Deterministic Wallets, Their Advantages and their Understated Flaws

by **Vitalik Buterin**

cIf you have been following the last year of progress in Bitcoin wallet development, you will have likely already heard of one of the latest trends in backend designbe: deterministic wallets. Unlike old-style Bitcoin wallets, which generate new Bitcoin addresses and private keys randomly as needed, in a deterministic wallet all of the data is generated using a specific algorithm from a single seed. That is to say, if you write down the seed to your deterministic wallet, and then after six months your hard drive gets corrupted and the wallet unrecoverable, you can simply create a new wallet using the same seed and all of the addresses and private keys from your old wallet will come back again exactly as they were before. This trend in wallet

implemented or is planning to create one.

However, deterministic wallets in Bitcoin do not stop there. In fact, the latest deterministic wallets go beyond the simple design described above and have two key properties that are heavily advertised by their developers. The first of these properties is the concept of a "**master public key**". A master public key is a key that can be generated from the wallet's master private key (either the same thing as the "seed" or a derivative of it) that has the power to generate all of the addresses in a Bitcoin wallet, but none of the private keys. Thus, someone with access to a master public key can look at the balance of a deterministic wallet, but cannot actually spend the balance because they have no way of generating the private key corresponding to each address. The second property is **hierarchy**: the private keys that you generate from a master private key are themselves master private keys and can in turn be treated as deterministic wallets in their own right.

How do Deterministic Wallets Work?

As it turns out, there are two major types of deterministic wallets currently in use: Electrum wallets and BIP32 wallets; they use a very similar algorithm, allowing them both to have the master public key property, although the BIP32 wallets go further by also including the hierarchy property – Electrum wallets are designed to only go down one level, although one certainly could extend the Electrum protocol to make it hierarchical as well.

The master public key property is perhaps the more surprising feature of deterministic wallets, and will be explored in detail first. The reason why it works is that Bitcoin public keys – not quite the same thing as Bitcoin addresses but a closely related form – can be added and subtracted just like normal integers can (although, notably, you cannot multiply two public keys together), and thus the same arithmetic operations can be done on two "levels" – to generate private keys, the arithmetic is done on the level of integers, and to generate public keys it is done on the level of public keys.

(technically, a hash) of the index and the master public key. Then, simply add the master private key and the offset together. To calculate the public key at index i, calculate the offset in the same way, convert the offset to a public key, and add the master public key and the offset public key together.

Here are a few examples using Electrum wallets, done with my own **pybitcointools** library. First, we generate a master private key and master public key from the seed:

```
> seed = random_electrum_seed()'afc3eef71d96c468ca52b437c385a621'>
mprivkey =
electrum_stretch(seed)'5df10c922a1c7888b5c3a5a7106e72576f09c17f0993f4f2c
mpubkey =
'04'+electrum_mpk(mprivkey)'04fd6d91db1bdfc231116fd7d44c61a02e032b38b90a
```

Now, we generate private key index zero:

```
> offset =
dbl_sha256('0:0:'+mpubkey[2:].decode('hex'))'429251ad9607fd39040072d23f5
priv0 =
add_privkeys(mprivkey,offset)'a0835e3fc02475c1b9c418794fc247a732bbabd16c
```

Now, the public key:

```
> pub0 =
add_pubkeys(mpubkey,privkey_to_pubkey(offset))'04d96f3a8ebb0de48a98a5d77
addr = pubkey_to_address(pub0)'14EkQ9qsKxWKiBJm5f7mT7ozSKKZbQoZGS'
```

And, just to show you that the math checks out:

```
>
privkey_to_pubkey(priv0)'04d96f3a8ebb0de48a98a5d77003c1d3ed5a36aae3eb20e
```

We can repeat this with index 1, index 2, etc; you can try it yourself with your own Electrum wallet if you have one. The takeaway is this: you can safely give put your master public key in an insecure place, or even give it out to third parties like

Hierarchy

Now, on to the hierarchical wallet property. This one is, once again, best described by simply showing it in action:

```
> w =
bip32_master_key('qweqweqweqweqwe')'xprv9s21ZrQH143K2KhRQVuMqhz798mvW89J
w0 =
bip32_ckd(w,0)'xprv9uyTuGongdyZAMxZ2euUBbpsAdtE2nxFBmcQn89UT4ZyzrMg5TXD7
w000 =
bip32_ckd(bip32_ckd(bip32_ckd(w,0),0),0)'xprv9zL8JVf2Us8VKFYoi3A8F3LSFuH
```

The main use case for which this feature is advertised is in hierarchical organizations: the treasurer of a company might have control over the root private key of a BIP0032 wallet, and then hand off a “child” seed to each of the company’s departments who will then use that seed to operate their own wallet. The treasurer will have the master key to everything, but each department will only have the key to their own part of the funds.

And, of course, BIP32 has that same master public key property as Electrum, but even stronger:

```
> wp =
bip32_privtopub(w)'xpub661MyMwAqRbcEomtWXSNCqvqhAcQuas9NoDxJcizDuXtuyiKD
wp000 =
bip32_ckd(bip32_ckd(bip32_ckd(wp,0),0),0)'xpub6DKUi1BvKEgnXjdGp4h8cBHAow
wp000_2 =
bip32_privtopub(w000)'xpub6DKUi1BvKEgnXjdGp4h8cBHAow8HMqR1ziuTKHdpEx8Usc
```

Thus, a BIP32 master private key can be thought of being at the top of an infinitely descending tree, capable of recovering every private key below it. And a BIP32 master public key is just the same, except it can only recover public keys and addresses. Another metaphorical way to think about it is in terms of the private keys sitting at the canopy level of a rainforest, and the public keys on the ground below them. You

ground you can't get back up (tree-climbing monkeys that can go up from the public key ground level to the private key canopy are, at least for now, purely theoretical).

An Understated Problem

From the descriptions we saw above, you likely understand that deterministic wallets have two properties. First, you can go from a parent key to a child key, but not in reverse. Second, you can give out your master public key with no risk to your funds – only your privacy. And this is how nearly all people, at least those technically skilled enough to know what a deterministic wallet is, view BIP0032 wallets today. The model of a company, which hands out child private keys to departments and master public keys to accountants and auditors, has come to take a central place in the mythology around the promise that BIP0032 wallets potentially hold. However, as we will see below, this description of hierarchical wallets is fatally flawed.

The problem is this: although you certainly can securely hand out child keys with no risk to the parent key, and you can hand out master public keys with no risk to the master private key, you cannot do both at the same time. The exploit for when that situation does arise is actually quite simple, and can be done with two lines of pybitcointools code. I will use Electrum in this example, since Electrum wallets are more transparent. These are the same master public key and child private key I created above:

```
>
mpubkey'04fd6d91db1bdfc231116fd7d44c61a02e032b38b90aad419ecf75acf501eebc
priv0'a0835e3fc02475c1b9c418794fc247a732bbabd16ca4ef38983923bd3dcd8177'
```

As we saw above, the first private key is calculated by a formula which can be summarized as `mprivkey + calc_offset(mpubkey,index)`. So, what do we do? If you look for it closely, the answer is surprisingly obvious:

```
mpriv2 = subtract_privkeys(priv0,offset)'5df10c922a1c7888b5c3a5a7106e72576f09c17f
```

And tada, we get the master private key back. Now, we can go ahead and pilfer all of the other addresses in the wallet, even those which the wallet's owner never intended to touch. I even included a command in pybitcointools to make this more convenient for you:

```
> crack_electrum_wallet(mpubkey,priv0,0)'5df10c922a1c7888b5c3a5a7106e72576
```

BIP32 has the same vulnerability:

```
> wp = bip32_privtopub(w)> w0 = bip32_ckd(w,0)> crack_bip32_privkey(wp,w0)'xprv9s21ZrQH143K2KhRQVuMqhz798mvW89J1aJMWEKNf
```

In the interests of fairness, it is important to note that this is not a sudden new zero-day vulnerability discovery; many Bitcoin developers have known about this for a while. However, given the intuitive understanding of hierarchical deterministic wallets that many people implicitly promote, including the idea of handing out child private keys to organization departments and master public keys to auditors, this has a great potential to result in a security breach. It may only be a matter of time until a large organization decides to actually adopt a hierarchical deterministic wallet to protect its Bitcoin funds, and suddenly finds a collusion of one of its department heads and an auditor running off with the entire company funds. So the obvious question is: can this be fixed? The answer seems to be no; because the only operations that can be done with public keys is adding and subtracting them, the only way to implement a deterministic wallet with the master public key property is using the "offset" mechanism described here. If this is indeed true, then raising awareness is the only solution, together with a change in BIP32 representation and in clients to make it clear that master public keys and hierarchical wallets do not mix.

three wallets down some particular child key derivation path. Then, an auditor can have one of the three master public keys, and search the blockchain for transactions whose script contains public keys generated from that master public key. The solution is complex, not supported by any existing client, and far from perfect, but something like it seems to be the only way to get around the issue. In most cases, however, simply not handing out the master public key may be the better approach.

So what is the future of deterministic wallets? At this point, BIP0032 is arguably as far as we can go; there are no known tricks in elliptic curve math that haven't been exploited yet. One obvious upgrade might be BIP0032 multisignature wallets, combining BIP0032's hierarchical deterministic magic with an advanced feature in Bitcoin that allows you to send bitcoins to an address that requires two out of three given private keys to spend the funds. Another further direction is brainwallets. The two current competitors for memorizing a Bitcoin wallet are (1) choosing a password and using the password or a hash of the password as a seed, and (2) randomly generating a seed and converting the seed into a passphrase in a way that can be reversed. The way that both approaches are implemented is currently somewhat flawed – the standard implementation of the first approach does far too little against brute force attacks, whereas the standard (Electrum) implementation of the second approach is too difficult to memorize – studies show that passphrases like "glow date cost bloody curve wheel cousin picture ring finally bubble press" are **no easier to memorize** than random strings of characters of an equal security level, and they offer no protection against forgetting one or two words. These are open problems – if you are a Bitcoin developer, you personally have the opportunity to come up with and standardize a solution.

#AUDITORS #PROPERTIES #DEPARTMENTS #FUNDS #ADDRESSES #ELECTRUM #BITCOIN #KEYS #WALLETS

Recommended

Bitcoin Price Analysis: Blowing Through Support Levels on the Way to $3,000

Bitcoin continues to tumble lower and lower as it struggles to claim any footing in the market. It's down almost 50% in three weeks and it's showing very little sign of stopping. It's currently clutching onto the $3,500 values but it doesn't look like it can hold on much longer.

BITCOIN SCHMITCOIN

Op Ed: SEC's Latest Declaration Creates Legal Minefield for Digital Assets

This broad, authoritative declaration is not unexpected, as, to date, the SEC has stated that all digital assets — regardless of whether they function as alt coins or utility tokens — are securities at least initially and, thus, subject to its jurisdiction.

HUHNSIK CHUNG AND NICHOLAS SECARA

Tax accountants and firms that specialize in cryptocurrency will emerge to capture and service this market. The first movers will be the ones who stand to capture the oversized profits.

DAVID KEMMERER

Op Ed: Anatomy of the Tether Attack: Are Stablecoins Vulnerable?

Last month's attack on Tether contains a cautionary tale: Only those coins that can survive such attacks have the slightest chance of becoming the “holy grail" of stablecoins.

HENRY HE