Stellar off-line fund exchange algorithm

(c) 2015 by sacarlson sacarlson_2000@yahoo.com

What are we trying to do?

The goal of this project is to create a method to allow making safe secure Stellar asset transactions without continuous Internet connectivity. This would allow remote locations without full coverage Internet services to still perform asset transaction between people or businesses. As an added benefit transactions without the Internet can be performed very fast in theory less than 1 sec.

What hardware would be needed:

To perform such an action would require hardware of at least a small low powered android phone or laptop computer.

Basic preview of the algorithm

In short what this algorithm does is use an automated trusted witness that will verify that funds of a target account are locked from being used to send money to anyone accept one other preselected added signer for some window of time. Before a transaction is made the receiver of funds, they will do a check of the signed witnessed document to verify funds are available in the account, and locked for a time window that they know they can get to a live Internet to collect it, and that they are the only other signer on this account that can possibly collect the funds. This trust bot or many of them will be on a list of known trust bots much like what we presently use in https certificate websites.

Detailed steps performed in the background

Note: there is an improved method using dzhams posted idea shown at the botom of this document. This now remains partly as a historic document. At some point this may be revised to incorporate the new improved method.

As far as the payer and payee is concerned, most of this would be done in software, so as complex as it may sound here to them it is much the same as if they are connected to the Internet to do transactions. In this list of actions below, we will be using terms of sequence number or seq in a relative sense of when these actions began and ended. In reality this sequence number is the sequence number from the records in the stellar database when the witness trust bot recorded and signed it. We will also be involved with two accounts and one signer keypair of the Witness Trust bot that is permanently locked to never be able to transact funds on it's own account. We will name these “A. master target account ”, “B. Witness Trust keypair (locked)”, “C. payee target account ”,

details on the Witness Trust keypair account B: This account is special in that it can't even make transaction with it's own account. This is done by setting it's thresholds settings to master_weight 0, low 0, med, 0, high 0. with this setting the account can be used as a signer but can't transact any of it's own funds.

seq 1 start with a fresh unlocked active account A

seq 2 - 3 the master of the target account A adds two signers to this account
one being the planed witness trust bot account keypair B and the other being the target payee account C

seq 4 the master of the target account A now changes the account thresholds to master_weight: 1, low: 2, med: 2, high: 2. This account is now locked with the master unable to send funds to anyone other than the payee account C that can sign for it. Part of the trust bot pledge is to not accept any sign-able funds received, to guarantee this the trust bot seed key account is not an active account just a signer key set.

The target account can now be funded with assets planed to be spent for the lock time window

With the funded locked account the master of the target account A will now have the account witnessed and signed by the witness trust bots keypair B. The master of the target account A will send his public address and a time-bound timestamp of when he wants the account to be allowed to again transact his own funds. The Witness Trust bot will return a signed data packet that includes a snapshot of all the info seen on the target account at this time from the stellar network database with the assets it now holds as well. The Witness Trust bot will also return a signed timebound transaction for seq 5 that the master of the target account can sign and use after the timebound window in the event no other transactions are made to the target payee before that time.

Some random time passes before the timebound expire

The master target account A holder gets thirsty and wants to buy a beer form a beer vendor that holds they they keypair seed of account C.

The master target holder creates a signed transaction on seq 5 for the payment of the beer and sends this with with witnessed data package from the witness trust bot for the vendor to analyze.

The beer vendor analyzes the transaction and signed witness document to verify that the locked funds needed for the purchase are available and that the time window is within a range that is acceptable for him to get to an active Internet to submit the transaction on the stellar network to collect the funds before the timebound expires.

If for some reason the master target A holder never finds the beer vendor and the timebound window expires. He would submit the transaction give to him by the witness trust bot above to unlock the unspent funds back to default thresholds of master_weight 1; low 0 , med 0, high 0. this unlocks the account allowing the master target A to transact normally with just his single signing.

If the master target A holder doesn't spend all the money but just part of the funds to buy beer, he can go back to the witness trust bot to have it issue him another timebound transaction with the needed updated sequence number seq 6 now included. If the timebound window has expired the master can then submit this transaction to the stellar network to get his remaining funds unlocked.

Details of what the payer and payee would see and do:
people looking at the above document may think this is too complicated for people to deal with. But remind you it's not people that perform most of the actions. It's software in the background that does most the hard stuff above. So here to clarify this is what humans could see when they transact.

A. the android phone auto transfers funds from a base account to smaller transit accounts and locks the funds into each of these accounts each morning before the master leaves to go out. These accounts single payee points can be collected over time be the phone listening for bluetooth broadcasts that will be picked up when within the proximity of an active vendor that the master may walk by or even purchase things from everyday. They can also be entered manually with a QR code scan of a person or vendors account before a purchase or payment is ever made.

B. the master gets thirsty and see's a known beer vendor on the list of accounts the master holds

C. The vendor enters the purchase into his android vendor app or PC cash register app by scanning the beer bar code with his phone or PC scanner. His software looks up the price of the beer and broadcast the purchase on bluetooth RF for the customer to pick up.

D. the customer brings up his android app to pay his bill. It scans for the broadcast from the vendor with the payment amount and a short description of each of the purchase items and a total. The master glances at the purchase and hits the OK button to send the funds to the vendor.

E. The vendor see's the payment packet from the master buyer and his android or PC app shows green as the funds have been accepted and cleared.

F. the master picks up his beer off the vendors counter, leaves the store and drinks his cold beer.

Person to person transaction

long story short there is no difference seen by the user making a transaction whether making it in the on-line mode or off-line mode but I already wrote this so go ahead and read on if you want.

In this case we have two people one person A (Apple) that owes the other person B (Ben) some money for buying him a beer the day before and he forgot his phone that day so didn't already pay him.
At this point there are two options as to who starts a transaction in this case Ben opens his android app in receive payment mode, enters in the box the amount he wants to be paid from Apple. He hits the broadcast button and his phone then starts sending out a signal. Apple can now open her phone app in the send funds mode if she didn't already and from there she can see the proximity broadcast from Ben asking for payment. In the payment field will also be identity info she can recognize including his name and even his id picture can be seen if she already had that stored in her trust list. She would also see the amount that Ben is asking for. All she needs to do is touch Ben's box as the destination field and hit the OK button and Ben will see on his screen that he received the transaction. The End.

There are other options as to how the above could transact including sharing QR code between one or the other. In reality the transaction would not be seen any different than a normal wallet transaction that had Internet connectivity as far as what Apple and Ben see. The same wallet software can detect that it has Internet connectivity or not and decide weather to use standard stellar transaction method or off-line mode. The off-line mode or the on-line mode can be disabled in the app if desired in the options of the app. If Internet is intermittent in respect to what the android phone sees, the app can also start sending off-line transactions it collected when it was off-line. It could also detect when the trust gateway starts failing to make some of the transaction redirects and auto disable any more transactions to that trust gateway.

How many payee account points or trust gateway transfer points will we need?

This is unknown and depends on the individual and his environment. As I now realize that what I spoke of before about trust gateway transfer points won't work. It would allow for double spending. So this method even with the improved dzham method, is limited in that you would be required to have as many prelocked funded account points as you as an individual thought you would need or use in future dealings with vendors and producers. Until some method is created to allow groups of people to be added on the fly to accept funds off-line I'm not sure this method will ever be feasible for most people, but it may be of use to some.

When will we have this?

As all this above text sounds like future talk in reality much of what is explained above has already been written. The new stellar-core network is already running and we already have written the infrastructure for multi sign transaction used in the above in what is called the multi-sign-server and muli-sign-websocket software . The witness trust bot used in this algorithm already exists and is now running on the stellar-core testnet for the world to see. We have already performed all the above explained transaction in tests to verify how it can be done and have provided published examples. So we already have most of it complete. The only part left is to write the android application to use the already existing infrastructure. So I would expect it should be possible to complete in less than 6 months even for a single programmer like myself to develop. For more details about multi-sign-websocket and other developments see https://github.com/sacarlson/stellar_utility/tree/master/multi-sign-websocket and https://github.com/sacarlson/stellar_utility

Updated better Payment Channel method from input from dzham:

```
Payment Channels in Stellar
    producer, consumer
setup:
    - create a joint 2-of-2 multi-signature account, with consumer and producer
as signers
    - producer sends consumer a tx that sends all the funds back to consumer,
valid only after a certain time
    - consumer funds account w/ a certain amount

while running:
    consumer consumes a service
    consumer sends producer a tx that:
        * sends the balance to the producer
        * sends back the change to the consumer (if asset other than XLM)
        * merges account w/ consumer
to cash out, producer signs the most recent tx, and submits it to the network
```

This to me looks like a better way than my original steps shown in **Detailed steps performed in the background** above that requires the trust bots signature to later unlock the funds. But dzham's method has the same weakness as the original method. It won't work with a trust gateway redirector as the producer doesn't have the key seed of the gateway redirector account to issue change back. So with this method you would be forced to hold prelocked accounts with assets for every vendor and producer you planed to deal with in the future. You could never hope to deal with any new vendors you run into on the streets on a given day that you already had prof of prelocked funds that they would accept. But this problem is also the same in my original method so this method is still better than my method. Nether this one nor my original method could use the gateway redirector other wise it would allow for double spending. So both would be required to have prelocked funds for every producer and vendor you planed to deal with off-line in any future dealings.