. CONCLUSION

In Bitcoin, transactions are publicly available making it possible for malicious actors to use heuristics to deanonymize users. Several crypto wallets emerged, which adopt and integrate privacy-preserving techniques such as mixing protocols to enable users to mitigate such privacy issues. While in theory, such privacy features are of utmost importance in ensuring user privacy, using them in practice by novice users requires a good understanding of the techniques' basic concepts and an intuitive and user-friendly design.
This paper provided a cognitive walkthrough to evaluate the usability of three leading wallets that support CoinJoin transactions. Our results show that further improvements are required to make these wallets usable by novice users. Despite being small-size, the user study supports the walkthrough results and outlines the difficulties in performing CoinJoin transactions by users. In particular, users who do not understand the CoinJoin technique basics may find mixing cryptocurrencies through the wallet and dealing with error messages difficult. Additionally, merging previously mixed coins with other UTXOs belonging to the same user hampers the mixing benefits by exploiting the "common input ownership" heuristic for correlating different addresses and revealing user identities. Furthermore, it is essential to design an intuitive, and informative interface that is understandable by novice users; this will helps them perform CoinJoin transactions and be aware of the risks associated with specific actions. For the sake of this study, we used the Bitcoin testnet, mainly because of the high fees associated with using the mainnet. This might have impacted on the time on tasks. Besides, it was not possible to perform CoinJoin transactions on the Samourai testnet. Future work extends the user study to include more 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 European Union's Horizon 2020 research and innovation programme under grant agreement No 826078 (FeatureCloud) (4) the FFG Basisprogramm Kleinprojekt 39019756. (5) OEAD (Austria's agency for education and internationalization) Special Grant. We would also like to thank our anonymous reviewers for their valuable feedback and suggestions.

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] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten, "Mixcoin: Anonymity for bitcoin with accountable mixes," in *International Conference on Financial Cryptography and Data Security*. Springer, 2014, pp. 486–504.

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

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

[6] 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.

[7] 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.

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

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

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

[11] ——, "User-perceived privacy in blockchain," *Cryptology ePrint Archive*, 2022.

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

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

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

[15] 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.

[16] E. Heilman, L. Alshenibr, F. Baldimtsi, A. Scafuro, and S. Goldberg, "Tumblebit: An untrusted bitcoin-compatible anonymous payment hub," in *Network and Distributed System Security Symposium*, 2017.

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

[18] 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.

[19] 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.

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

[21] A. Mai, K. Pfeffer, M. Gusenbauer, E. Weippl, and K. Krombholz, "User mental models of cryptocurrency systems-a grounded theory approach," in *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*, 2020, pp. 341–358.

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

[23] ——, "Coinswap: transaction graph disjoint trustless trading (2013)," *URL: https:// bitcointalk. org/ index. php*, 2013.

[24] 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.

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

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

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

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

[29] 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.

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

[31] 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.

[32] Timmy2905, "Samourai wallet] whirlpool stuck on "joined a mix"," *URL: https://www.reddit.com/r/Bitcoin/ comments/in2kzt/samourai_wallet_whirlpool _stuck_on_joined_a_mix/*, 2020.

[33] 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.

[34] 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.

[35] J. H. Ziegeldorf, F. Grossmann, M. Henze, N. Inden, and K. Wehrle, "Coinparty: Secure multi-party mixing of bitcoins," in *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, 2015, pp. 75–86.

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

[37] ——, "Wasabi wallet," *URL: https://wasabiwallet.io/*, 2018.

APPENDIX

RELATED WORK

Blockchain privacy from the user perspective has been studied in [19], [8], [21], [11]. The studies indicate the lack of users' knowledge of privacy issues in blockchain and consequently, the users are not well informed on why and how they should use privacy techniques to mitigate the risk of de-anonymization in the blockchain. Krombholz et al. [19] 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. [8] 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 the CoinJoin technique. Apart from the need to improve users' knowledge, the usability of implemented privacy techniques has a significant role in their adoption in practice.

The usability research in key management [34], [7] performed a clear methodology for 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. [7] 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 [20] defined criteria for evaluating the top five Ethereum mobile wallets which are inspired by Norman [28] 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 [14]. 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 [33], 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.