# [bitcoin-dev] BIP proposal - Dandelion: Privacy Preserving Transaction Propagation

**Gregory Maxwell** greg at xiph.org
*Tue Jun 13 01:00:50 UTC 2017*


- Previous message: [bitcoin-dev] BIP proposal - Dandelion: Privacy Preserving Transaction Propagation
- Next message: [bitcoin-dev] Proposal: Demonstration of Phase in Full Network Upgrade Activated by Miners
- **Messages sorted by:** [ date ] [ thread ] [ subject ] [ author ]


---

On Mon, Jun 12, 2017 at 2:46 PM, Andrew Miller via bitcoin-dev
<bitcoin-dev at lists.linuxfoundation.org> wrote:
> *Dear bitcoin-dev,*
> *    We've put together a preliminary implementation and BIP for*
> *Dandelion, and would love to get your feedback on it. Dandelion is a*
> *privacy-enhancing modification to Bitcoin's transaction propagation*
> *mechanism. Its goal is to obscure the original source IP of each*
> *transaction.*

I'm glad to see this out now, so I'm not longer invading the git repo
uninvited. :)

> *- Stronger attacker model: we defend against an attacker that*
> *actively tries to learn which nodes were involved in the stem phase.*
> *Our approach is called "Mempool Embargo", meaning a node that receives*
> *a "stem phase" transaction behaves as though it never heard of it,*
> *until it receives it again from someone else (or until a random timer*
> *elapsess).*

The description in the BIP appears inadequate:

> *That is, Alice will not include the embargoed transaction when responding to MEMPOOL requests, and will not respond to GETDATA requests from another node (Bob) unless Alice previously sent an INV to Bob. The embargo period ends as soon as Alice receives an INV advertising the transaction as being in fluff mode.*

For example, it's not clear if I can query for the existence of a
transaction by sending a conflict.  If this doesn't seem problematic,
consider the case where I, communicating with you over some private
channel, send you a payment inside a payment protocol message. You
announce it to the network and I concurrently send a double spend.
Only nodes that were part of the stem will reject my double spend, so
I just learned a lot about your network location.

It's also appears clear that I can query by sending an inv and

noticing that no getdata arrives. An example attack in the latter is that when I get a stem transaction I rapidly INV interrogate the entire network and by observing who does and doesn't getdata I will likely learn the entire stem path upto permutation.

The extra network capacity used by getdata-ing a transaction you already saw via dandelion would be pretty insignificant.

I believe the text should be simplified and clarified so just say:

"With the exception of her behavior towards the peer sending in the stem transaction and the peer sending out the transaction Alice's behavior should be indistinguishable from a node which has not seen the transaction at all until she receives it via ordinary forwarding or until after the timeout." -- then its up to the implementation to achieve indistinguishably regardless of what other protocol features it uses.

*> Are there other ways we haven't thought of? We think the alternative*
*> approach (bypassing mempool entirely) seems even harder to get right,*
*> and foregoes existing DoS protection.*

I think avoiding the is the most sensible way; and from a software maintenance perspective I expect that anything less will end up continually suffering from serious information leaks which are hard to avoid accidentally introducing via other changes.

The primary functionality should be straightforward to implement, needing just a flag to determine if a transaction would be accepted to the mempool but for the flag, but which skips actually adding it.

Handling chains of unconfirmed stem transactions is made more complicated by this and this deserves careful consideration. I'm not sure if its possible to forward stem children of stem transactions except via the same stem path as the parent without leaking information, it seems unlikely.

This approach would mostly take additional complexity from the need to limit the amplification of double spends. I believe this can be resolved by maintaining a per-peer map of the not yet expired vin's consumed by stem fowards sent out via that peer. E.g. vin->{timeout, feerate}. Then any new forward via that stem-peer is tested against that map and suppressed if it it spends a non-timed-out input and doesn't meet the feerate epsilon for replacement.

Use of the orphan map is not indistinguishable as it is used for block propagation, and itself also a maintenance burden to make sure unrelated code is not inadvertently leaking the stem transactions.

*> After a random number of hops along the stem, the transaction enters the fluff phase,*

The BIP is a bit under-specified on this transition, I think-- but I know how it works from reading the prior implementation (I have not yet read the new implementation).

The way it works (assuming I'm not confused and it hasn't changed) is that when a new stem transaction comes in there is a chance that it is treated as coming in as a normal transaction.

An alternative construction would be that when a stem transaction goes

```
out there is a random chance that the stem flag is not set (with
suitable adjustment to keep the same expected path length)

For some reason I believe this would be a superior construction, but I
am only able to articulate one clear benefit:  It allows non-dandelion
capable nodes to take on the role of the last stem hop, which I
believe would improve the anonymity set during the transition phase.

Unrelated:

Has any work been given to the fact that dandelion propagation
potentially making to measure properties of the inter-node connection
graph?  e.g.  Say I wish to partition node X by disconnecting all of
its outbound connections, to do that it would be useful to learn whom
is connected to X.  I forward a transaction to X, observe the first
node to fluff it,  then DOS attack that node to take it offline.  Will
I need to DOS attack fewer or more nodes  to get all of X's outbounds
if X supports rapid stem forwarding?
```

---


- Previous message: [bitcoin-dev] BIP proposal - Dandelion: Privacy Preserving Transaction Propagation
- Next message: [bitcoin-dev] Proposal: Demonstration of Phase in Full Network Upgrade Activated by Miners
- **Messages sorted by:** [ date ] [ thread ] [ subject ] [ author ]


---

More information about the bitcoin-dev mailing list