Critiquing Bitcoin's rendition of usable key management

ABSTRACT

In all of the excitement surrounding Bitcoin, it is easy to forget that the decentralized currency assumes a solution to the longstanding problem of designing secure and usable public key cryptography for authenticating users. Despite decades of research on this issue in parallel technologies, such as digitally signed email, we show that Bitcoin technologies for managing user keys stall on many of the same basic issues. At the same time, the developer-heavy Bitcoin community has been very prolific in producing deployed technologies with a wide variety of approaches to solving these issues. We therefore argue that studying this suite of technology offers the clearest picture yet of the challenges in usable key management for digital signatures.

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

In all of the excitement surrounding Bitcoin, it is easy to forget that the decentralized currency assumes a solution to the longstanding problem of designing secure and usable public key cryptography for authenticating users. Despite decades of research on this issue in parallel technologies, such as digitally signed email, we show that Bitcoin technologies for managing user keys stall on many of the same basic issues.

At the same time, the developer-heavy Bitcoin community has been very prolific in producing deployed technologies with a wide variety of approaches to solving these issues. We therefore argue that studying this suite of technology offers the clearest picture yet of the challenges in usable key management for digital signatures.

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.

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.

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

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

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

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

	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.

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.

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

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

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

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

[7] `https://bitcoinpaperwallet.com`

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 is not connected to internet in any given time. One implementation of these method would be discussed on section 6.2.3.

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.

[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/

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 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.

[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

[15] http://www.theverge.com/2013/4/5/4186808/bitcoin-banker-coinbase-phishing-attacks-user-information-leaked

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 and balance 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.

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,

[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/`

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

We assume the user knows how to download the client from the main site. It has a straight forward wizard installation procedure, user runs the application for the first time.

CT1.

On the first run user would see the "Overview" page that might be confusing for the first time user because it is just showing "Out of Sync" in red. However the first task that is to have a bitcoin address is already done (CT1), and he can find the address with clicking on the "Receive Coins" tab of the application that might be confused with the "Addresses" tab (that is like a contact list for other addresses), this violates G2 and G3. User should be aware that the application must be connected to internet to get synced (G1), however except small status indicator on the bottomright side of the window that shows a small red cross in-between two black windows, there is no other alerts and that is a violation of G1, even with mouse over on the icon it would say "0 active connections to bitcoin network" that for the user not familiar with the terminology, does not reflect the meaning that the client should be online(G4 and G6). For the Balance he would not see the final balance until the sync is done (This is more than 10 GB of download and because of the peer to peer nature of the download it may take days to be synced [20]).

CT2.

This is the easiest task for this client as all the keys are already loaded in the application from `wallet.dat` and user can use the "Send" tab to authorize a transaction to a given bitcoin address.

CT3.

By installing `bitcoin-qt` on the new device, a new wallet file is generated. In order to complete CT3, user might try finding import options in the newly installed wallet client or drag and dropping the file into the client, but failed to do so because there is no such an option, even if the user tries to find the documentation he would fail as there is nothing in the help menu except debug window that is for advance user to tweak the application.Most of the users would stop here, However there is one way to do so and that is to replace his newly generated wallet file with his previously owned wallet file, the one that was setup in CT1, to have access to his funds, but nothing has been mentioned about this in the documentations (G1). To do so user have to back up the `wallet.dat` with the "backup wallet..." option in the "File" tab and chose a directory to save the `wallet.dat`.

Now he need to have a secure way to transfer this file to the secondary device, which could be a USB flash drive. Depending on the nature of primary and secondary devices and also the wallet clients the method of copying the wallet

[20] `https://en.bitcoin.it/wiki/Satoshi_Client_Block_Exchange#Performance`

file might differ [21]

One more obstacle is to find the path to copy `wallet.dat` on the new device since there is no import option on the `bitcoin-qt`, this would be possible by either searching the local file system for `wallet.dat`, which he might not succeed due to non-searchable system reserved folders or not knowing the exact file name, or searching on the internet for the answer. On Mac OS X, the path is /Users/User/Library/Application Support/Bitcoin/wallet.dat [22]. The next step is to replace the new `wallet.dat` with the one from the primary device. it should be noted that the name of the file should be exactly `wallet.dat` for the `bitcoin-qt` to be able to read the file. Some of errors that the user might encounter during this procedure are as follows:

- User might accidentally copy `wallet.dat` from the primary device wallet client path instead of the one he backed up, this would cause to have a corrupted `wallet.dat` and not readable by the secondary device's `bitcoin-qt`, `bitcoin-qt` has a procedure to lock `wallet.dat` while it is in use, this may cause to have a corrupted file if the file has been copied while it was locked

- There is the possibility to replace the backed up `wallet.dat` instead of the newly generated wallet file and lose the backup file on the secondary device (G4)

- On the secondary device the final balance might be wrong and there would be the need to resynchronize the blockchain to have the correct final balance (G3)

As for CT3, all guidelines (G1-G8) has been violated.

CT4.

For recovery options, the procedure is the same as CT3 but on the same device. It should be noted that the saved backup `wallet.dat` should be kept in a secure storage(*** NEED REFERENCE FOR SECURE STORAGE****) as it contains all the keys and so all the funds stored in the addresses. User might try to store the backed up file on the same device that could lead to losing all his funds due to hardware failure (G5).

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 recommended 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 amongst new users.

CT1.

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, request and transaction tabs are and how to password protect the wallet file and also help options for all the other functionalities (G1,G2).

`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 bottom left status bar, the balance of the user's wallet on the up left and the latest price of bitcoin on the up right of the window(G8). There are not too many jargons and technical vocabulary used (G6). After the first run the receiving address in finalized and it is possible to receive bitcoins (CT1). By clicking on "Request" tab the finalized address and it's relative QR Code[23] appears, User can now enter his desired amount to be received to generate the appropriate QR code with his input value, even though just having the address suffice to complete this task. On the section titled "Your receiving addresses" all the addresses stored in the wallet file is shown, By clicking on the "New" button he can generate a new bitcoin addresses. `MultiBit` also has the short come of not showing the user that he has to be connected to internet to get synced and it will stay on "connecting" mode,this slightly violates G1 and G2. As it is mentioned on the "Welcome page", every option in `MultiBit` would show help tips with hovering the mouse over the option.

CT2.

To authorize a transaction user has to click 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 QR code or from the clipboard or just 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). Also if the wallet client is not synced, the send button would be disabled. If it's synced and there is no error on the balance and the destination address, the transaction is complete after clicking on send button and approving the transaction.

CT3.

Same as the "Key in file" this task is not a main functionality of `MultiBit`. On the primary device user has to look in the options to find the backup options, here it is in Tools - "Export Private Keys" (violates G6). It will show details about current wallet file, the path for the export file to be saved and also password of export file that is enabled by default. In case user tries to save the exported file without password, there would be a warning saying "Anyone who can read your export file can spend your bitcoin." in red. By having a password protected export file, use can copy the file safer to the secondary device. By clicking on "Export Private Keys" button it will save the wallet file in the given path and also checks if the file is readable (no violation of G4 and G5). On the secondary device user installs a fresh copy of `MultiBit` and looks for Import options that is Tools - Import Private Keys. The window looks the same as the Export window but with import functionality. User has to browse for his exported file from the primary device and type in the password and then click on "Import Private Keys" button. It will confirm the completion of the import and changes the balance according to the balance of the new imported addresses. Now he's able to authorize a transaction to any given destination addresses.

[21] A Comparison of Secure File Transfer Mechanisms `http://www.process.com/tcpip/sft.pdf`

[22] `https://en.bitcoin.it/wiki/Data_directory`

[23] `http://en.wikipedia.org/wiki/QR_code`

CT4.

This task is the same as CT3 but on the same device. The password-protected back up file should be stored in a secure storage that would prevent it from being lost or stolen, however it would be hard to recover the wallet file without knowing its password.

6.2.3 Air gap

This technique has not yet been documented well and it is suitable for advanced users only. For the cognitive walkthrough, we assume user has good knowledge of bitcoin terms (G6) and how the air gap method works (G1). Bitcoin Armory is one of the advanced bitcoin wallets that is going to be used for this walkthrough, although for the online system it should be executed while `bitcoin-qt` is already running, to be able to load the blockchain. It is possible to use some other applications to do so, however we are going to walkthrough the most secure way to implement air gap for bitcoin wallet.

CT1.

First, User should install Bitcoin Armory on the offline computer. On the start, the welcome page offers the option to "Import Existing Wallet" and "Create Your First Wallet!". User creates his wallet, passphrase is a mandatory option, Armory also asks for the third time verification of the passphrase and warns the user not to forget his passphrase (appropriate guidelines for G5). After this step, a backup window pops up with the options to print a paper wallet or save a digital backup of the wallet, also warns the user if he decides not to backup his wallet. Now the Bitcoin address is generated, however in order to see the final balance of the account, it needs to be online. User with less knowledge of air gap technique might click on the "Offline Transaction" button that offers a short documentation of the steps to be taken to sign a transaction. One point that has not been mentioned in the documentation is that user should click on "Receive Bitcoins" to generate the bitcoin address in the wallet file (G3,G4). On wallet window, there is the option to "Create Watching-Only Copy". This option is being used to copy the bitcoin addresses, not the private keys, for the online computer to display the balances.
Now user should install Bitcoin Armory and `bitcoin-qt` on the online computer to have access to the updated balances of his accounts. This time on the first run, user should click on "Import Existing Wallet" and chose to import the digital backup or watch-only wallet, Then chose the watch-only back up file that has been copied from the offline computer. After the application is done with syncing with the blockchain and scanning it for transactions that contains the imported bitcoin address, the balance is shown and the user has finalized his receiving address.

CT2, CT3.

In Airgap technique, There is no primary or secondary device in the sense that we have defined these terms. It's the offline computer that should be kept secure and disconnected from the internet, and there is the other device that should be online. Thus, the same steps (CT1) are required on any given new online device, to add the watch-only wallet and wait for it to synchronize with the blockchain.
Now on the online device with the watch-only wallet added, user should click on "Offline Transactions" on the main window. In should be noted that unless the wallet client is online, the "Create New Offline Transaction" is disabled. Now by clicking on this option, user will be asked to enter the receiving addresses, amounts and any comment on the transaction. There should be enough balance in the sending account for the offline transaction option in Armory to generate the transaction data. If everything is fine with the configuration, by clicking on "Continue" the unsigned transaction data is generated. User can now save these data on a file in a usb stick and should transfer this data to the offline computer. Armory has good documentation of the needed steps on each window (good use of G1). As also mentioned in this step's documentation, these unsigned transaction data has no private data and no harm can be done by the attacker whom captures this file, but only privacy issues that these data includes the sending address. Next step is on the offline computer to sign the transaction. By clicking on "Offline Transaction" and then "Sign Offline Transaction" , there is now the option to load the unsigned transaction data file. Armory would alarm the user to review the transaction information such as the amount and the receiving addresses in each step (Good behaviour for G5). By clicking on "sign" button, it will show the transaction details and waits for user approval. Next it will show the signed transaction data and user can save it in the file. It should be noted that on the side of the window it will show if the shown data is a signed or unsigned transaction (follows G4). Once again this file should be transferred to the online computer and get loaded in the offline transaction window. Now the difference is that the "sign" button is disabled but the "broadcast" button has been enabled. By clicking on "broadcast" it will once again show the detail of the transaction for the user to review. This would be the last window for the transaction and by clicking on "continue" button it will broadcast the transaction to the bitcoin network and the bitcoins has been sent to the receiving address.

CT4.

Armory gives lots of options and opportunities for the user to back up his keys. In the time of creating the wallet, there are multiple windows and alerts try to convince user to back up his wallet, either in digital format or paper wallet even if user decides not to back up his wallet at the start, he would have the chance to do so afterwards by going to his wallet and clicking on "Backup This Wallet". On the backup window, There are options to back up a digital copy or paper copy and also more options that are mostly more secure ways to backup the wallet and would fall outside the scope of this paper. By clicking on the "Make Paper Backup", The paper backup is shown to the user containing a Root key that consist of 18 sets of 4-characters words and a QR code with the ability to print the page. To restore the paper wallet backup, on the main page, user can click on "Import or Restore Wallet" and select "Single-sheet Backup" option, now he will be asked to input Root Key from the paper wallet, and on confirming the Root Key the wallet would be restored and added to the Armory wallet client. That being said, there is the "Digital Backup" option that would backup an unencrypted version of the wallet file and should be kept secure. The digital backup is an easier way to backup and recover as it the recovery would be just importing the file in the "Import or Restore Wallet" window by selecting "Import Digital Backup or watch-only wallet" as we did for the online

computer in CT1. Armory also gives this option to do the import procedure for testing the backups to see if there is no error in the backup file.

6.2.4 Offline Storage

There are different methods to use 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 Air gap technique. Bitaddress [24] is a well-known Bitcoin paper wallet generator in the Bitcoin and Open-source community. Also as it would be the first result in searching "Bitcoin Paper Wallet" in google, we would assume the new user would go with this site.

CT1.

On entering the site, The user is being asked to move the mouse or enter some random characters in the textbox to make the random value to generate the bitcoin address. When the randomness is enough it will automatically redirects to the main page that shows the receiving bitcoin address and it's paired private key. The public key (Bitcoin address) is labeled by a green "SHARE" text and the private key by a red "SECRET" text that has to be kept secret. By clicking on "Print" Button the paper wallet is printed and is finalized. There is a short but complete documentation of the steps user has to take to achieve his desired functionality. So far user has his own bitcoin receiving address and as it is mentioned in the documentation to check his balance he has to go to block explorer sites such as blockchain.info and search for his Bitcoin Address. Although it has been mentioned to keep the private key secret, there might be users whom do not fully read the documentations and might expose the private key to public [25] (G5). The terminology and interface used in this site is really simple and user-friendly.

CT2, CT3.

For this technique, as the keys are printed on a paper, there is no difference between primary and secondary device. To send funds from a bitcoin address that has been stored on a paper wallet, as it is mentioned in the documentation, user has to import his private key in one of the wallet clients available such as Armory(6.2.3) blockchain.info online wallet (6.2.6). For instance, on blockchain.info wallet, user can input the private key address and use the funds, however in this case the funds' changes might be sent to his other addresses included in his online wallet and he might need to make another bitcoin paper wallet to send the remaining funds to. To do so, after making an online account in blockchain.info (6.2.6), user would go to "Import/Export" tab and there is an option "Import Using Paper Wallet, Use your Webcam to scan a QR code from a paper wallet" that he can scan his paper wallet with the webcam and import the address into his online wallet. It is also possible to type in the private key in the "Import Private Key" text field. After this step, the address now is hosted on the online wallet and is the same as CT2 in Hosted wallet client (6.2.6).

CT4.

The printed copy of the private key is the only way to hold on to the funds stored in the paired address. User has to acknolewdge this (also mentioned in the short documentation), and save a copy of his keys in a safe place such as a personal vault, to be able to recover the funds in case the first copy is destroyed. In case of theft, there is no such a way to recover the keys.

6.2.5 Password-drived key

We would use the implemetation of brain wallet in brainwallet.org, as it is the most popular and complete implementation on the time of writing.

CT1.

On entering the site, There is an already generated address with the passphrase "correct horse battery staple". It should be mentioned that this is the default value and should not be used for personal use as the private key is publicly available[26], this is a violation of G5. Now user has to type in his own passphrase and make sure his passphrase is not a commonly used phrase or anything that could be brute forced by a dictionary attack [27] as this passphrase is everything that is needed to access the funds stored in the bitcoin address. On entering the desired passphrase, the public and private keys are displayed in the same page. Now user has his own bitcoin address, and to check his balance, same as paperwallets, user has to go to a block explorer service and searches for his bitcoin address (nothing is mentioned in the site and this is a violation of G1,G2). There are lots of fields displayed that is not meaningful for the new user and might confuse him (violates G6 and G7).

CT2, CT3.

In any device, User can enter the brain wallet site and enter his passphrase to recover his keys. Although to authorize a transaction, user should import the private keys into another wallet client to be able to spend the funds, the procedure is the same as it was used to import paper wallets.

CT4.

As long as user knows his passphrase, his keys are safe. That being said, one of the cons of using brainwallet is that if the site stops its service, the algorithm used to derive the keys from the passphrase is no longer available, in this case the best practice is to save the private key in a secure storage to be able to access it in case the site no longer worked.

6.2.6 Hosted

There are different online services that offer online hosted wallet clients to users such as blockchain.info, coinbase [28], same as previous sections we would work on todays popular online wallet, blockchain.info wallet.

CT1.

User needs to go to the site[29] and start a new wallet with

[24] `https://www.bitaddress.org`

[25] A Bloomberg TV Host Gifted Bitcoin On Air And It Immediately Got Stolen`http://www.businessinsider.com/bloomberg-matt-miller-bitcoin-gift-stolen-2013-12`

[26] How to steal bitcoins `http://www.palkeo.com/code/stealing-bitcoin.html`

[27] Bitcoin âĂIJBrainwalletsâĂİ and why they are a bad idea`http://insecurety.net/?p=866`

[28] `https://coinbase.com/`

[29] `https://blockchain.info/wallet`

his email address and a minimum of 10 character long password. There is a warning on the bottom of the form that on password loss the account is unrecoverable (Good use for G5) . The first pop up window titled "Wallet Recovery Mnemonic" would show a phrase that can be used to recover the account in case the password is forgotten. After this step user can login to the site and view his bitcoin address and the balance it holds.

CT2, CT3.

One of the advantages of using hosted wallets is that it is accessible from any browser on any device. User can go to the site, log in and he has access to his funds. To do a transaction, user would click on "Send Money" tab of the main page, and there is a simple easy to use interface that asks for the destination bitcoin address and the amount of Bitcoins to be sent. Although in case of errors, some of the errors might use a terminology that is unfamiliar for the user such as for insufficient funds this error would be shown "No free outputs to spend" (violates G4 and G6).

CT4.

There are two methods of recovering, one is to use the phrase given on account registration, and the other is to use the backups. To use the wallet recovery mnemonic, on the login page, there are different options if anything has been forgotten (identifier or alias for username and/or password). By clicking on the "Recover Wallet" option, it will ask for the mnemonic phrase and the email address or mobile phone number associated with the account to send the new confirmation details. The better recovery option is to have backups and import them in case recovery is needed. To do so, in the main wallet page, user has to click on the "Import/Export" option and Export either encrypted or unencrypted backup. It should be noted that unencrypted backup do not need a password to be read and should be kept in a secure and safe storage. There are different options for the unencrypted backup procedure that could confuse user and might result in unrecoverable back ups (G5 and G6), also the back up would be shown on a text field that the user has to copy and paste in a text file to be able to save it on his computer (G2,G3 and G7). To restore the backups, He need to go to Import Wallet option and copy and paste the backup file in the text field, in case of having the encrypted backup, the previously used password is needed.

7. DISCUSSION

...

Some setups of the bitcoin wallet clients might not work as they are supposed to in a non-default settings. Specially setups such as Bitcoin-QT and Armory that need to have the access to the complete Blockchain files in local computer. As the blockchain size increases (21 Gigabytes in May 2014), it could be a hard task to keep all the files in the computers hard drive and user might have the need to store them in an external storage such as an external hard. One obstacle for normal user would be that as soon as there is the need to use a setting other than the default one, user has to go through command-line switches run the application. For setups such as Armory in Airgap, most of the non-default settings would fail, such as standalone Bitcoin-QT installation or non-default Bitcoin-QT data directory to store the blockchain file in an external storage. All these would be in usability issues that should be overcome to have a better user-friendly Bitcoin wallet clients.

8. 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?

9. RELATED WORK

Blah blah blah.

10. 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