From: bitcoin-dev-request@lists.linuxfoundation.org
Subject: bitcoin-dev Digest, Vol 99, Issue 45
Date: 23 Aug 2023 at 20:18:38
To: bitcoin-dev@lists.linuxfoundation.org

Send bitcoin-dev mailing list submissions to
bitcoin-dev@lists.linuxfoundation.org

To subscribe or unsubscribe via the World Wide Web, visit
https://lists.linuxfoundation.org/mailman/listinfo/bitcoin-dev
or, via email, send a message with subject or body 'help' to
bitcoin-dev-request@lists.linuxfoundation.org

You can reach the person managing the list at
bitcoin-dev-owner@lists.linuxfoundation.org

When replying, please edit your Subject line so it is more specific
than "Re: Contents of bitcoin-dev digest..."

Today's Topics:

1. Re: Combined CTV+APO to minimal TXHASH+CSFS (Brandon Black)
2. Re: Combined CTV+APO to minimal TXHASH+CSFS (Brandon Black)
3. Re: Concern about "Inscriptions" (Erik Aronesty)
4. Concern about "Inscriptions" (martl.chris@proton.me)

Message: 1
Date: Tue, 22 Aug 2023 12:08:36 -0700
From: Brandon Black <freedom@reardencode.com>
To: bitcoin-dev@lists.linuxfoundation.org
Subject: Re: [bitcoin-dev] Combined CTV+APO to minimal TXHASH+CSFS

Message-ID: <ZOUHtBvuyRPNaUJS@console>
Content-Type: text/plain; charset=us-ascii

Quick update to the proposal thanks to James O'Beirne: CTV is 2-bytes less expensive than I thought when used alone. I thought that script success required exactly OP_TRUE not just a CastToBool()=true value on the stack.

This means that my proposal is 2 weight units (0.5vBytes) larger than CTV when both are used in Tapscript.

--Brandon

Message: 2
Date: Tue, 22 Aug 2023 13:23:09 -0700
From: Brandon Black <freedom@reardencode.com>
To: Greg Sanders <gsanders87@gmail.com>
Cc: Bitcoin Protocol Discussion
  <bitcoin-dev@lists.linuxfoundation.org>
Subject: Re: [bitcoin-dev] Combined CTV+APO to minimal TXHASH+CSFS
Message-ID: <ZOUZLdQUdJetkPaR@console>
Content-Type: text/plain; charset=us-ascii

> * If the top item on the stack is not a minimally encoded `OP_0`, `OP_1`, or
> `OP_2`; succeed immediately[^2].
>
> I presume this was supposed to go to OP_4 now.

Fixed, thanks!

> ### How does the efficiency compare to [bip118][]?
>
> Just noting BIP118 also allows pubkey of "1" to stand in for the taproot inner pubkey, which would be a common use-case. "simply" adding an opcode ala OP_INNER_PUBKEY could also have the same effect of course.

Updated the spec for OP_CSFS to replace OP_0 as pubkey with the taproot internal key. That's a great feature to keep!

> Also, BIP118 also opens the door for non-APO signatures to have a sighash digest that commits to additional data, closing a couple of taproot malleability bugs. See https://github.com/bitcoin-inquisition/bitcoin/issues/19 for more discussion along those lines. These aren't make or break, but would be nice to clean up if possible

Agreed. If this proposal moves forward, I will carefully consider the contents of the hash (as shown in the table at the end) for each mode, and add (or remove) committed data. It might be worth having mode 0 (CTVish) commit to the spend_type and annex as well.

Thanks much,

--Brandon

Message: 3
Date: Wed, 23 Aug 2023 13:34:19 -0400
From: Erik Aronesty <erik@q32.com>
To: symphonicbtc <symphonicbtc@proton.me>, Bitcoin Protocol
Discussion <bitcoin-dev@lists.linuxfoundation.org>
Cc: John Tromp <john.tromp@gmail.com>
Subject: Re: [bitcoin-dev] Concern about "Inscriptions"
Message-ID:

<CAJowKg+5NCrnyb9P1uhKvT75dpA=n8hWU4R_DxcUPuVpBvhCEg@mail.gmail.com>
Content-Type: text/plain; charset="utf-8"

indeed, i once added a proof-of work requirement to public keys on an open relay server protocol, in additon to posk, in order to make it harder to roll new keys and access the network as a spammer/scammer. not hard to embed anything in a public key, but you can't embed too much, especially if you want mobile devices to be able to generate a new key in under a few minutes.

On Mon, Aug 21, 2023 at 6:46?PM symphonicbtc via bitcoin-dev <bitcoin-dev@lists.linuxfoundation.org> wrote:

> It is important to also note that proof of secret key schemes are highly data inefficient and likely would have a higher cost for users than simply allowing arbitrary data to continue. In ECDSA, purposely re-using k values allows you to encode data in both k and the entire secret key, as both become computable. Or, one could bruteforce a k value to encode data in a sig, as is done in some compromised hardware key exfiltration attacks. Additionally, one can bruteforce keys in order to encode data in the public key.

It is very difficult and expensive to attempt to limit the storage of arbitrary data in a system where the security comes from secret keys being arbitrary data.

Symphonic

------- Original Message -------
On Monday, August 21st, 2023 at 4:28 PM, John Tromp via bitcoin-dev <bitcoin-dev@lists.linuxfoundation.org> wrote:

> If we ban "arbitrary data", however you want to define it, then actors will
> simply respond by encoding their data within sets of public keys. Public
> key data is indistinguishable from random data, and, unless we are willing
> to pad the blockchain with proof of knowledge of secret keys, there will be
> no way to tell a priori whether a given public key is really a public key
> or whether it is encoding an inscription or some other data.

Note that in the Mimblewimble protocol, range proofs already prove knowledge of blinding factor in Pedersen commitments, and thus no additional padding is needed there to prevent the encoding of spam into cryptographic material. This makes pure MW blockchains the most inscription/spam resistant [1].

> [1]
> https://bitcointalk.org/index.php?
> topic=5437464.msg61980991#msg61980991
>
> > _____
> > bitcoin-dev mailing list
> > bitcoin-dev@lists.linuxfoundation.org
> > https://lists.linuxfoundation.org/mailman/listinfo/bitcoin-dev
>
> _____
> bitcoin-dev mailing list
> bitcoin-dev@lists.linuxfoundation.org
> https://lists.linuxfoundation.org/mailman/listinfo/bitcoin-dev

---------- next part ----------
An HTML attachment was scrubbed...
URL: <http://lists.linuxfoundation.org/pipermail/bitcoin-dev/attachments/20230823/5c0013ce/attachment-0001.html>

Message: 4
Date: Tue, 22 Aug 2023 05:15:12 +0000
From: martl.chris@proton.me
To: "bitcoin-dev@lists.linuxfoundation.org"
  <bitcoin-dev@lists.linuxfoundation.org>
Subject: [bitcoin-dev] Concern about "Inscriptions"
Message-ID:
  <eKbLe6Hv-k3VjciB8gnpX1TvM2Tv1MjlW87n_iZUrY7DY_y6H4g-yZcmKGC8esRWQHsAbABI7o7PTHIWY45Y1csT326_ZCiW3oDphfq-jkU=@proton.me>

Content-Type: text/plain; charset="utf-8"

Good Morning List,

understanding the strategy wrongly or purposelydriving the interaction into a false framing doesn?t benefit Bitcoin. Mentioning <ban"arbitrary data"> or <governments try tocensor> distracts from the proposed strategy.

The strategy aims to increment the coercion cost of mining-entities relative to the cooperation cost of mining-entities in regards of arbitrary data insertion; it is not about banning or censoring.

Mentioning other ways or methods to insert arbitrary data should not be understood as a threat or menace, but much more as topics which each Bitcoin developer should have in mind to solve.

It is healthy to assume that arbitrary data insertions are nothing but innocent or negligible. The Bitcoin system is still in the struggle of two visions:

- One controlled by every node operator and highly decentralized. (Nobody alone controls)

or

- One controlled by a few very highly capitalized entity node operators and highly centralized. (A committee controls).

The best tactical way to reach the latter is via prohibitive cost increment for operating a regular Bitcoin node (aka. archival full node). That will reduce the network decentralization making it susceptible for central entity elimination or control acquisition; and not necessary by a national-state government.

Chris

------- Forwarded Message -------
Von: Russell O'Connor <roconnor@blockstream.com>
Datum: Am Montag, 21. August 2023 um 16:47
Betreff: Re: [bitcoin-dev] Concern about "Inscriptions"
An: martl.chris@proton.me <martl.chris@proton.me>, Bitcoin Protocol Discussion <bitcoin-dev@lists.linuxfoundation.org>

> It's been said before, but I'll say it again:
>
> If we ban "arbitrary data", however you want to define it, then actors will simply respond by encoding their data within sets of public keys. Public key data is indistinguishable from random data, and, unless we are willing to pad the blockchain with proof of knowledge of secret keys, there will be no way to tell a priori whether a given public key is really a public key or whether it is encoding an inscription or some other data.
>
> When certain governments try to censor certain internet protocols, users respond by tunnelling their protocol through something that appears to be innocent HTTPS (see Tor bridge nodes). This works because, after a handshake, the remaining HTTPS stream, like public keys, is indistinguishable from random data, and can be used as a communications channel for arbitrary data. If we attempt to ban "arbitrary data", those users will simply respond by "tunneling" their data over innocent-looking public key data instead.
>
> Please correct me if I'm wrong, but I believe Counterparty has, in the past, encoded their data within public key data, so this concern is not hypothetical.
>
> > On Sat, Aug 19, 2023 at 10:29?AM Chris Martl via bitcoin-dev <bitcoin-dev@lists.linuxfoundation.org> wrote:

It is already more than a half year since the probably mayor Bitcoin script exploit started.

These exploits are nothing new in the Bitcoin history and mostly are due to the loose flexibility of the system in regards of processing predicatives (Bitcoin script). The very first mayor bug; if you wish, vulnerability, was the CVE-2010-5141, which still engages us without end even after 14 years.

Subsequent Bitcoin historical events let to build more ?improvements? upon this wobbly basis exposing even more ground for exploits.

As long as this loose flexibility is not modified in a way its exposure for exploits is eliminated remains nothing else than to pursue other strategies; and ones which are compatible with the current status quo and furthermore, with a permission-less system.

Here a strategy proposal:

Let?s name it: #Ordisrespector and #Ordislow.

Why #Ordisrespector and #Ordislow are compatible with a permission-less system.

#Ordisrespector gives the option to a regular Bitcoin node operator to opt-in or not to a self-defense of his/her storage property (and thus of his/her integrity); by giving a signal of dissatisfaction with the current affairs of aggression via insertion of arbitrary data into the witness structure. This dissatisfaction signal is manifested by not taking into the mempool and relaying transactions with inserted arbitrary data in the witness structure.

> #Ordislow gives the option to a regular Bitcoin node operator to opt-in or not to a self-defense of his/her storage property (and thus of his/her integrity); by increasing the coercion cost of mining-entities relative to the cooperation cost of mining-entities due to the current affairs of aggression via insertion of arbitrary data into the witness structure. This coercion cost increment is manifested by not propagating a found block, unless a configurable or maximum delay has elapsed, which contains at least a transaction with inserted arbitrary data in the witness structure.
>
> Chris_____
> bitcoin-dev mailing list
> bitcoin-dev@lists.linuxfoundation.org
> https://lists.linuxfoundation.org/mailman/listinfo/bitcoin-dev

---------- next part ----------
An HTML attachment was scrubbed...
URL: <http://lists.linuxfoundation.org/pipermail/bitcoin-dev/attachments/20230822/9c0898f4/attachment.html>

Subject: Digest Footer

---

bitcoin-dev mailing list
bitcoin-dev@lists.linuxfoundation.org
https://lists.linuxfoundation.org/mailman/listinfo/bitcoin-dev

---

End of bitcoin-dev Digest, Vol 99, Issue 45

****