比特币可扩展性

--大众的加密货币

讲座大纲

比特币的可扩展性问题

方块辩论

建议的可扩展性解决方案

比特币的可扩展性问题

比特币可扩展性问题

比特币可伸缩性问题是指关于比特币网络可处理交易量限制的讨论。这与比特币区块链中的记录 (称为块) 的大小和频率有限有关。

- 比特币的块包含比特币网络上的交易。
- 比特币网络的链上交易处理能力受平均块创建时间10分钟和块大小限制的限制。

这些共同限制了网络的吞吐量。事务处理能力最大值估计在每秒3.3 到7个事务之间。

有各种建议和激活的解决方案来解决这一问题。

可伸缩性

可伸缩性：在此上下文中，系统处理交易量增加的能力。

我们通常使用的单位**Tps**或***每秒交易记录***，以描述比特币等金融系统的可扩展性。

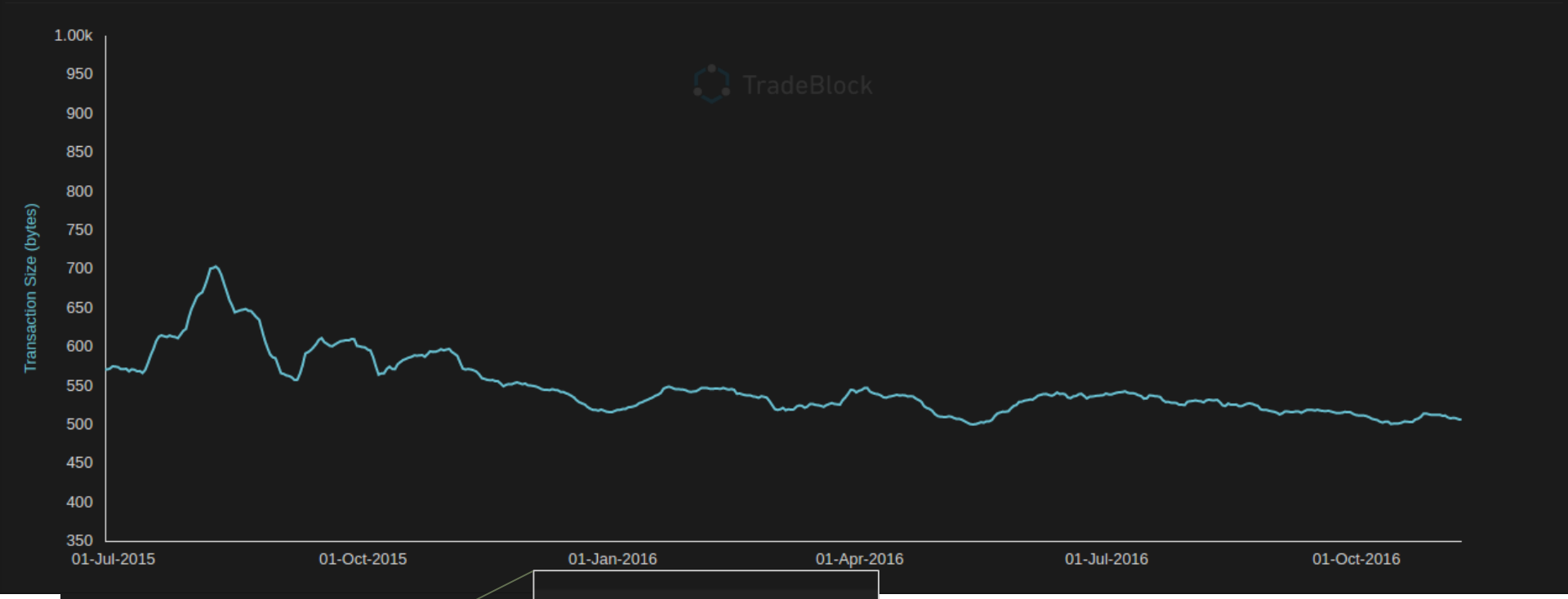
在考虑可伸缩性时，还应该考虑大小和速度。

比特币区块链目前已经是 80 gb。

我们能不能让它能让节点有效地将其存储到未来？

Historical Data

https://tradeblock.com/bitc... 1H 2H 6H 1D 1W



Statistics

	Transaction Size
Observations	500
Mean	546.38
Median	530.94
Mode	391.77
Std. Dev.	69.58

500

546.38

530.94

https://tradeblock.com/bitcoin/historical/1d-f-tsize_per_avg-00271

比特币的当前可扩展性

从上一张幻灯片:

- 平均546字节每交易记录。
- 当前块大小为1 mib.
- 下一个块的预期时间是10分钟.

因此, 我们可以计算以 tps 为单位的持续最大交易量:

$$\frac{1\ MiB}{1\ block} \times \frac{1\ txn}{546\ bytes} \times \frac{1\ block}{10\ min} \approx 3.2\ tps$$

可扩展性比较

比特币与其他传统支付系统相比如何?

	平均	高负载/最大值
比特币	2.2 tps	3.2 tps
贝宝 *	150 tps	450 tps
签证 * *	2, 000点	56, 000tps

*http://www.fool.com/investing/general/2016/02/04/5-things-paypal-holdings-inc-wants-you-to-know.aspx
**https://usa.visa.com/dam/VCOM/download/corporate/media/visa-fact-sheet-Jun2015.pdf

比特币的当前可扩展性

从上一张幻灯片:

- 平均546字节每交易记录。
- 当前块大小为1 mib.
- 下一个块的预期时间是10分钟.

因此, 我们可以计算以 tps 为单位的持续最大交易量:

$$\frac{1\ MiB}{1\ block} \times \frac{1\ txn}{546\ bytes} \times \frac{1\ block}{10\ min} \approx 3.2\ tps$$

我们可以玩的变量是什么?

- **块的大小**
- **交易记录的大小**
- **块创建速率**
- **是否添加交易记录**

天真的解决方案

只需提高块的速度。 最简单的解决方案。 只要减少 nonce 就可以了。

缺点:

- 传播时间更短
- 更多方块规划
  - 更多的人同时找到解决方案
- 搞砸了减半
- 更频繁的叉子
  - 因此, 双倍支出的潜力
- 比增加块大小更不可能被接受
  - 被视为协议的一个基本部分, 而不仅仅是保护性的软限制

更少的时间进行确认, 但不像真的, 因为你现在需要更多的确认

天真的解决方案

这里是维塔利克·布特林, 以太的创造者,提出正是这样的, 并被完全忽视。

Soft-forking the block time to 2 min: my primarily silly and academic (but seemingly effective) entry to the "increase the blockchain's capacity in an arbitrarily roundabout way as long as it's a softfork" competition (self.btc)

submitted 9 months ago * (last edited 9 months ago) by vbuterin

So given that large portions of the bitcoin community seem to be strongly attached to this notion that hard forks are an unforgivable evil, to the point that schemes containing hundreds of lines of code are deemed to be a preferred alternative, I thought that I'd offer an alternative strategy to increasing the bitcoin blockchain's throughput with nothing more than a soft fork - one which is somewhat involved and counterintuitive, but for which the code changes are actually quite a bit smaller than some of the alternatives; particularly, "upper layers" of the protocol stack should need no changes at all.

Notes:

- Unlike the "generalized softfork" approach of putting the "real" merkle root in the coinbase of otherwise mandatorily empty blocks, this strategy makes very little change to the semantics of the protocol. No changes to block explorers or wallets required.
- The point of this is largely academic, to show what is possible in a blockchain protocol. That said, if some segwit-as-block-size-increase supporters are interested in segwit because it increases the cap in a way that does not introduce a slippery slope, block time decreases are a viable alternative strategy, as there is a limit to how low block time can go while preserving safety and so the slippery slope has a hard stop and does not extend infinitely.
- My personal *actual* preference would be a simple s/1000000/2000000/g (plus a cap of 100-1000kb/tx to address ddos issues), though I also believe that people on all sides here are far too quick to believe that the other side is evil and not see that there are plenty of reasonable arguments in every camp. I recommend this, this and this as required reading.
- There's some chance that some obscure rule of the bitcoin protocol makes this all invalid, but then I don't know about it and did not see it in the code.

The attack vector is as follows. Instead of trying to increase the size of an individual block directly, we will create a softfork where **under the softfork rules, miners are compelled to insert incorrect timestamps, so as to trick the bitcoin blockchain into retargeting difficulty in such a way that on average, a block comes every two minutes instead of once every ten minutes**, thereby increasing throughput to be equivalent to a 5 MB block size.

First, let us go over the bitcoin block timestamp and difficulty retargeting rules:

- Every block must include a timestamp.
- This timestamp must at the least be greater than the median of the previous eleven blocks (code here and here)
- For a node to accept a block, this timestamp must be at most 2 hours ahead of the node's "network-adjusted time" (code here), which can itself be at most 70 minutes ahead of the node's timestamp (code here); hence, we can never go more than 3.17 hours into the future
- Every 2016 blocks, there is a difficulty retargeting event. At that point, we calculate D = the difference between the latest block time and the block time of the block 2016 blocks before. Then, we "clamp" D to be between 302400 and 4834800 seconds (1209600 seconds = 2 weeks is the value that D "should be" if difficulty is correctly calibrated). We finally adjust difficulty by a factor of 1/D: for example, if D = 604800, difficulty goes up by 2x, if D = 1814400, difficulty goes down by 33%, etc. (code here)

The last rule ensures that difficulty adjustments are "clamped" between a 4x increase and a 4x decrease no matter what.

So, how to we do this? Let's suppose for the sake of simplicity that in all examples the soft fork starts at unix time 1500000000. We could say that instead of putting the real time into blocks, miners should put `1500000000 + (t - 1500000000) * 5`; this would make the blockchain think that blocks are coming 5x as rarely, and so it would decrease difficulty by a factor of 5, so that from the point of view of actual time blocks will start coming in every two minutes instead of ten. However, this approach has one problem: it is not a soft fork. Users running the original bitcoin client will very quickly start rejecting the new blocks because the timestamps are too far into the future.

天真的解决方案

- 但等等, 以太有 1 7秒的拦网次数。
- 更少的时间进行确认, 但不像真的, 因为你现在需要更多的确认

- 优点和缺点
  - https://blog.ethereum.org/2015/09/14/on-slow-and-fast-block-times/
  - 或者, 阅读：关于慢速和快块时间的研究

建议的可扩展性解决方案

容量增加

隔离证人

西德金

雷电网络

阻止容量增加

增加块大小

> “
> It can be phased in, like:
> if (blocknumber > 115000)
>
> maxblocksize = largerlimit
>
> -- Satoshi Nakamoto

块大小限制已创建了瓶颈在比特币, 导致增加的交易费用和延迟处理的交易, 不能适应一个块。关于如何缩放比特币, 提出了各种建议, 并由此引发了一场有争议的辩论。*业内人士*2017年, 这场争论被描述为 "关于比特币未来的意识形态之争"。

增加块大小

想法：如果我们只是增加块大小，我们可以在一个块中容纳更多的事务。

优点：

- 这是一个 "简单" 的实现。 只要让矿工们同意 (显然我们知道这并不容易)。
- 可能要做闪电网络反正
- 降低交易费用 (如果您是用户)

缺点：

- 硬叉子胡说八道
- 减少交易费用 (如果您是矿工)
- 尺寸增加非常快
  - "滑斜坡"
  - 区块链已经是 80 gb 了!
- 更长的传播时间
  - 创作矿工在下一个街区最好射门
- 一次线性容量增加
  - 临时修复

例子

- 比特币经典
  - 2 mb
- 比特币 xt
  - 8 mb
- 比特币无限
  - 如果一个方块到达链中深处的一定数量的块, 则允许矿工接受一个大于您的最大可接受块大小的块。
  - 从本质上讲, 通过交易费市场, 将由 "供求关系" 决定封锁规模
    - 删除比特币经济中唯一的中央控制点
    - 自由市场经济将迫使人们达成共识
    - 将实时适应现实世界的条件
  - viabtc 的最新情况
    - 他们试图阻止 segwitness 支持比特币无限
    - 他们失去了一半的哈希力量
    - 赛格证人通过

隔离证人

隔离证人

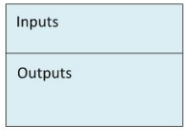
想法：每个事务的数字签名在每个块中占用了大量空间。他们没有理由需要在那里。 让我们删除它们。

如何：

Segwit P2W*

For Old Nodes:

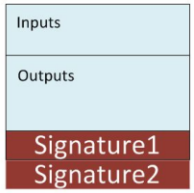
ScriptPubKey: 0 e4873ef43eac347471dd94bc899c51b395a509a5
ScriptSig: Empty
Result: **Valid**



Segwit P2W*

For New Nodes:

ScriptPubKey: 0 e4873ef43eac347471dd94bc899c51b395a509a5
ScriptSig: Empty
WitScript: Signature1
Result: **Valid**



隔离证人

但现在, 封锁链没有任何证据表明交易中包括了正确的签名。

- segwiit 矿工用反映交易树的隔离证人创建了一棵 merkle 树
- 此树的 merkel 根包含在硬币基事务的输入字段中。
- 这将更改硬币基事务的事务 id
- 因此, 签名会影响块标头, 并最终影响区块链的构成。

隔离证人

想法：每个事务的数字签名在每个块中占用了大量空间。他们没有理由需要在那里。 让我们删除它们。

优点:

- 只有柔软的叉子
- 修复事务易变性
  - 允许闪电网络和 sidechains 工作
- 无滑坡
- 效率收益
- 区块链的尺寸更小

缺点:

- 一次线性容量增加
- 介绍新型 dos 攻击 (go-ch即 s-w即 ddos)
- 非常复杂和丑陋 (超过500行代码)
- 解决恶意的其他方法
- 钱包必须把它合并

施诺尔多签名

想法：不要要求每个成员的签名, 而是将它们组合在一起, 只有一个签名

优点:

- 可使用软叉或硬叉 (硬叉清洗剂) 实现
- 多 sig 事务将明显变小
- 更快的验证
- 对参与者的可诉否认

为什么它没有被实施?

当比特币刚出来的时候, ecdsa 是最受欢迎的, 因为 schnorrs 仍然受到专利保护。 已经不是了 各方面都好多了。 只是需要有人来实现它。

https://en.wikipedia.org/wiki/Schnorr_signature

https://en.wikipedia.org/wiki/Elliptic_Curve_Digital_Signature_Algorithm

比特币的当前可扩展性

我们可以玩的变量是什么?

- 块的大小
- 交易记录的大小
- 块创建速率

$$\frac{1\ MiB}{1\ block} \times \frac{1\ txn}{546\ bytes} \times \frac{1\ block}{10\ min} \approx 3.2\ tps$$

现在怎么办?

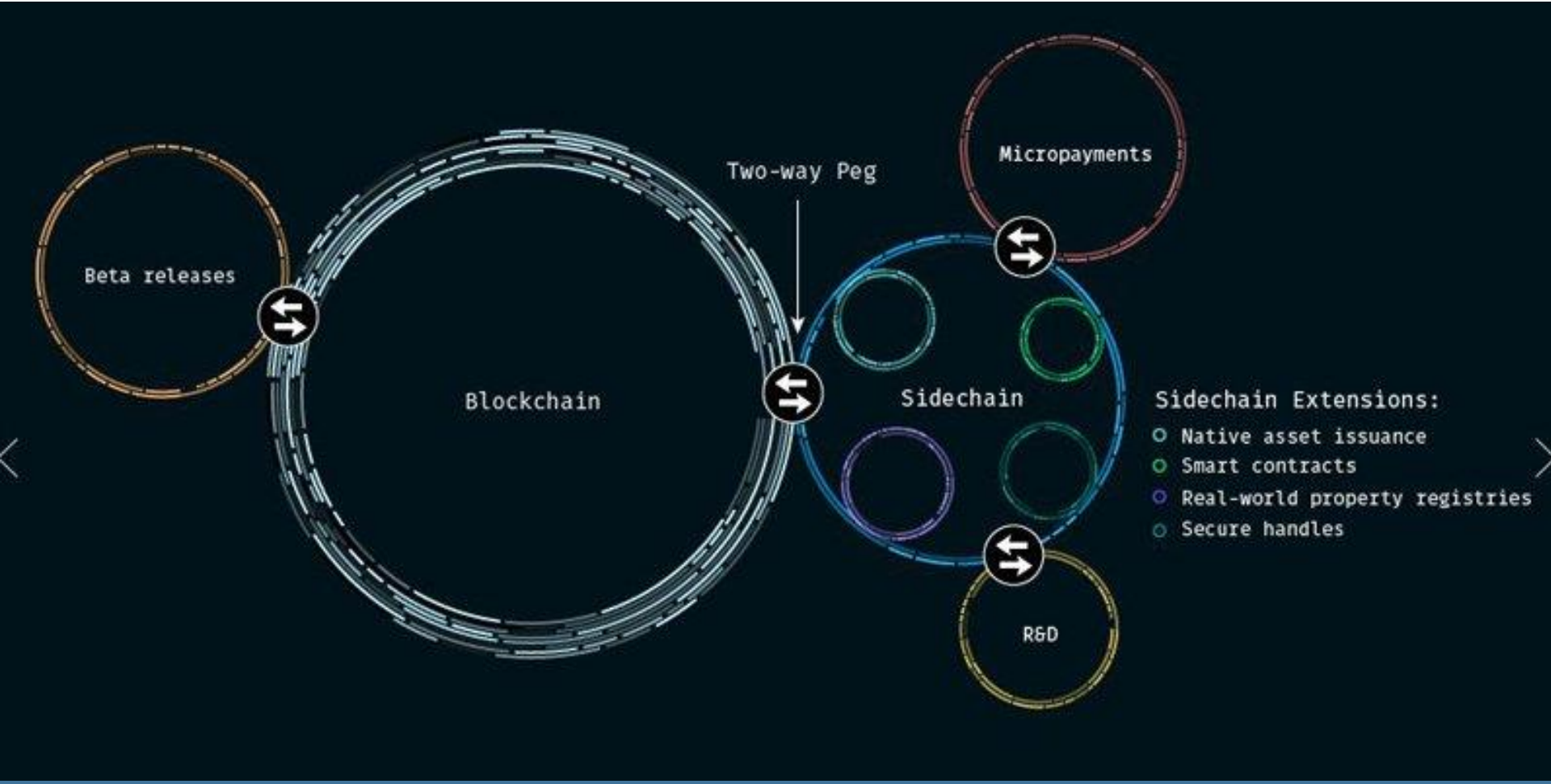
我们需要改变其他的东西。

我们不要用封锁链吧!

西德金

"侧链是一种水平扩展和分片的方式，同时仍保留主链的主要安全参数。未来是一个连锁的互联网，一个以比特币为根链的链森林：事实调查/证明存在，chaindb，区块流侧链，合并开采侧链，以及更多。 通常，多链设计的可扩展性要大得多。

--比特币核心开发者杰夫·加尔齐克


Micropayments
Two-way Peg
Beta releases
Blockchain
Sidechain
Sidechain Extensions:
Native asset issuance
Smart contracts
Real-world property registries
Secure handles
R&D

西德金

想法:

有了 sidechains, 人们可以将他们的比特币转移到一个更快、更不安全的区块链, 用于购买他们的早晨咖啡。当链只用于小型、不重要的事务时, sidechain 具有更大的块大小限制的事实就不会成为问题。

优点:

比特币区块链上的东西较少, 但仍可与之挂钩。

缺点:

不是 "真正" 的解决方案。 只是把东西搬到不同的链子上。

雷电网络

记得:

比特币交易:

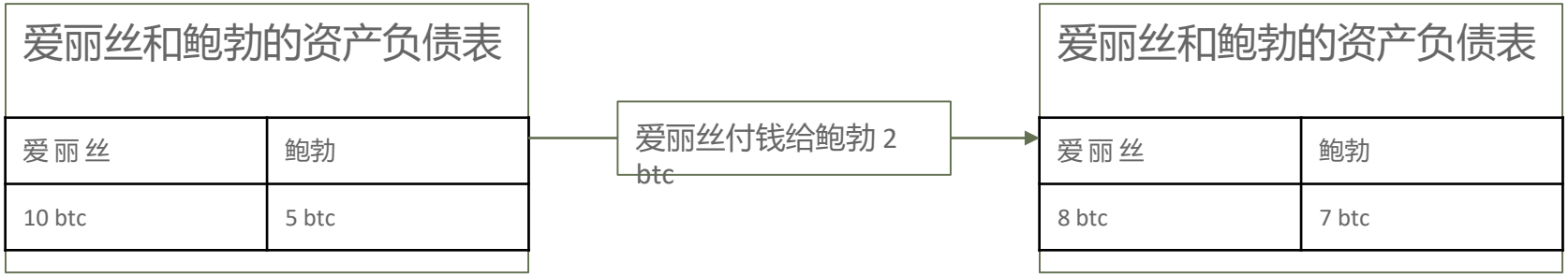
当 alice 想要支付 bob 1 btc 时, alice 签署一项交易, 将其广播到网络, bob 等待一些确认*之前*他认为爱丽丝的付款是有效的。

通过等待块被挖掘在爱丽丝的付款, 鲍勃可以肯定, 爱丽丝不能双倍消费和欺骗鲍勃。

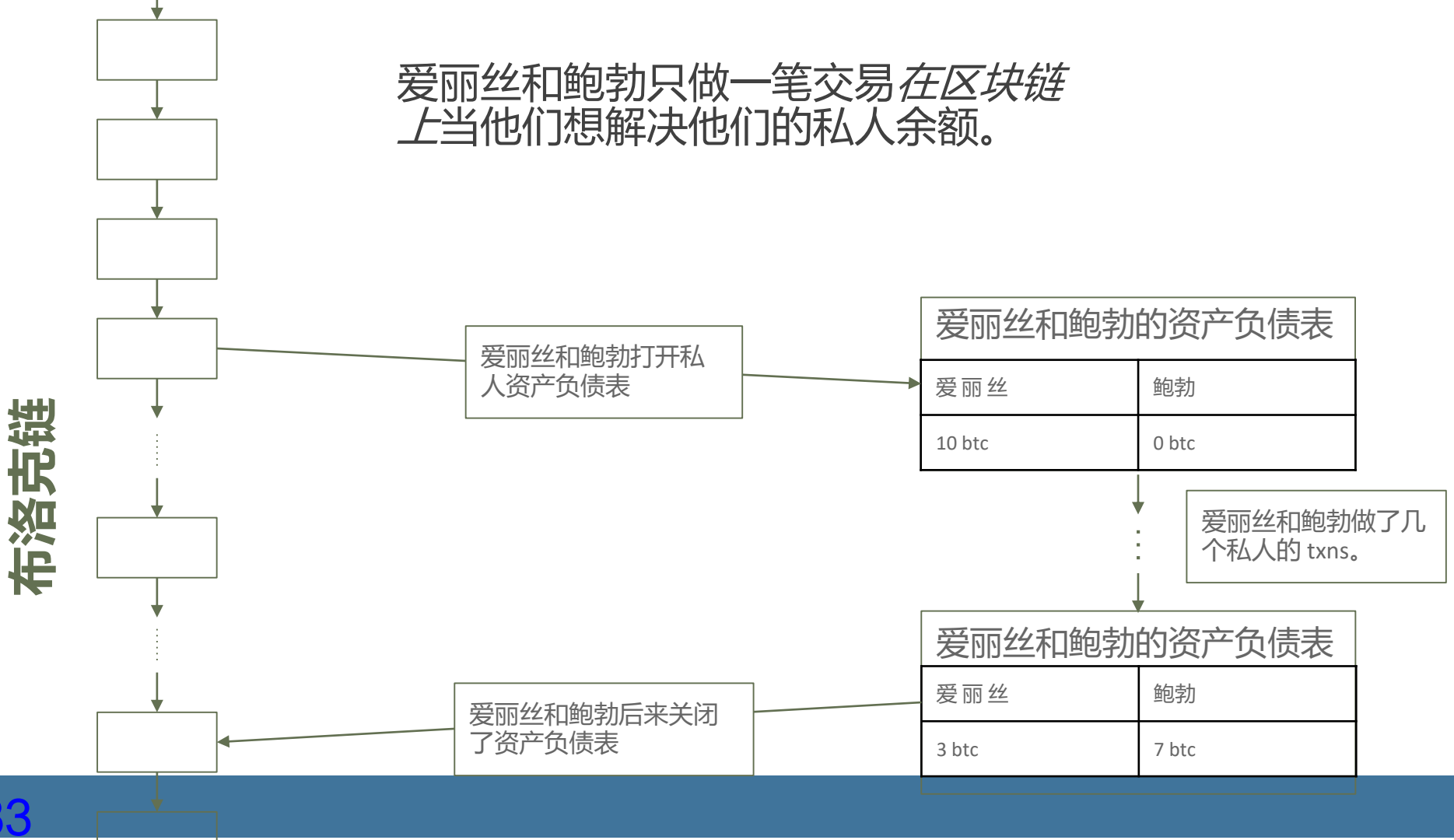
想法:
爱丽丝和鲍勃可以在他们之间进行支付, 而不需要总是咨询区块链, 以防止一个欺骗另一个?

建议：

如果爱丽丝和鲍勃保持一个***私人资产负债表***，更新欠条与每笔付款，只有在一方想要结算时，才咨询区块链？



爱丽丝和鲍勃只做一笔交易*在区块链上*当他们想解决他们的私人余额。



想法：使用比特币脚本创建爱丽丝和 bob 之间的区块链强制执行合同，这样任何一方都不能欺骗对方，同时保持私人资产负债表功能！

在区块链的土地，我们称之为这些**支付渠道**.

这怎么会"*支付渠道*"事情真的管用吗？

哈希锁定时间合同

哈希蒂梅洛克合同是一个非常技术性的加密货币支付的实现。它要求付款的收件人在截止日期之前确认收到上述转账，这是通过生成付款的加密证明或丧失索赔能力，并将其退还给付款人来完成的。到目前为止，这听起来并不异常。然而，整个概念有一个小的转折。

然后，接收方生成的加密付款证明可用于触发其他付款中的其他操作。这使得 htlc 成为比特币中产生条件支付的强大技术。不难理解为什么这项技术会如此强大和受欢迎，事实上，有多个起诉案件，这种技术会派上用场。

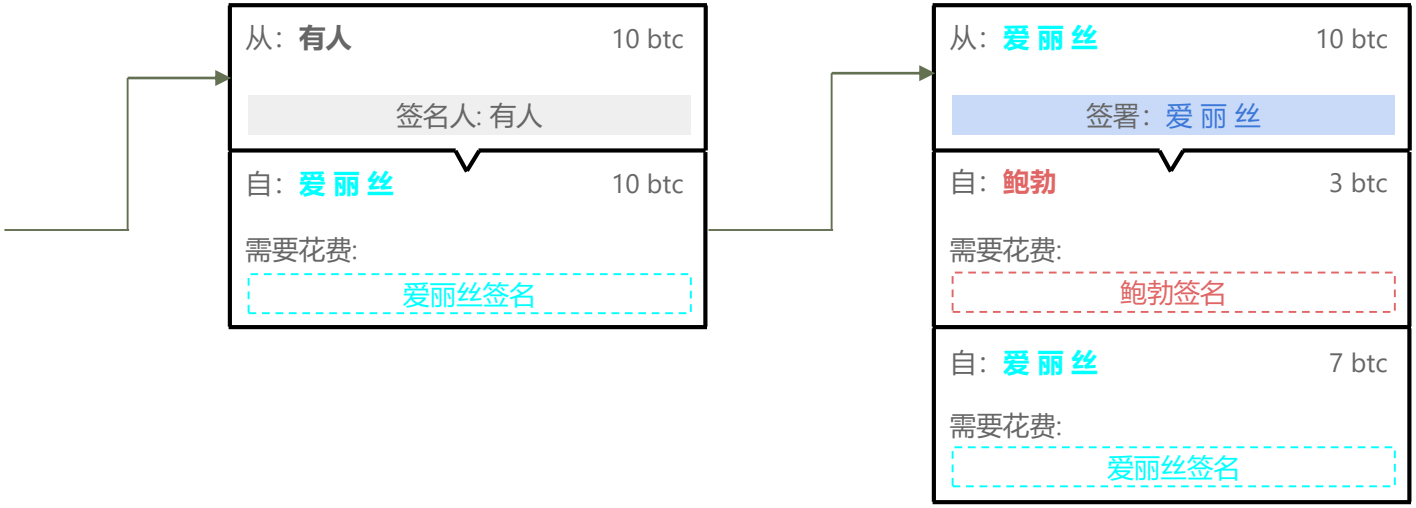
我们现在将建立一个所谓的*哈希时锁定双向支付通道*或*htlc*.

https://captainaltcoin.com/hashed-timelock-contract-htlc/

哈希时锁定双向支付渠道

简单地说, 一些符号:

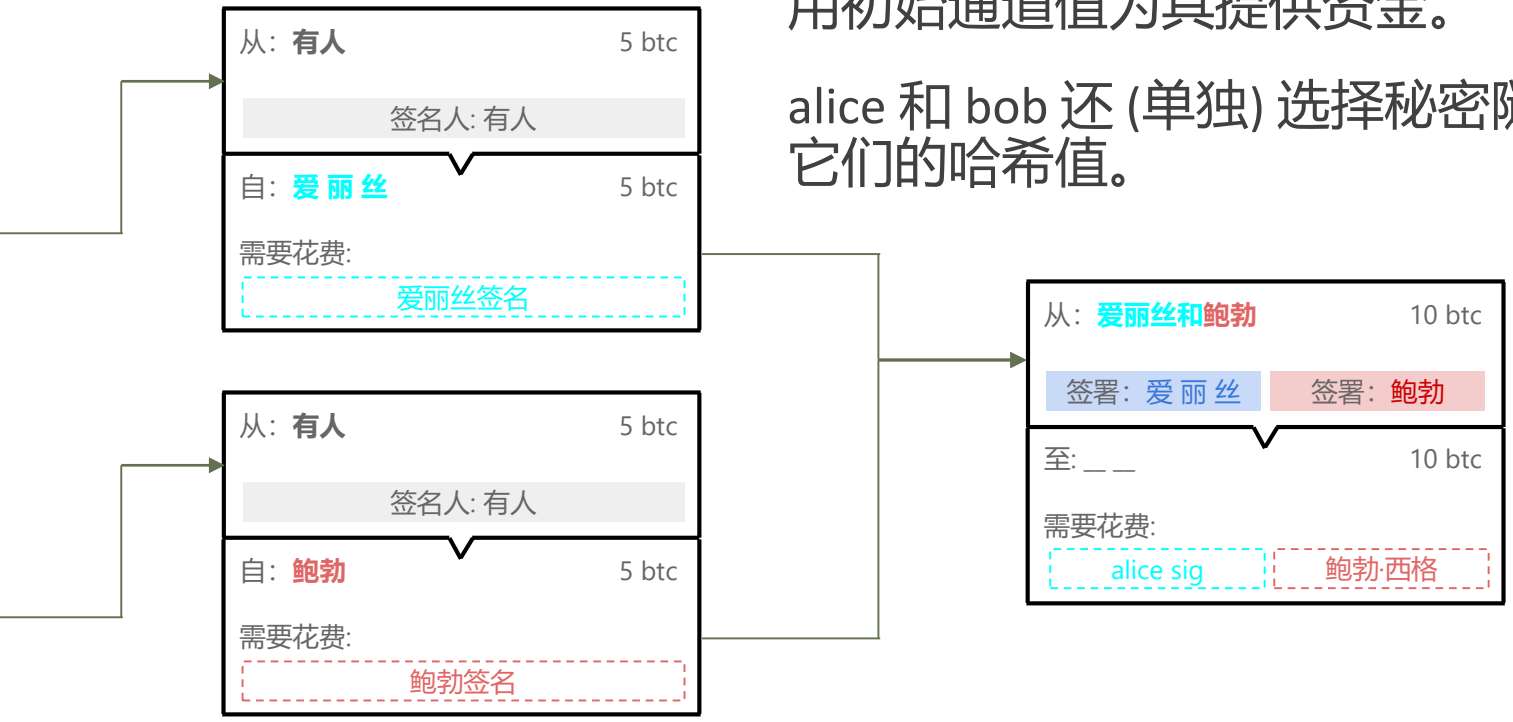
+ alice 花费了 10 btc txn 的输出
+ 爱丽丝发送 3 btc 鲍勃和 7 btc 回到自己。



哈希时锁定双向支付渠道

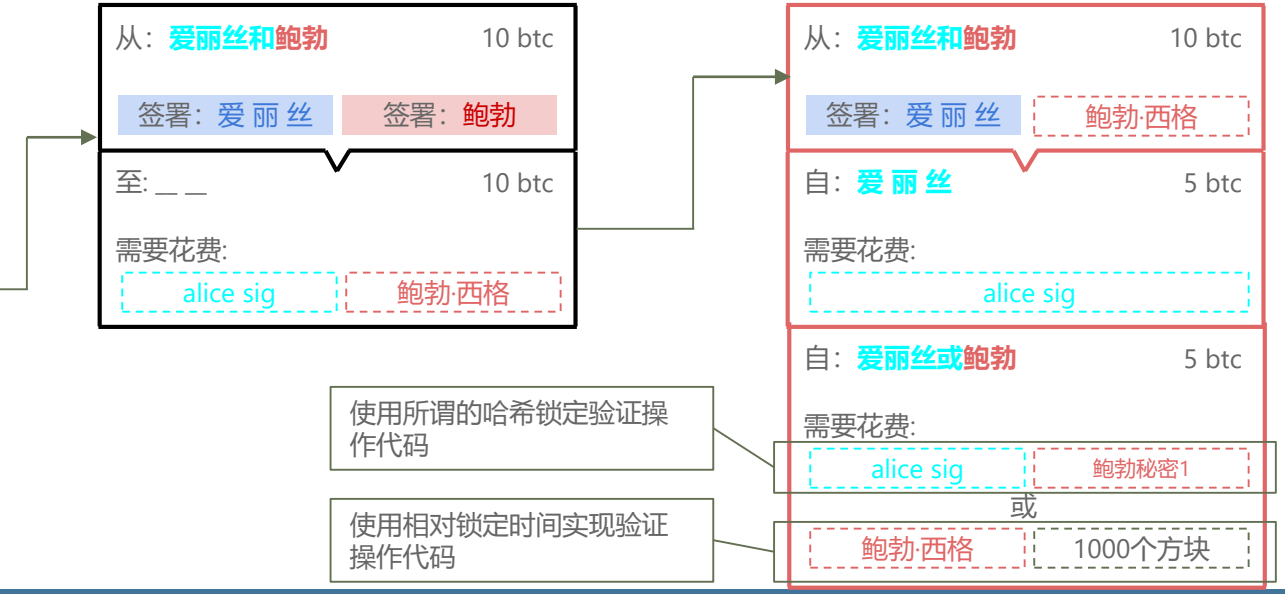
alice 和 bob 创建一个2比2多西格地址, 并使用初始通道值为其提供资金。

alice 和 bob 还 (单独) 选择秘密随机值并交换它们的哈希值。



哈希时锁定双向支付渠道

alice 创建一个承诺交易记录。爱丽丝签署承诺 txn, 并发送 txn 和哈希鲍勃,*而不向网络广播!*

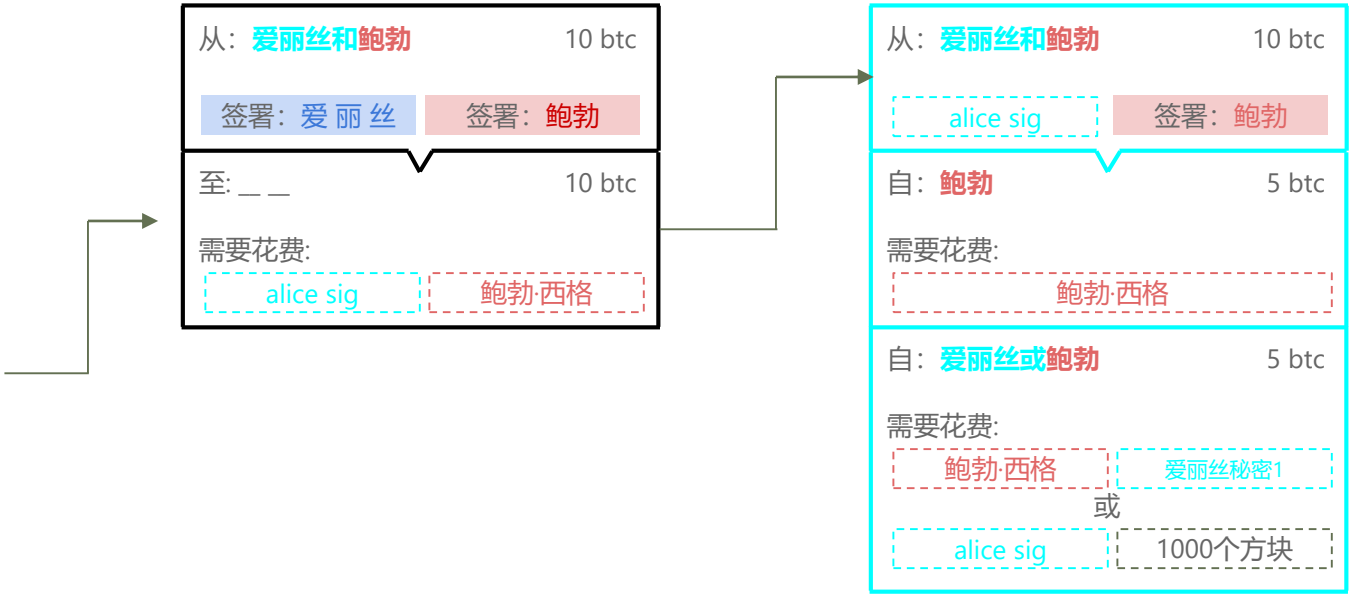


由于 alice 已经签署了她一半的多字输出, bob 可以随时将此事务广播到网络。

但是, 如果 bob 广播, 此事务将自动提供 alice 5 btc, 并且仅返回 bob 的 5 btc*经过1000个方块的等待期*,**或**爱丽丝签署并使用鲍勃的哈希预置图像 (目前只有鲍勃知道)。

哈希时锁定双向支付渠道

鲍勃*也*创建承诺交易记录。bob 签署了 txn 的承诺, 并将 txn 和哈希发送给 alice。

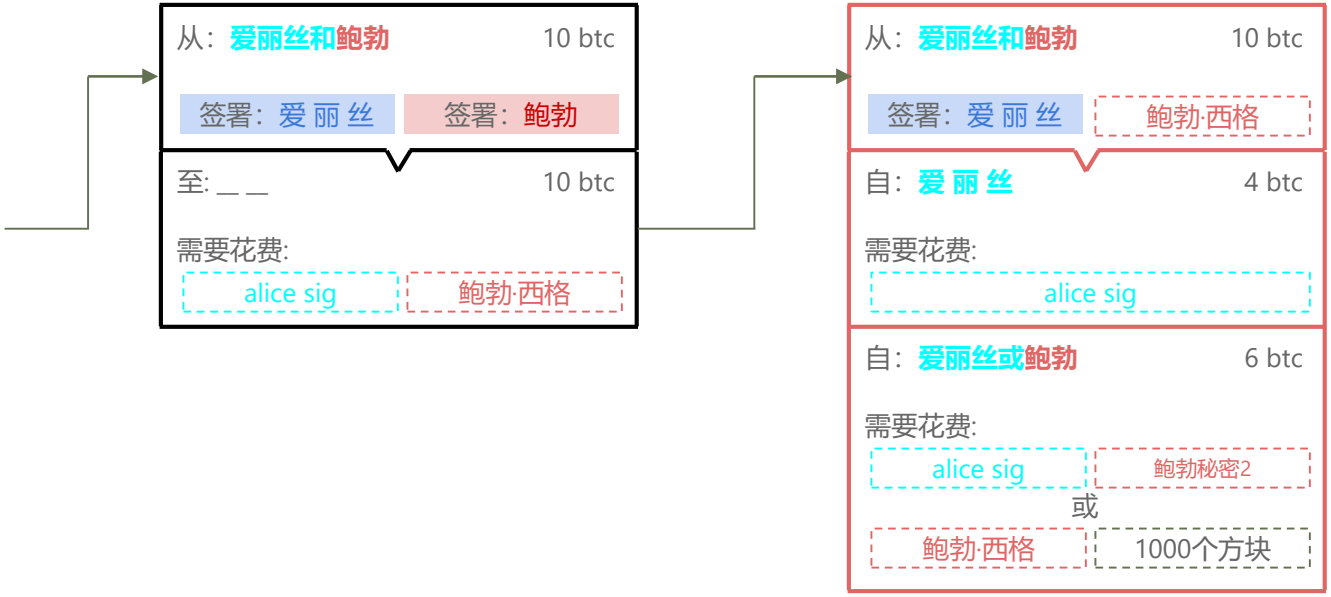


哈希时锁定双向支付渠道

爱丽丝现在想通过支付渠道支付 bob 1 btc 。

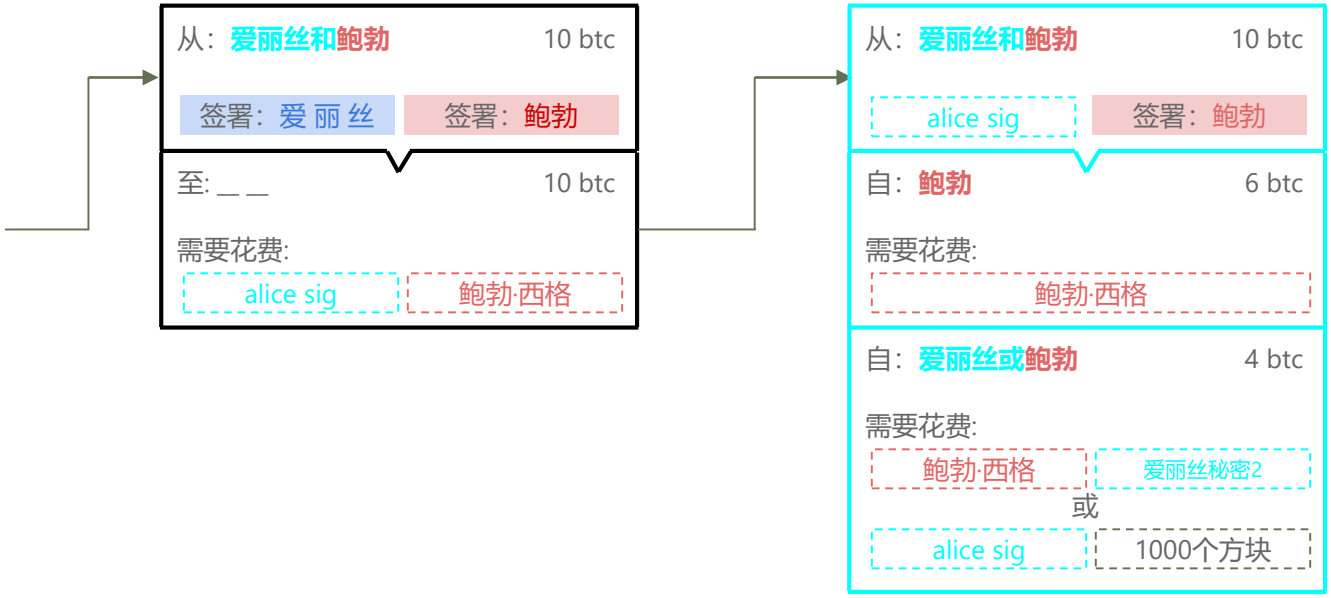
爱丽丝和鲍勃都选择新的秘密值, 并交换他们的哈希。

alice 创建了一个新的承诺 txn, 它只将 4个 btc 发送回自己, 并将 6个 btc 发送到哈希时锁合同中。随后, 爱丽丝在 txn 上签字, 并将其发送给鲍勃。



哈希时锁定双向支付渠道

鲍勃也创建一个新的承诺 txn, 将 6 btc 发送回自己, 并将 4 btc 发送到哈希时锁合约中。鲍勃在 txn 上签字, 然后把它寄给爱丽丝。



关键步骤:alice 和 bob 现在交换前两个秘密值, 以锁定更新后的通道余额!

如果爱丽丝想让她的 1 btc 回来, 欺骗鲍勃呢?

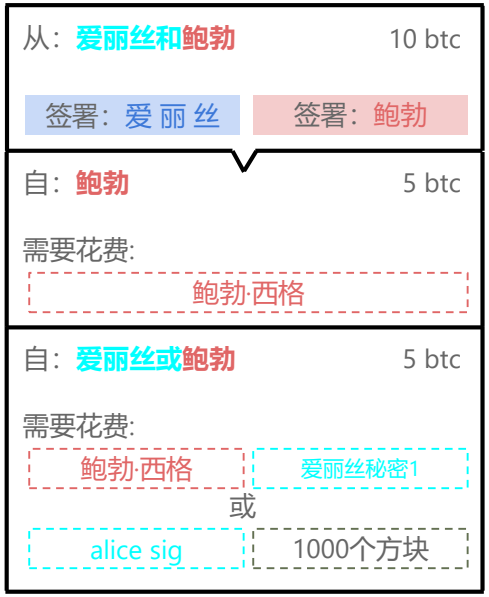
爱丽丝广播旧的承诺 txn 到网络..。

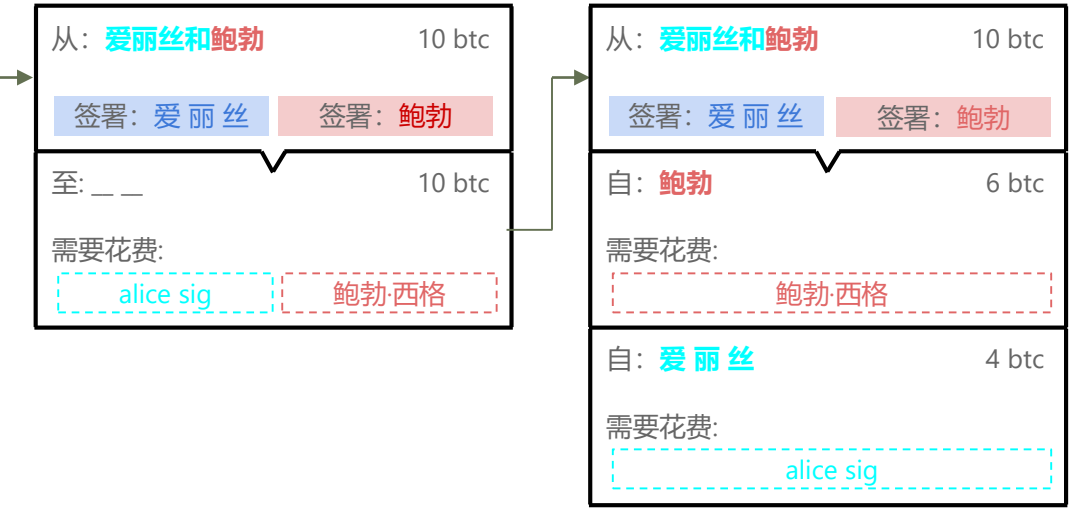
鲍勃自动收到他原来的 5 btc。

爱丽丝现在必须等待1000个块, 以获得她的 5 btc 回来。

然而, 因为爱丽丝和鲍勃只是交换了他们原来的秘密, 鲍勃现在知道爱丽丝秘密1!鲍勃可以看到爱丽丝试图欺骗他, 并采取所有 10 btc!

爱丽丝广播对网络的承诺





沉降：

当 alice 和 bob 想要将余额结算到区块链时，他们只需从原始存款地址进行合作和支出，或者等待足够长的时间，以等待以前承诺到期的锁。

观察：如果爱丽丝和鲍勃总是合作，他们永远不必触及封锁链，除非在创建支付渠道和结算余额的时候。

观察：爱丽丝和鲍勃永远不必互相信任，因为如果一个人试图欺骗，另一个人总是可以覆盖存款中的所有钱。

问题：爱丽丝和鲍勃需要有资本锁定在这个 htlc (哈希锁合同频道) 之前, 他们可以相互汇款。

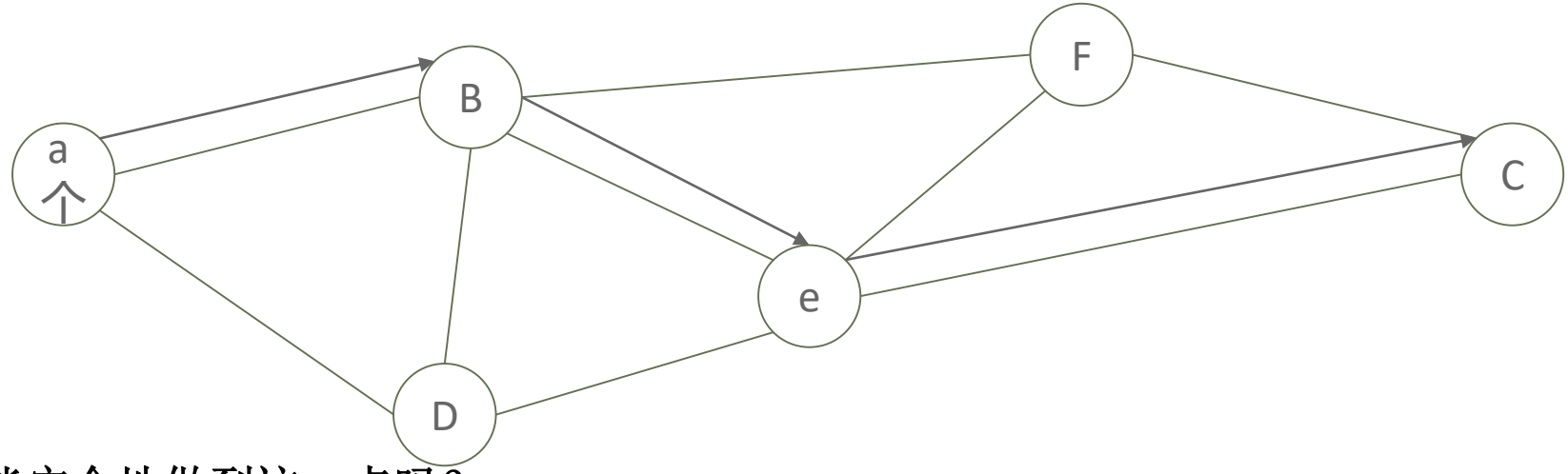
问题：有了这个支付渠道, 爱丽丝和鲍勃只能*容易*并在他们之间大规模地汇款。

如果爱丽丝想在不碰区块链的情况下给查理汇款，但她没有或不希望自己和查理之间建立一个支付渠道？

我不想和一些亚马逊商人建立一个支付渠道，如果我只想和他们交易一两次的话。

想法：创建支付渠道网络。只要爱丽丝和查理有联系，她就能给他寄钱。

爱丽丝通过这个假设的支付渠道网络给查理汇款



我们能安全地做到这一点吗？

事实证明，与我们的 htlc 建设上的一些小补充，我们可以*无信信赖*通过 htlc 网络汇款！

·雷电网络

雷电网络

中。**雷电网络**是在区块链 (最常见的比特币) 之上运行的 "第二层" 支付协议。

- 它支持参与节点之间的快速事务, 并被吹捧为比特币可伸缩性问题的解决方案。
- 它的特点是对等通过双向支付渠道网络进行数字加密货币小额支付的系统, 而不需要下放资金保管权。
- 闪电网络的实现简化了原子交换。

闪电网络的正常使用包括通过向相关区块链提交融资交易来打开支付渠道, 然后进行任意数量的闪电交易, 这些交易更新了该渠道资金的暂定分配, 而不广播到区块链, 然后选择关闭支付渠道, 广播交易的最终版本, 以分配该渠道的资金。

比特币闪电网络: 可扩展的离链即时支付

joseph poon & thaddeus dryja

https://lightning.network/lightning-network-paper.pdf

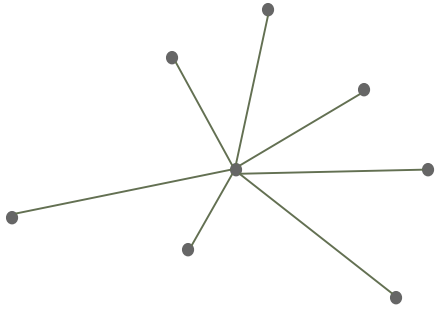
闪电网络对可扩展性意味着什么？

1) 如果我们假设这个支付渠道网络有足够的资金，人们可以支付 "*立即*.
2) 仅使用比特币区块链作为*仲裁*解决纠纷，关闭支付渠道，这意味着区块链上的 (昂贵) 交易少得多。
3) 而不是3Tps，比特币网络可以支持 1000 ' s +Tps通过委托付款到简单的簿记在每个支付渠道，这是保持连锁99% 的时间！
4) 由于闪电网络的交易相对便宜，费用可能也会便宜得多。

雷电网络的问题

1) 节点需要将大量资金锁定在支付渠道中。
2) 强大的集中力，因为只有拥有大量资金的节点才能长期持有支付渠道。
3) 在网络上的节点更少，对中心-分支网络拓扑的趋势所需的资本更少。

集散拓扑



社区和可扩展性

闪电网络 (ln) 建议通过允许节点在将某些事务数据提交到链之前将其保存在缓存中, 从而降低交易成本。

BlockstreamCore is not against hard forks, BlockstreamCore is against scaling Bitcoin.

Scaling Bitcoin would harm their business model - end of the story.

permalink embed save give gold pocket

[–] **dlogemann** 3 points 9 months ago

Please describe this business model of Blockstream. How is their future revenue coming from in your point of view?

permalink embed save parent give gold pocket

There is a lot of money invested in Blockstream Inc, and the only way to get it back is to cripple Bitcoin and then force everyone to use LN on Blockstream owned hubs.

But LN is at least 1-3 years away and as soon as everybody understands that on-chain scaling can be done, they have a hard time forcing everybody to use LN.

There are also huge secondary opportunities for Blockstream. LN needs some central hubs. Whoever wants to run such a hub needs to be very very rich, because LN locks some dedicated funds with each user. Together with other factors like network effect, the Blockstream will eventually have basically no competition (like Facebook today). This will enable them to exclusively! (since everything is off-chain) know and monitor basically every single transaction that happens in the Bitcoin world. Now image how much that data is worth!

总结

- 比特币和其他类似的区块链存在可伸缩性问题:
  a) 如果这些技术希望在全球范围内使用，它们需要支持适当的交易量。
  b) 我们能否在不损害比特币最初关于安全、分散、不可信赖支付的愿景的情况下解决这一问题?
- 建议的解决方案:
  a) 容量增加
  b) 隔离证人
  c) 西德金
  d) 雷电网络

1) 容量增加
   a) 具有较大块的可扩展性较小。
   b) 随着节点的最低服务器要求的增加，集中化风险。
2) 隔离证人 (segwiit)
   a) 小的可伸缩性提升，因为块不需要存储签名。
3) 西德金
   a) 提高可扩展性的潜力
   b) 具有更好可扩展性的潜在新颖的 sidechains (在实践中尚未看到)
4) 雷电网络
   a) 具有提升幅度级可扩展性的巨大潜力。
   b) 基本重组支付流程。
   c) 资本预置带来的集中风险。

धन्यवाद

Hindi 印地语

Спасибо

俄语

شكراً

阿拉伯语

格拉齐

意大利语

நன்றி

Tamil

泰米尔语

以任何努力

多谢

繁体中文

谢谢

英语

多谢

简体中文

ありがとうございました

日语

ขอบคุณ

泰语

谢谢

西班牙语

奥布里加多

西葡萄牙语

丹克

德语

谢谢

法语

감사합니다

朝鲜语