Key Management in Bitcoin: Meet the new paradigm, same as the old paradigm

Position Paper

Shayan Eskandari
Concordia University
s_eskand@encs.concordia.ca

Jeremy Clark
Carleton University
clark@scs.carleton.ca

David Barrera
Carleton University
dbarrera@ccsl.carleton.ca

ABSTRACT

We don't know if Bitcoin millionaires sleep at night, only that they shouldn't. The average user is not prepared to securely manage a large amount of Bitcoins. This is due, in large part, to Bitcoin's use of public key cryptography which forces every one of its users into the unfamiliar territory of key management. We expect this to prevent Bitcoin from gaining traction beyond its enthusiasts. In this paper, we review the state of the art in Bitcoin key management tools, with particular focus on attempts to emulate the functionality of a password-based system within Bitcoin. Each approach presents trade-offs in security, usability and availability that lead us to conclude none is a silver bullet and the area deserves further attention. We thus propose a research agenda of possible future directions in improving key management techniques for Bitcoin users in the short- and medium-term.

Categories and Subject Descriptors

K.4.4 [**Electronic Commerce**]: Cybercash, digital cash; K.6.5 [**Security and Protection**]: Unauthorized access

General Terms

Security, Human Factors

Keywords

Key Management, Bitcoin, Usability

1. INTRODUCTORY REMARKS

- Different paradigm: protecting a key vs protecting a password. Bitcoins are secured to their owners by the private key of the address that the bitcoins are stored in. This private keys are not a key or password that the user knows the value so the normal user might not have any ideas about how this works and might lose the private key by simple mistakes
- Thesis: users aren't ready to manage keys.
- State of the art in terms of usability is to somehow convert the key into a password
- security and deployability (availability) problems
- keys=something you have.

Due to the meteoritic rise in Bitcoin's exchange rate, users are coming to the realization that it is no longer a crypto plaything; it is actual money.

We are not arguing that it will hinder adoption, as users will not realize what they are getting into. Rather it will harm traction, since users won't stick around to get fooled twice.

"Fool me once, shame on... shame on you. Fool me... you can't get fooled again." –W

Scope: bitcoin can be split into 3 parts: buying bitcoins (involving exchanges, OTC, etc), holding bitcoins (involving wallets, keys, etc), and spending bitcoins (creating transactions, waiting for verifications, etc). this paper focuses on the middle point.

2. PRELIMINARIES

2.1 Bitcoin Background

2.2 Other PKI-based Systems

- Same paradigm
- SSH - nerds
- PGP - turbo nerds
- SSL - client certs (no one, or spooks)
- SSL - server

3. BENEFITS

In this paper, we evaluate different approaches to secure and use bitcoins (Key management techniques). Our approach is by defining the benefits that the user would get by using each application in the manner of usability and security. Some benefits might not inclusively be in usability category or security, thus our categorization is not completely error prone, however it is, by the time of the writing, the most comprehensive study in this subject. The result of this evaluation is in Table 11. There are three different scores for each technique:

- Full score
- Half Score - Not the full score but has some features related to the evaluated benefit

  No circle - no score at all, either not applicable or does not have any feature for the evaluated benefit

3.1 Usability Benfits

3.1.1 Resilient to Equipment Failure

Keys are stored in `wallet.dat` or other wallet file formats. With hard disk failure or any relevant equipment failure that prevents the user to access this file, the keys and thus the bitcoins stored in it would be unusable.

3.1.2 Compatible with Change Keys

User can send the bitcoins from one address to the other, in this way the key that stores the bitcoins would be changed. In some approaches this might be a hard task to do, but in default client it would be as simple as a transaction.

3.1.3 Immediate Access

With the increasing size of the blockchain having access to the bitcoins and the ability to do transactions gets more important everyday. The user should get access to the up-to-date synced blockchain to be able to see his full amount of bitcoins to the date.

3.1.4 No New Software

Some approaches would need a new software to be installed on the system for the user to be able to access his funds or do transactions.

3.1.5 Portable

Portability in this case means the access to the funds from different resources or places, either it's a new computer or different computer in a different location.

3.2 Security Benfits

3.2.1 Malware Resistant

The value of bitcoins has increased in the past months and there has been malwares that focus on stealing keys[1]. The ability to resist these kind of malwares and attacks is a viral feature of the key management techniques.

3.2.2 Key Kept Offline

One way to secure the keys is to keep them offline, whether in a usb drive disconnected from the internet or cold storages. There are also methods to keep the keys in two parts, that both factors should be online for the user to be able to do a transaction.

3.2.3 No Trusted Third Party

By trusting a third party, there would be another place that the keys are stored and this would be a security risk if the party is compromised.

[1] `http://www.zdnet.com/blog/security/new-bitcoin-malware-steals-bitcoin-wallets-infostealer-coinbit/8804`

3.2.4 Resistant to Physical Theft

On the event that the hardware containing the keys is stolen, the thief can access the keys if they are not securely stored, such as strong encryption.

3.2.5 Resistant to Physical Observation

Evesdropping is not applicable on the keys stored in the file but with the new approaches different ways of physical observation could be used to get the keys.

3.2.6 Resilient to Password Loss

Password loss usually is handled by the service provider and either there would be a password reset option or not. On cases that there is no user provider or third party to do so, it is only the key management techniques toward this issue

4. BITCOIN PRIVATE KEY MANAGEMENT

At the core of Bitcoin's functionality are keypairs. The public key allows users to receive coins and check their wallet balance, and the private key allows users to send coins to other addresses. In this Section, we review the main mechanisms used in the current Bitcoin ecosystem to manage these keys.

4.1 Default Client

On first launch, the official `bitcoin-qt` client creates a `wallet.dat` file in the Bitcoin data directory (usually a hidden folder inside the user's application folder). The `wallet.dat` file contains the set of all private keys belonging to the user, allowing the user to sign transactions (*i.e.,* send coins). Anyone with access to the private keys inside `wallet.dat` can spend the coins associated with those keys. Thus, access control on the `wallet.dat` file is extremely important.

The `wallet.dat` file can be read by any application with access to the user's application folder. Malware is a particularly noteworthy example here, since theft of the `wallet.dat` file by a malicious developer results in immediate access to the victim's funds. In 2011, Symantec discovered the *Infostealer.Coinbit*[2] malware, which targeted Windows systems in an attempt to find `wallet.dat` files and sent them via email to the attacker.

Unintentional sharing of the `wallet.dat` file is also a concern in the default client. Users must be cautious to not inadvertently share their bitcoin application folder on the Internet or to a location outside of the user's control. Possible sharing includes peer to peer (P2P) file-sharing networks, off-site backups, or shared network drive. Physical theft of the system hosting the `wallet.dat` file is also a concern, especially in the case of portable computers. Although it is possible to encrypt `wallet.dat` with a custom password.

By keeping the `wallet.dat` file locally, users must also be wary of *threats to digital preservation* [**?**] such as general equipment failure due to natural disasters and electrical failures; acts of war; mistaken erasure (*e.g.,* formatting the wrong drive or deleting the wrong folder); bit rot (*i.e.,* undetected storage failure); and possibly others.

[2] `http://www.symantec.com/security_response/writeup.jsp?docid=2011-061615-3651-99`

	Malware Resistant (3.2.1)	Key Kept Offline (3.2.2)	No Trusted Third Party (3.2.3)	Resistant to Physical Theft (3.2.4)	Resistant to Physical Observation (3.2.5)	Resilient to Equipment Failure (3.1.1)	Resilient to Password Loss (3.2.6)	Compatible with Change Keys (3.1.2)	Immediate Access (3.1.3)	No New Software (3.1.4)	Portable (3.1.5)	Blank	Blank	Blank
Traditional cash	●	●	●		●	●	●	●	●	●	●			
Key in file (default)			●		●		●	●	○	●				
Password-protected		○	●		●			●	●	●				
Air gap	○	●	●		●		●	●						
Offline storage		●	●			●	●				●			
Password-derived key		●	●	●		●			●		●			
Hosted (hot)						●	○	●	●	●	●			
Hosted (cold)		●				●	○	●		●	●			

Table 1: A Comparison of Key Management Techniques for Bitcoin.

Advantages of using the `bitcoin-qt` client in it's default configuration no trust needed in a third party, and no need to recall yet another password. Additionally, the user can spend coins and receive change without the need to perform extra steps (see Section 4.4).

One of the disadvantages of using `bitcoin-qt` is with the increase size of blockchain *(Should explain blockchain in Bitcoin background section)* it takes up storage space on the user's computer. On the time of writing this paper it uses around 13 gigabytes [3] for storing blockchain . Also for new `bitcoin-qt`'s users it might take days to synchronize their local copy of blockchain.

To score the `bitcoin-qt` it does not have malware resistant nor keep keys offline, there is no need for third party trust ad nothing to be physically observed. It is not resistant to physical theft nor equipment failure cause the keys would be lost in either way. On password loss, there are no ways to reset the password so resilient to password loss, however password is not set by default and should be enabled by the user. Due to huge size of blockchain it has somehow the immediate access score and none in portability, also there is no need to for any new software and it is compatible with change keys as it would be just a transaction to a new address.

4.2 Simplified Payment Verification (SPV)

It is possible not store all the block-chain and verify everything but to connect to an arbitrary full node and download only the block headers. This would resolve the storage and synching problem `bitcoin-qt`'s are facing now.

bitcoinj[4] is an implementation of SPV. `MultiBit`[5] is a bitcoin wallet that uses this feature. `MultiBit` is an open-source wallet with the intention to be fast and easy to use, it also keep its wallet file encrypted. SPV is more a technique to sync blockchain with the client, however because there are some clients different than the `bitcoin-qt` that uses SPV and have some different features with include this in our evaluation. These clients are a new software to be installed, however they could be installed on a usb key to be used on any system (same operating systems) so they are partly portable and keys could be kept offline. Even though they are stand alone applications they trust their provider's nodes for the blockchain headers, albeit the attack or fake transactions on this level due to specific features of SPV protocol is really unlikely it gets the half score. Other benefits are as the same as the default client.

4.3 Password-protected Wallet

Some Bitcoin clients allow wallets (specifically the private keys in the wallet) to be encrypted with a user-chosen password. This would be good for a stolen wallet file but still malwares could use keyloggers or other method to get the password. Also users might think that the password is applicable for their bitcoins in every client, like a password for an online wallet, however it's more of a two factor key to get to the funds. By forgetting the password, there might not be any solutions to recover the lost bitcoins, although there are services available to bruteforce the password only feasible depending on the complexity of the password[6]. All the benefits of these technique are the same as the default method, however it would be somehow more resistant to malwares.

[3] `https://blockchain.info/charts/blocks-size`

[4] `https://code.google.com/p/bitcoinj/`

[5] `http://multibit.org`

[6] `http://www.walletrecoveryservices.com`

4.4 Offline Storage

The most secure way to save the private keys from being stolen is to have them disconnected from the internet, although it has drawbacks of not available to send the funds immediately. Offline storage don't have the Immediate access benefit.

Paper Wallets are just a print out of the private keys, there are designs to keep it more secure but in the end whoever has the paper in hand can spend the bitcoins.[7]

, thus no resistant to physical observation but no need to trust any third parties. It is also possible to put the wallet file into a usb and keep it in a safe offline place that also needs to be physically secure and also a longer procedure to import the keys into a wallet and spend the bitcoins, also vulnerable to malwares and need new softwares for the import process. There has not been a Hardware wallet in production yet, but TREZOR[8] has promised a full functional and secure hardware wallet for bitcoin, it is a plug and play USB key that offers transaction signing for the common wallets on the computer without revealing the private keys. If the paper wallets tear off or the hardware wallets fail or even if the password for the encryption is lost, the keys would could not be retrieved thus resilient to equipment failure and password loss applies. although these methods are pretty portable, Changing keys due to the hard-coded-like keys needs to generate new keys and redo all the necessary work to get the new keys in proper formats.

4.5 Air Gap

Air gap falls into offline storage methods but because some features differs we have it as a separate technique. The difference is that the device that holds the key would not be connected to internet in any time given. One implementation of these method would be discussed on section **??**.

DISCUSSION ABOUT THE BENEFITS IS NEEDED! I THINK THE TABLE IS WRONG IN SOME FIELDS!

4.6 Mobile Wallets

Bitcoin-wallet[9] is a Bitcoin wallet for Android and BlackBerry OS. It uses SPV to take less storage for the blockchain and it's completely peer-to-peer and does not use It uses a custom format for wallets which should be compatible between clients using bitcoinj and also possible to backup the wallet.

Mycelium Bitcoin Wallet for Android[10] is another bitcoin wallet for mobile devices. It uses SPV for block-chain synchronization, has the ability to import private keys for secure cold-storage integration and it's possible to export the keys to external storage of the device.

However having the wallet file in a mobile device always has the cut back of losing your wallet when the device is stolen. Mobile wallets in the matter of evaluation do not differ from wallets, but may have more vulnerability to physical theft.

5. MANAGING WITH A PASSWORD

cf. online banking (cormac herley).

5.1 Password-Derived Key

Also known as deterministic wallet, is a system that derives the keys from a starting point known as seed that could be random or user chosen string. This is good in the security manner that there is no need to save the private keys in a computer and just save the string or write it down on a paper. There are different methods to secure this technique such as high iteration counts to slow down the attacker on simple user-chosen strings. It is important to use a strong string as it would be the key to access the funds, services such as brainwallet [11] make this technique available to users. Even though the keys are not stored in the computer and are resistant to physical theft, this technique is not malware resistant as a simple key logger could make the funds available to the attacker so does the physical observation. Depending on the key generation method, the most common methods are open source and also could be done in offline systems so no trust in third party is needed. However if the user forgets his keys the funds are lost forever as there is no way to recover the private keys without the initial seed, also for a new key the whole process should be repeated. One problem with this technique is that it is highly dependent on the software it use to generate the keys from the initial string and if the software is not available it would be hard to reproduce the keys from the string. That being said, as long as the software is available the funds are accessible from any computer that has access to the software.

- entropy reduction
- salt, iteration count
- parsing (spec that was followed) requires a tool. if you lose the tool, you may not be able to recover your keys. standard may not be detailed enough, or not properly implemented.
- change accounts gets messy

5.2 Hosted (Hot Storage)

One easier way to manage your bitcoin wallet is by using online wallets. It has it's own advantages and cutbacks. Trusting third party is one issue and the party getting hacked is another. There has been known trustable third parties that just went out of business on their first big hacks, such as instawallet[12], Bitcoinica[13], etc. Bitcoinica in 2012 lost more than 60,000 bitcoins due to two successful attacks.

In the time of writing this paper the most popular online wallet is blockchain.info[14] that offers two-factor authentication via email verification, but resistant to forgetting the

[7] https://bitcoinpaperwallet.com

[8] http://www.bitcointrezor.com

[9] https://play.google.com/store/apps/details?id=de.schildbach.wallet

[10] https://play.google.com/store/apps/details?id=com.mycelium.wallet

[11] http://brainwallet.org/

[12] http://www.theverge.com/2013/4/3/4180020/bitcoin-service-instawallet-suspended-indefinitely-after-hack

[13] http://www.theverge.com/2012/8/10/3233711/second-bitcoin-lawsuit-is-filed-in-california

[14] https://blockchain.info/wallet

password.
These online services might shutdown due to legal implications or attacks, Also the user might be tricked to reveal his password on a phishing attack[15], thus these services are not resistant to physical observation. In short, Hosted (Hot) wallets are not malware resistant nor keep the keys offline, also there is the need of trust and also it is not resistant to physical theft, however because of the usual backup of such services it is resilient to equipment failure. They might offer password reset options to recover the lost password. Compatibility with change keys is as easy as doing a transaction to the new account and they offer immediate access to the funds as they are online services and synced with block chain and usually accessible through a browser thus portable and no need for a new software.

5.3 Hosted (Cold Storage)

It is possible to apply the offline storage method to hosted wallets but there are more drawbacks than benefits, not assuming security as the only measure. coinbase announced that more than 87% of it's users bitcoins are stored in cold storage[16]. It would be secure for most uses but if the total withdrawal in a period of time becomes greater than the funds available in online storage there would be a 48 hours delay on the payments, for scoring purpose we assume this service is fully implemented in cold storage and there is no bitcoins in hot storage of the service. With keys being offline, it would have resistance toward malwares but no immediate access to the funds, however the user should trust the third party service. The rest of the benefits are the same as hosted on online storage.

6. EVALUATION METHODOLOGY

Another method to evaluate the usability of these techniques is heuristic evaluation (citation). In this paper, we employ cognitive walkthrough as our methodology for usability evaluation. ...

6.1 The Core Tasks

The core tasks that we are going to perform for each key management technique are as follows.

CT-1 Finalize a receiving address from the primary device[17]

CT-2 Authorize the transaction from the primary device

CT-3 Authorize the transaction from the secondary device[18]

CT-4 Losing the main credential (recovery options)

The core tasks will be performed with the following clients with default configuration on each category:

1. Key in file (default): `bitcoin-qt`
2. Password Protected: `MultiBit`
3. Air gap: Bitcoin Armory [19]
4. Offline Storage: Paper Wallet
5. Password-drived key: Brain Wallet
6. Hosted: Blockchain.info wallet

6.2 Usability Guidelines

The set of guidelines that we use to evaluate each of the core tasks are as follows.

G1 Users should be aware of the steps they have to perform to complete a core task.

G2 Users should be able to determine how to perform these steps.

G3 Users should know when they have successfully completed a core task.

G4 Users should be able to recognize, diagnose, and recover from non-critical errors.

G5 Users should not make dangerous errors from which they cannot recover.

G6 Users should be comfortable with the terminology used in any interface dialogues or documentation.

G7 Users should be sufficiently comfortable with the interface to continue using it.

G8 Users should be aware of the application's status at all times.

These guidelines are drawn from a variety of sources [**?**, **?**, **?**, **?**, **?**, **?**] and are intended for evaluating Bitcon Key management specifically. However they are suitably broad and may find application in other usable privacy walkthroughs. We now individually justify the inclusion of each.

G1: Users should be aware of the steps they have to perform to complete a core task.

This is a restatement of the first guideline of Whitten and Tygar [**?**]. Every user of a new application knows certain things before using the system and learns certain things during the use of the system. In the cognitive walkthroughs we carry out here, the presupposition is that the user knows enough to start the process for each core task—in the case of installation, the user can download the installation file and open it; in the case of configuration, the user can explore the user interface or follow cues. We are evaluating how the application cues the user to perform the intermediary steps between these broadly defined tasks.

[15] `http://www.theverge.com/2013/4/5/4186808/bitcoin-banker-coinbase-phishing-attacks-user-information-leaked`
[16] `http://blog.coinbase.com/post/33197656699/coinbase-now-storing-87-of-customer-funds-offline`
[17] primary device is the initial device that the key is generated on
[18] any device other than the primary device
[19] `https://bitcoinarmory.com/`

G2: Users should be able to determine how to perform these steps.

Once the user is aware of what intermediary steps are necessary, she must be able to figure out how to perform these steps. This is the second guideline in [**?**]. It is assumed the user has a mental model of how the system works. It is thus important that the system model be harmonized with the user's mental model if the user is to be successful in performing the necessary steps required to complete each core task [**?**]. What is less obvious is why we cannot fully rely on the user to simply modify her mental model when given conflicting information.

A predominate reason is that humans have a stronger preference for confirming evidence than disconfirming evidence when evaluating their own hypotheses. This cognitive bias is well illustrated by Wason [**?**], who conducted a study where a set of subjects were given the following sequence of numbers: 2,4,6. The subjects were told that the numbers followed a rule, and their task was to determine the rule by proposing their own sequence of numbers, which would be declared as matching the rule or not. The rule was any ascending sequence of numbers. However most subjects derived a more complicated rule, such as every ascending sequence of numbers differing by two. The point of this test was that the subjects, on average, had a preconceived idea of what the rule was and only proposed sequences to confirm that rule, instead of also proposing sequences that would falsify their perceived rule.

Confirmation bias is important in usability because it proposes that users are biased toward only seeking and accepting information that confirms their mental model, and thus may avoid or even ignore information that contradicts it. It cannot reasonably be expected that users will easily and quickly adapt their mental model to new information.

A second concern with how users perform these steps is that security is a secondary goal [**?**, **?**]. If the user is given two paths to completing a core task—one that is fast but not secure, and one that is slower but more secure–it cannot be assumed that the user will take the latter approach. In fact, studies in behavioural economics demonstrate that humans often prefer smaller immediate payoffs to larger future payoffs, such as $50 today instead of $100 a year from today [**?**]. Using software securely has a greater (usually non-monetary) payoff for the user, but this utility has to be substantially higher than the alternative to justify the delay in achieving it.

G3: Users should know when they have successfully completed a core task.

In other words, users should be provided with ample feedback during the task to ensure they are aware of its successful completion. This principle has been proposed in the context of heuristic evaluation [**?**] and for a cognitive walkthrough [**?**]. It was also mentioned by Cranor [**?**]. In Bitcoins, it is essential that the user is provided with confirmation of the task's finalization, such as successful back up of `wallet.dat`.

G4: Users should be able to recognize, diagnose, and recover from non-critical errors.

Users will likely make errors in performing the core tasks and it is important for them to be able to recover from these errors [**?**]. It is important for users to be given concise error messages.

G5: Users should not make dangerous errors from which they cannot recover.

This guideline is from Whitten and Tygar [**?**]. In Bitcoin subject, the most dangerous error is to reveal the private key which is associated with the address that holds the funds. Also in case of backups, the corrupted `wallet.dat` would be useless for recovery.

G6: Users should be comfortable with the language used in any interface dialogues or documentation.

Wharton *et al.* emphasize that applications should use simple, natural, and familiar language [**?**].

G7: Users should be comfortable with the interface.

This is the fourth principle of usable security of Whitten and Tygar [**?**], and is an essential part of the principal of psychological acceptability quoted by Bishop [**?**].

G8: Users should be aware of the system status at all times.

This principle was proposed in the context of heuristic evaluation [**?**] and cognitive walkthrough [**?**]. Cranor advocates the use of 'persistent indicators' that allow the user to see all the required information at a glance [**?**]. In terms of Bitcoin, we are looking for indicators that show the balance and the addresses that is included in the `wallet.dat` and also the transaction history.

6.2.1 Key in file

First step for user is to download the `bitcoin-qt` from bitcoin.org website[20]. For the first time user choosing between the wallets and figuring out which section of the site to start with might be confusing, However it is much more user-friendly now than when bitcoin was introduced. When clicking on `bitcoin-qt` download option, the page shows different types of operating systems compatible files to be downloaded, albeit, the default download button would automatically determine the system the user is using and downloads the relative setup file. It has a pretty straight forward wizard installation procedure. Now the user runs the application for the first time. On the first run user would see the main page of the application that might be confusing for him because it needs to sync with blockchain to show the information instead of just saying "Out of Sync", however the first task that is to have an bitcoin address is done (CT1), and he can find the address with clicking on the "Receive Coins" tab of the application but might be confused with the "Addresses" tab that is like a contact list for other addresses, this being said, he would not see the final balance until the sync is done (this might take days and about 10GB of storage space and also violates G8 as there is no indication of how long this would take to be synced), Although G1,G2 and G3 are

[20] `https://bitcoin.org/en/choose-your-wallet`

slightly violated, There is no indication or pop ups that the address has been made and is ready to use or the balance is not yet finalized. As for CT2, the minimum knowledge of bitcoin addresses is needed, to not to make any mistakes. for example if the receiving address is in the wrong format, there is no error to show that is so but just a red background appears on the address that is a slight violation of G4. For Backing up the `wallet.dat`, after clicking on the "Backup Wallet" in "File" and choosing the directory there is no sign that the backup has been successfully done and incase of any problems the keys are not recoverable (G5), however on encrypting the wallet the message box indicates that on losing the passphrase there is no way to recover the encrypted keys (CT4). CT3 is not easily possible with `bitcoin-qt` as the new installation is needed on the new device thus new wallet is generated, however user can replace his own wallet with the newly generated one to have access to the funds, but nothing has been mentioned about this in the documentation(G1).

6.2.2 Password Protected

Although it is possible to encrypt the `wallet.dat` with `bitcoin-qt`, there is no emphasize or alerts to do so, however in `MultiBit` client one of the first steps is to password protect the wallet file, this is one of the reasons that this client has been chosen to be analyses for this cognitive walkthrough, also because it uses SPV for faster blockchain synchronization this client is more popular among new users. The download[21] and installation process is really straightforward. On the first run the welcome page would pop up that has the explanation of the core tasks that could be done with `MultiBit` such as where the send, requenst and transaction tabs are and how to password protect your wallet and also help options for all the other functionalities (G1,G2). Unlike `bitcoin-qt`, `MultiBit` help option gives direct and non-technical guides on how to do the desired functions. The interface is pretty easy to understand and it shows the status of the program (Online, offline, out of sync) on the left down side, the balance of the user's wallet on the up left and the latest price of bitcoin on the up left of the window(G8). There are not too many jargons and technical vocabulary used (G6). Same as the "key in file" after the first run the receiving address in finalized and it is possible to receive bitcoins (CT1). For CT2, the user would know that he needs to send some funds to another address, it is by clicking on the "send" tab, `MultiBit` has a really simple and complete interface to do so(G7), where it is possible to import the sending address by QRcode[22] or from the clipboard or by typing it in the "Address" field. If the address in not correctly formatted or the amount of the transaction is more than the balance, a fully detailed error message would pop up to explain what went wrong (G4). The interface also shows the USD value of the transaction amount on the side of the bitcoin amount. With the main Functionalities, CT3 would not be possible, but it is possible to export the private keys and import them on the secondary device to do the transactions. To do so, user would look on the menus and finds the option "Export Private Keys". By default the password protected backup is enabled and needs to input the password and incase user disables it, there would be a red sign saying "Any one who reads your export file could spend your bitcoins", also on clicking the export button there would be the message confirming that the backup has been completed and the second check that is to match the backup with the current wallet info has been done (G5). Now with this exported file in hand the user can do an "Import Private Keys" on any given computer (secondary device) to authorize a transaction (CT3) and also do a recovery of the lost wallet file (CT4).

6.2.3 Air gap

6.2.4 Offline Storage

There are different options for Offline Storage of a bitcoin as described in Section **??**. For the cognitive walkthrough we choose the paper wallet to include a different approach of saving the keys as the other methods has similar parts to key in file or the Air gap technique.
For CT1 user could enter any of the bitcoin paper wallet sites (e.g www.bitaddress.org). By mouse movement or adding some text to the proposed fields, user can make a random seed to generate a random unpredictable address. The address would appear on the left of the screen with the private key on the right labeled as secret. As for CT2 and CT3 there is no difference in the primary device and the secondary as the key should be imported from the paper wallet into the wallet client. The best way to do so is to have an online wallet on blockchain.info or use the Bitcoin Armory application to import the private key or use the signing transaction option to authorize the transaction. On loosing the key (CT4) there is no way to recover the keys without the use of any other backups.

6.2.5 Password-drived key

To start using this technique user has to either run the offline brain wallet or use the online services such as brainwallet.org. On entering the site user is asked to enter a customized personal string to produce the random seed to generate the bitcoin address (CT1). For example with the string "brainwallet strong password" the address of "12HH5MmDhPsZWLb1QZKhptzGpPG1R73gwy" has been generated. note that the string should not be shared or the funds would be available to whoever that has the string. Same as the offline storage on any system that has access to the software it is possible to produce the keys from the users string so CT2 and CT3 are the same and easy to do on any system. On forgetting the credentials, it would be impossible to recover the keys (CT4).

6.2.6 Hosted

Hosted services are the easiest ones to use for the new users. There would be a sign up for an online wallet that we chose blockchain.info [23] as it is the most famous online wallet on the time of the writing. The user clicks on "Start a new wallet" and provides an email address with a password for his login. On the first page it has a warning that on forgetting the password there is no way to recover the account (CT4), however it gives the user a "Wallet Recovery Mnemonic" that is a set of words that could be used to recover the account. On the first login the user would have a finalized address that he can use to receive and send bitcoins. The interface is pretty user-friendly and easy to use. As for CT2 and CT3

[21] `https://multibit.org/`
[22] `http://en.wikipedia.org/wiki/QR_code`
[23] `https://blockchain.info/wallet`

there is no difference between the devices as a simple web browser would give access to the account for the user.

7. A RESEARCH AGENDA

- better import/export of keys
- better language and guidance on key management
- how do you know the key is in wallet.dat ?
- How do you know if a file has been read? Specifically, how do you know if your private key file has been read by an app other than your wallet? We have unix `atime` (time of last access), but we can't log what process read the file.
- publishing public keys. people need to keep these around, and they can use to check if they have the right privkey.
- if you have your pubkey and you have the salt, you can avoid rainbow tables, but your pubkeys get longer. (add salt via commitcoin)
- boyen iterated hashing of passwords -> compute amount of iterations based on money, script
- digital death
- sinkhole accounts
- incentives are there because it's money. not like ssh or ssl.
- min password entropy as a function of how much money its protecting: examine in terms of falling cost to brute force (e.g. w/ EC2), vs rising price of bit coins. -> future proof against lowering EC2 costs and raising bitcoin costs
- bitbills, oblivious printing and other physically intuitive media that can behave (sort of) like cash
- theft/loss mitigation, making it easier to split wallets to mitigate loss
- account rollover vs. password changes
- graphical passwords
- security of RNGs. i read that vanitygen (for making vanity bitcoin addresses) uses OpenSSL's RNG and it's secure because it's "used on thousands of websites".
- air gap -> how good is it?

8. RELATED WORK

Blah blah blah.

9. CONCLUDING REMARKS

In this paper, we have ...

	CT-1	CT-2	CT-3	CT-4
default	very easy	easy	difficult	not possible
Password-Protected	easy	easy	difficult	not possible
Airgap				
Offline Storage	easy	difficult	difficult	not possible
Password-derived key	easy	easy	easy	not possible
Hosted	easy	easy	easy	difficult

Table 2: Cognitive Walkthrough summary result