3-factor Authentication

James Rhodes, Glen Low, Adam Brady

June 21, 2011

1 Problem Definition

With the recent breaches of BitCoin exchanges, it brings to light the requirement for decent security in these exchange systems. Coupled with the built-in security of BitCoin requiring the use of a private key to sign sending transactions, within this paper a 3-factor authentication system is proposed, allowing exchange users to safely place orders on an exchange. Even in the event that the exchange is breached, the user's computer is breached or both, an attacker should not be able to place sell or buy orders without significant effort.

2 Explicit Requirements

1. In the event the exchange is breached, the attacker should not be able to withdraw funds or place trades.
2. In the event that user's computer is breached or otherwise monitored, the attacker should not be able to withdraw funds or place trades on the user's behalf.
3. In the event that both the exchange and the user's computer is breached, the user has not previously placed trades while under surveillance and the user has set up SMS authentication, the attacker should not be able to withdraw funds or place trades on the user's behalf.

3 Account Architecture

In a secure BitCoin exchange, each user is given a BitCoin wallet on the server, as well as there being a global ”Active Transactions” wallet which pools all of the BitCoin funds together in active sell trades. In the event that the exchange was fully breached, an attacker would be limited to withdrawing coins in this ”Active Transactions” pool.

The actual process from user A to user B in a trade would proceed something like:

A (local) ->A (exchange) ->P (exchange) ->B (exchange) ->B (local)

where ’local’ indicates BitCoin wallets stored outside of the exchange.

3.1 Double Spending

Often a concern here would be the issue of delays between transactions, however, as confirmations are used only to perform double spending, once we know that the funds in A’s exchange wallet are confirmed, we implicitly know that there will be no double spending all the way from A’s exchange wallet, through the pool and to B’s exchange wallet. This is because the users themselves never hold the private keys to the exchange wallets, rather they provide a master key by which the server can use it.

4 Master Key

The first element in preventing the unauthorized reading or writing to fields by an attacker is the use of a master key; provided by the user on each request and never stored on the server (in anywhere other than memory while the page is executing). This master key acts somewhat like a password, but rather than one-time authorization, it is used as the encryption key for not only user-specific information, but also to encrypt the user’s exchange wallet.

By encrypting the exchange wallet with a master key, we prevent attackers from making transactions on the user's behalf, including that of the exchange owner (unless the owner has previously logged the master keys, however doing so would defeat the purpose of the owner setting up a secure exchange in this manner). This prevents compromises of the exchange affecting a user's BitCoin balance, but also prevents rollbacks of trades (the exchange owner may emulate a rollback using the transaction pool, but they can't withdraw or decrease the BitCoin balance of a user without the user providing authorization to do so).

4.1 Considerations

It is important to consider the security issues surrounding the master key given it's importance in decrypting the private key for the wallet. First, the complexity of the master key should be enforced, minimum 15 characters (no maximum), upper and lowercase letters, numbers and special characters. Users of the exchange should not write down or store the master key on their computer either; rather the only known location of the master key at any given time is either in the user's head, temporarily in the client computers RAM while the user is logged into the exchange, passing through the SSL connection to the server, or temporarily in the server's RAM while the user's transaction is being processed.

As the master key remains on the client's computer for the duration of the session (i.e. stored in the browser's cache), it is important to also consider the security implications client side. These are discussed in Appendix A.

5 Differencing Code

To prevent a compromise of the user's computer and the exchange allowing an attacker to automate transactions on the user's behalf, SMS code verification is used with a rotating PIN number to ensure that the user is physically authorizing the transaction.

However, it must be considered that for a physical authentication to be useful, it must also prevent decryption of the private key so that an attacker can not bypass this requirement; the rotating PIN number on the other hand provides uniqueness to the confirmation, but it is impossible to use a changing value as part of the encryption key.

For this reason, it is not the exact PIN code sent that is the concerning value, but rather the PIN code sent back that is important to the server. When the PIN code is sent to a user's phone, they are not required to enter it back into the computer; instead they are required to enter a transformed version of this PIN code into the machine. This can be as simple as ”add 2000” to the 6-digit PIN code; the server can then determine the difference between the sent PIN code and the received PIN code and use that differencing code as part of the master key with which the wallet's private key is encrypted.

Importantly, an attacker viewing the user's computer will not be able to determine the original PIN code sent (it will appear on the phone) and so they will be unable to determine the correct differencing code for the account; even if the attacker was to capture the master key on the user's computer and then breach the exchange, they would still be unable to decrypt the private key without the differencing code.

6 Conclusion

In this paper we have outlined a security mechanism by which exchanges can provide secure trading services.

In the event that the exchange is breached, the user's coins are secure from unauthorised transactions; in the event that the user's computer is breached (while using physical authentication), the user's coins are still secure from unauthorised transactions. Only in the implicit situation where an attacker breaches the exchange, begins logging master keys, and then attacks the user's computer to obtain a differencing code (which can be done if the attacker knows the PIN code) can the user's coins be spent without authorisation.

In addition, using an implementation of one wallet per user has the added advantage that all transactions through the exchange are verifiable (it is possible to ensure the exchange holds all the BitCoins they advertise to).

A Master Key Security Considerations

The master key in a proper setup, will only ever be stored or transmitted in the following situations:

1. Stored in the user's memory.
2. Entered into the web browser's text field.
3. Stored in the web browser's cache.
4. Transmitted over SSL to the exchange.
5. Stored in the exchange's RAM before being discarded.

The problem lies when the master key is stored by the browser's cookie or local storage cache (in this case, we are invisioning a system where the user enters their master key when they sign into the site, and it is remembered for the rest of the session). In order to prevent XSS and CSRF attacks, the implementor must be careful that the master key when stored, is only ever readable by a process which transmits it to the server.

At this time, we are still investigating a reliable solution to this problem and may issue another paper outlining a solution. In the meantime, implementors should be aware of the possible XSS and CSRF issues arising from the storage of the master key in cookie or local storage.