

Hierarchical Deterministic Wallet

sanemat {AT} tachikoma.io

経過

やってみてるけど、俺たちの戦いはこれからだ!ってかんじ。できてるか出来てないか微妙なところですね。資料作成時点ではできてない。作る過程でいろいろ調べた。

自分の理解

あやふやなところもあるから間違ってたら突っ込んでね

アドレスの固定

さっき bitcoin 専門社内おやつ商店の話で、1 アドレスを紙に書いてしまった話をした。



図 1: おやつ商店

ビットコインアドレスって、使い捨てにしたほうがいい。取引金額なんてバレてもいいじゃんってはじめは思う。こちらのセキュリティ的な話。相手のセキュリティ的な話。51%攻撃受けやすくなる。

表示側もワンタイム生成のアプリにすべきですな。

着金したかどうかってどうやってわかるんじやろう hdwallet にして、拡張公開鍵だけ入れておけばいいのでは？ 一見良さそうに見えた。

アプリ側の実装によるのかもしれないが、copay android アプリからだ、拡張公開鍵だけで送金「依頼」のようなものが出せた。wallet client 側の見た目の数字は減って、秘密鍵を持っているアプリに、「この送金を verify しますか y/n」のようなダイアログが出て、y を押したら送金処理になった。もちろん n を押したり、無視したりすれば送金されないが、ちと、ちと。

HD Wallet と BIP0032/BIP0044 の説明

HD wallet(Hierarchical Deterministic Wallet) 階層的決定性ウォレット
mastering bitcoin を読んでくれ! 以上

ランダムウォレット (非決定性ウォレット)

ウォレットはランダム生成した秘密鍵の束

決定性ウォレット

ウォレットで共通の seed から一方向ハッシュで秘密鍵生成

HD wallet(Hierarchical Deterministic Wallet)

seed から master key 生成、そこからツリー上に child keys, grandchild keys と生成

(mastering bitcoin からコピペ)

ツリー構造に情報付加できるユーザーが秘密鍵に触れることなく
公開鍵を生成できる

気になるところ

拡張公開鍵が chain code を含んでいる

拡張公開鍵って「公開鍵」って言いつついろいろ出来てしまうので、ふつうに ssh の公開鍵みたいに全世界に公開していいの？ って問題意識とも関連してくるんだけど

拡張公開鍵は chain code を含んでいるため、もし子秘密鍵が知られているまたは漏洩してしまった場合、この chain code を使ってその他すべての子秘密鍵を導けてしまうこのリスクへの解決策として、HD ウォレットは hardened child key derivation 強化子公開鍵導出 関数を使っているもし拡張公開鍵の利便性を使い、しかも chain code の漏洩リスクを回避したいのであれば、通常の親(親公開鍵)ではなく、強化された親(親秘密鍵)から拡張公開鍵を導出すべきです。ベストプラクティスとしては、マスターキーの1階層目の子供を常に hardened derivation を通して導出されるようにしておくことが良いでしょう。

なんか小さくベストプラクティスを書いてあるが、一般に使われるクライアントは、みんなこれやってると思っていいんだろうか...

UTXO

UTXO (unspent transaction output)

だいたい wallet で wallet.balance() とかでとれるやつ

ここまで来てやっと意味がわかった。wallet は未使用含めて bitcoin アドレスを持っていて(作成できて)、それには path に 0 とか 1 とか 2 とか順に名前がついてて、path は小さい順から使う暗黙ルールがあるらしい(未確認) なぜなら、未使用が 20 個続いたら探索打ち切り、などがウォレットアプリ側で実装されているので、条件に依っては、あるウォレットで見つからなくなってしまう bitcoin が、アドレスの使用数が進んだら出てくることもあるようだ(未確認)

ツリー状のはどうたどるんだ？ これが bitcoin-explorer って名付けられてるタイプのライブラリの仕事なのかな(未調査)

まとめ

俺たちの戦いはこれからだ!!

ツールを作ると学習が進む (作れてない)

ヨタ話

プログラム言語は何で実装見てたんですか

Nodejs です

golang 学習がてらやろうとしたら全然そんな余裕がなく、土地勘のある ruby と nodejs だと nodejs にライブラリが結構あった。bitcoin core が C++ だし、C++ が一番楽なのでは。俺書けないけど。それか python。俺書けないけど。btcd が golang だから golang もよさそう。俺書けないけど。ほんとは golang で bitcoin プロトコルしゃべろうと思ったんだ...

nodejs でなんとなく見たのは

- bitcoinjs-lib
- bitcore-lib (bitpay.io)
- bcoin (purse.io)

この順番がリリース順で、API の洗練され具合も下が良い。あと、full node か SPV node か、web service との連携具合とかが違う。なんだけど、npm ソムリエ的には、おススメは上から下。

ただいずれにしてもそこまでアクティブではないように見える。一応メンテはされてる。