兰花《一个通过带宽挖矿激励的完全分布式的匿名代理网络》

- 作者： David L. Salamon, Brian J. Fox, Jay Freeman, and Gustav Simonsson with Stephen F. Bell, and Dr. Steven Waterhouse, Ph.D.

- Version 0.9.0

- 译者：郭光华

摘要

互联网的飞速发展， 越来越多的方法和途径可以很有效的快速发现和查找个人及组织的隐私， 因此， 大众对匿名化保护隐私的需求越来越迫切。 虽然现在有些方法(如 I2P 和洋葱网络)被广泛采用，但是这些网络只有几千名志愿者愿意加入中继和出口节点， 这样会导致攻击者可以用很少数量的节点轻易的监视整个加密网络。我们提出基于市场的“带宽挖矿“的方法， 来激励大家加入中继和出口节点。

本文旨在描述一个一直在开发的系统。因此会实时的更新该文内容，用以描述在实施当中遇到问题而对系统的调整； 并且该系统可以灵活的使用库组件和特定的加密算法。然而， 不管怎样调整系统，系统的本质会保持目标和目的不变。

我们的设计内容包括：

- 基于区块链(具有顺序打包交易的数据)的随机支付机制
- 带宽销售规格的商品说明书
- 归纳证明一个在对等分布式系统中使得 Eclipse 攻击非常困难的方法
- 在攻击者可能会将其出价作为攻击的一部分的情况下，适用于销售带宽的高效安全硬化拍卖机制
- 完全分布式的匿名带宽市场

1. 介绍

兰花协议将带宽卖家组织成一个刚性结构化的对等（P2P）网络，称为兰花市场。客户连接到兰花市场，并向多个带宽卖家做支付，形成特定网络服务器的代理链。 代理链允许客户从全球互联网发送和接收数据。

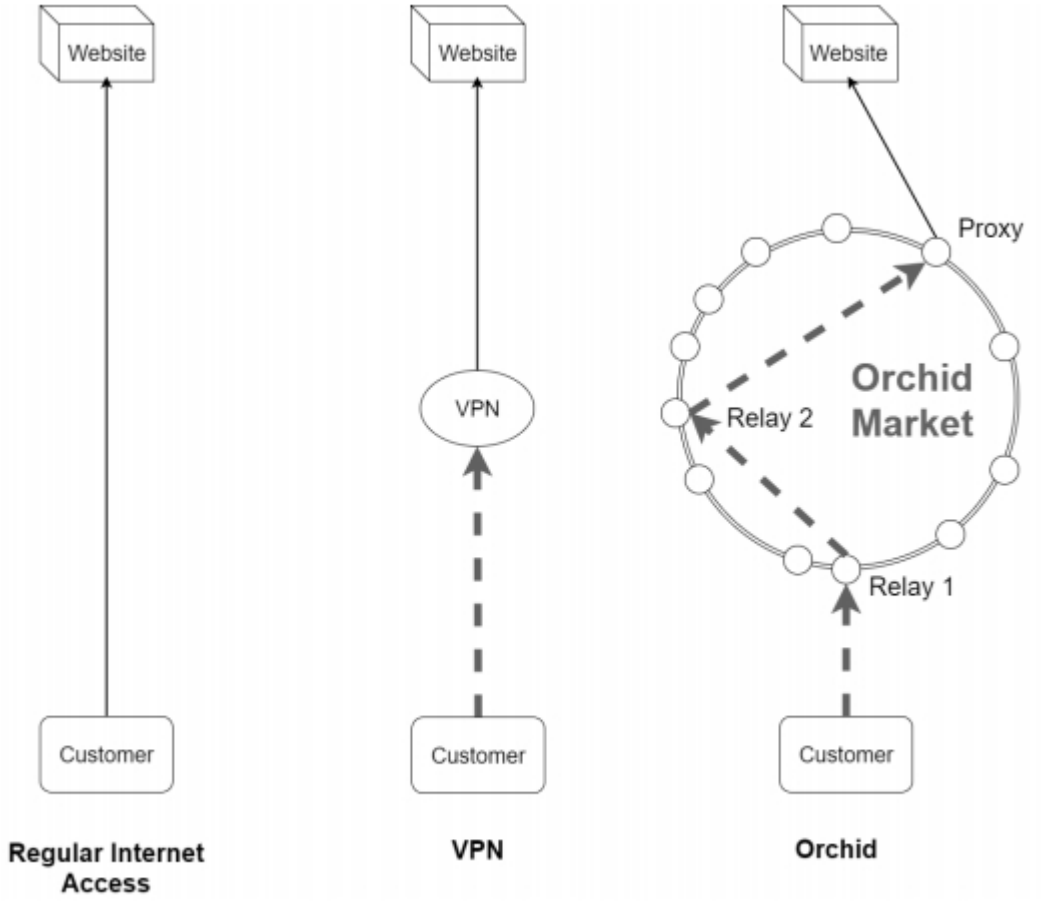


Figure 1: Direct connection, VPN connection, Orchid connection

与从全球互联网发送和接收数据的更常见的方法不同，兰花市场中的代理链自然地将关于数据的来源和目的信息分离;没有一个中继或代理同时持有两条信息，或者知道某人的身份。兰花市场的刚性结构进一步支持信息分离，提供强大的抵抗串通攻击的能力 — 一组带宽卖家克服知识分离的能力。不同于互联网数据常用的传输方法，兰花市场提供固定速率中继，以防止流量分析，并通过代币激励大家发现检举隐藏或者信息无关的参与的参与节点。在我们描述系统的细节之前，我们将简要回顾一下解决的核心问题，以及我们为系统基础选择的一般解决方案。

1.1. 信息加密传送问题

问题陈述：想象一下，你在一个充满数学家的食堂里，希望在没有任何人知道这个事实的情况下，向你的朋友发送消息。 您尚未通过指定协议传递消息，因此所有实施细节都必须向所有人公开声明。 可以怎么做呢？

Chaum 在 1981 年提出的这个问题的一个特别优雅的解决方案是让每个人都作为一个中继和一个接收者。 在这个方案中，参与者准备加密的消息，它们是数字等效的“包含信封的信封” – 向 Alice 发送消息，你会计算

```
Enc(“T oBob 00 ||Enc(“T oAlice 00 ||Enc(message, Alice), Bob), Carol)
```

并将该消息发送给 Carol，他们将其解密并发送给 Bob，Bob 将其解密并发送给 Alice。 为了防止流量分析，每个周期都会发送固定数量的消息。 为了处理返回地址，我们可以让 Bob 和 Carol 记住一个唯一的消息标识符，然后沿着这一条链路发送消息。

对使用上述方法的系统特别重要的是共谋的可能性。 如果 Bob 和 Carol 合作，他们可能会决定谁发送给定的消息以及发送给谁的消息。

1.2. 女巫问题

上述食堂问题陈述使用每个人本身来防止女巫攻击 – 一个参与者可能假装是任意大量的用户的情况。 不幸的是，在数字系统中，这种方法是不能使用的。

问题陈述：我们如何知道某人在纯数字环境中是“真实的”？

这个问题的解决方案可以在 Hashcash [81]中找到。如果我们要求声称是“真实的”来消耗计算资源的话，那么我们可以把女巫攻击者置于只有拥有不可思议的计算资源才有能力攻击的位置。

1.3. 随机选择问题

上述食堂问题陈述假定了一种向系统的每个其他用户发送消息的简单方法（例如，横跨自助餐厅）。为了实现最大限度地抵抗“共谋攻击”的 Chaumian 组合，我们需要能够从那些“真实”的中继节点中随机选择。无论何时加入或离开网络，都需要收到通知。不幸的是，在现实世界的 P2P 网络中，每个用户维护这样的列表将导致不可接受的网络流量（O（n 2）通知）。

问题描述 ：我们如何维护所有当前“真实”中继节点的分布式列表，以最大限度地减少网络开销，并支持高效率的随机选择对等体？

Chord [81]分布式哈希表（DHT）中可以找到一个特别优雅的解决方案。 在该方案中，对等体在大空间中分配唯一的地址，然后以在 O（log（n））时间内执行查找的方式连接。 添加或删除用户只需要通知 O（log（n））对等体。

1.4. 系统总览

“兰花协议”的核心是上述解决方案的组合。在我们的方法中，同行们需要制作奖章来证明他们的“真实性”，然后被组织成一个称为兰花市场的分布式 P2P 网络。为了保持兰花市场的参与者诚实，每个同行都会检查其邻居行为的正确性。客户然后使用兰花市场为 Chaumian 消息转发选择随机对等体。为了激励参与，我们的客户以每转发字节为单位向中继节点支付报酬。

这想法虽然简单，但依然需要魔鬼的细节。系统要完全分散，完全自主，完全匿名，并且能做金融交易。因此，本设计文件的大部分内容集中在防止对客户安全，系统性能和系统经济性的攻击稳健。虽然攻击分析是重要的，并且将占用我们大部分时间，但最终只不过是市场运行背景下的必要条件; 如果你发现自己“迷失在森林里”，我们希望你将使用前面的总览作为你的北极星 – 系统的设计细节都是为了实现上述三个问题的真实解决方案。

2. 攻击

本文的大部分内容都是针对攻击预防，我们首先回顾一下关于这些攻击的文献对 P2P 网络尤其常见。

2.1. 攻击目标

2.1.1. 信息收集

兰花协议必须保卫的最大的攻击类型是 揭示有关其用户的信息。因为兰花被实现为现有互联网上的叠加层，一些信息不可避免地与一些同行共享。 在下面的列表中，这样的信息被标记为“*”。任何未被明确列出的信息在本文档中不可避免地共享，但是发现了一种方法来发现信息被称为信息攻击，并被 Orchid 的 White Hat Bug Bounty 所覆盖。有关共享内容的更多信息，请参见第 8 节中的协议规范和第 9.2 节中的串通讨论以及网络参考实现[1]。

被认为是攻击者感兴趣的数据的类型（永恒的）：

- 现实世界的身份信息。 如用户的姓名， SSN， 地址等。
- 网站帐户信息。用户帐户在特定的网站。 注意这可以查找不同现实世界身份信息。
- *IP 信息。 用户访问兰花网络的 IP 地址。 请注意，在某些使用场景中，这可能等同于知道“真实世界身份信息”。
- *ethereum 信息。 与用户钱包相关联的*公钥和私钥。 请注意，在某些使用场景中，这可能等同于知道“真实世界身份信息”。
- *兰花网络信息。 与兰花网络上的节点当前业务相关联的密钥*（公钥或私钥）。

被认为是攻击者感兴趣的行为信息的类型（时间和链相关联数据）：

- *客户识别。 攻击者探悉客户的 IP 地址。
- *中继识别。 攻击者探悉中继的 IP 地址。
- *代理识别。 攻击者探悉代理的 IP 地址。
- *链接识别。 攻击者探悉到在链中使用了两个 IP 地址。
- *网站访问。 攻击者探悉到兰花网络进行了 outbound 连接到一个特定的网站。
- *Web 服务器访问。攻击者了解到，从兰花网络到特定的网络服务器（可能会托管多个网站）进行了出站连接。
- *以太坊联系。 攻击者了解到，一个兰花用户持有一个 Ethereum 公钥。
- *购买联系。 攻击者了解到两个交易共享一个付款人。
- *购买信息。 攻击者了解通过链发送的带宽的数量和时间。

尽管上述行为信息在正常操作期间与兰花网络上的其他节点共享，如下所述，在大多数情况下，假设用户只有在行为信息收集时才会受到直接的伤害，如果攻击者可以快速探悉几条信息。例如，要说用户 X 访问了网站 Y，攻击者将需要：买方识别，网站访问信息和几条链接标识。 因此，遵循参考规格的同行不会存储或共享上述任何信息，除非提供客户购买的服务所需。

2.2. 经济攻击

与相关系统不同， 兰花协议必须关心支付机制的攻击。本文归纳两大类如下：

-1. 经济利益。 有利的不良行为，例如用户提供“免费样品”带宽，允许用户专门使用免费样品带宽。

-2. 经济拒绝服务（EDoS）。 使用付款来淹没兰花网络上的另一个节点进行购买，从而使其脱机。

2.3. 服务质量攻击

一些对手可能会通过放慢兰花网络用户的系统性能来满足，从而潜在地减少使用。

2.4. 中间人攻击

只有在两个互动方之间插入自己才能执行的操作统称为中间人攻击。可以记录加密的信息用于分析元数据（第 2.8 节），而非加密数据可能另外被更改为控制行为。如果密钥交换没有得到保障，中间人也可能会欺骗双方使其错误地认为攻击者的密钥是对方的密钥。

2.5. 女巫攻击

伪装成多个用户执行恶意行为称为 Sybil 攻击，使被攻击者承受多方伪装用户的假冒数据。这种攻击的应用包括：

- 向 Yelp, Amazon 等平台提供多个 reviews。
- 通过假装是多个倾听者来实现 BitTorrent 从而可以快速下载[72]。

2.6. 日蚀攻击

在日蚀攻击中，攻击者的目标是隐藏系统的一部分。 采用的方法是一般网络相当于特权升级攻击：获得网络位置的控制权更多地控制网络，然后使用该控制来获取更多的控制权。

- 比特币的 p2p，所有节点都是平等地位，端口数是有限制的，需要"51%"攻击的比特币网络把临近节点的端口数全被攻击者给占用，以致用小于 51%的算力通过欺骗临近节点加入自己的网络，对比特币形成"51%"攻击。
- 通过接管与磁链接相关联的地址空间，从 BitTorrent DHT 中删除文件[83]。

2.7. 拒绝服务攻击

拒绝服务攻击即是攻击者想办法让目标机器停止提供服务。 "意外"情况下的系统行为通常指定和测试不足。 DoS 攻击对 P2P 网络中的节点进行去匿名化是非常有用的。

- 对特定目标 DoS 攻击结合女巫攻击来监测 Tor 的流量从而获悉匿名化的信息[53]。
- 只需要 20 个女巫节点，Dos 就能 完全可以泛滥式攻击 I2P 的数据库使其脱机，从而对网络上的所有流量进行归档。

2.8. 推理攻击

源于系统行为的统计建模的去匿名化被称为推理攻击或监视攻击。这些通常与精心制作或定时的"探测动作"相结合请求或其他攻击，例如 DoS 将特定对等体脱离网络并观察流量模式
响应。

- 从 SSL 加密的 Web 流量推断医疗疾病、家庭收入和最终用户的投资选择[58]。
- 来自全球传输日志的 Tor，I2P 和兰花流量的分析去除匿名[60]。
- 通过时序分析学习 OpenSSL 服务器的私钥[55]。

2.9. 黑客

通过将历史上可信赖的对等体转换为攻击向量，激励的攻击者可能会直接危及网络上的节点。当使用链部署带宽时，迭代黑客可能最终允许攻击者"回溯"连接。这种攻击具有重要的安全隐患，但不符合兰花网络的范围。 如果兰花网的设计达到目标，这将是对系统用户的主要攻击。

3. 可替代的方法

3.1. 不受保护的互联网访问

在没有保护的情况下访问互联网的用户将其完整的浏览历史记录和网站使用情况提供给 ISP，然后他们可以共享或出售该数据。

3.2. 虚拟专用网络（VPN）服务

虚拟专用网（VPN）使用加密技术将 VPN 用户的流量安全地传输到更大的不安全网络。 一旦 VPN 接收到流量，它将被解密并通过不同的大型不安全网络重新传输。 重传可以帮助用户规避网站的访问限制，并在较小的程度上减少对他们网站浏览习惯的跟踪。加密防止用户的 ISP 看到他们的流量，从而防止监视攻击。 这是通过使 VPN 成为用户的新 ISP 来实现的。 以前 ISP 可以执行的任何攻击都可以由 VPN 提供商轻松执行。

VPN 用户不应该认为他们的 VPN 提供商是值得信赖的。 尽管 VPN 服务提供商面临比 ISP 更多的竞争，但他们最终从相同的来源获得人才，并且具有类似的带宽 - 强制型商业模式。 VPN 提供商不可能不会受到导致用户不信任他们的 ISP 的同样的激励。 另外，在 VPN 设置中重复使用 IP 地址来中继流量，可以相对容易地阻止商业网站的使用[13]。

3.3. TOR

Tor [61]是一个免费软件项目，以向更广泛的受众介绍洋葱路由的思想而闻名。 在这个系统中，用户下载一个中继和出口节点的全局列表，随机从列表中选择，并从他们的选择形成洋葱路线。 洋葱路线是中继的有序列表; 为每个对等体轮流发送的数据包依次加密，确保每个节点都必须收到一个数据包才能被出口节点理解。结果是，除非有多个节点被同一个用户入侵或运行，否则两个中继都不知道谁发送了一个数据包，也不知道它到了哪里。

4. 扩展库

兰花的功能是建立在几个重要的组件上。 由于一些读者可能不熟悉这些原语，或者不熟悉兰花网络中使用的特定属性，我们在这里简要总结一下。

4.1. WEBRTC

WebRTC [48]是最初设计用来促进 Web 浏览器之间实时通信的系统。 它提供了优秀的 NAT 和防火墙穿越方法，包括 STUN，ICE，TURN 和 RTP-over-TCP。 通过选择 WebRTC 作为我们网络协议的基础，而不是定制编码的 TCP 和 UDP 网络代码，我们都获得了这些技术的世界级实施，并且（在一定程度上）将用户的流量掩盖为一般的网络流量。

4.2. NACL

NaCL [49]（发音为“salt”）是 Daniel J.Bernstein 等人的一个密码学库，专注于构建构建高级加密工具所需的核心操作。 它被选为这个项目的密码原语的来源，因为它和它的作者的英镑声誉。 下面介绍的所有加密操作都是使用 NaCL 实现的，除了以太坊智能合约加密代码。

4.3. 以太坊

以太坊[56]是一个分散的区块链平台，包括一个本地货币（ETH）和 Turingcomplete 智能合约。 这些智能合约对于 Orchid 的设计非常有用，使我们能够减轻与追踪付款余额有关的大量设计问题，以及 Orchid 付款门票的验证和公正性。

5. 奖章

完全分散的，完匿名的数字系统遭受单个恶意用户假装成千上万用户（Sybil 攻击）的攻击。

为了打击这类攻击，兰花协议使用奖章 - 数据表明给定的公钥在给定时间拥有相当数量的计算。 由于计算是一种昂贵的资源，因此使用奖章会对给定的攻击者假冒多个用户的能力造成预算限制。

5.1. 奖章规则

为了生成奖章，对等体采用公钥 K，并且最近的以太坊块哈希 E，然后（迭代地或并行地）定位盐 S，使得 H（K，E，S）≥ N，其中 N 是一些 难度缩放因子。

因为它是特定的密钥，所以不能用来模拟多个公钥。 因为它被绑在一个以太坊块哈希，不能预先计算。

5.2. 证明类型的选择

熟悉其他基于分布式市场的网络的读者将会认可 Medallion 在工作证明系统（比特币等）的前提下是相似的，并且可能会倾向于问：为什么不使用权益证明（proof-of-stake）， 空闲证明(proof-of-idle)，还是其他不那么有力的浪费方法来证明“真实性”？

权益证明取决于没有攻击者将控制大多数代币的假设。 由于我们的攻击模式包括压迫政府，这是不能指望的。 即使比特币的惊人的市值也远远低于一个中等规模国家的国内生产总值。更为复杂的是，在不久的将来，我们打算将这个系统扩大到支持匿名支付，这将使得发现这种“敌意收购”变得更加困难。简而言之：我们并没有使用权益证明，因为我们不想设计一个系统，让我们的用户的隐私权可以出售给出价最高的人。

空间证明(proof-of-space)看起来更有趣。尽管我们不确定将找到合适的方法，但我们会在即将到来的兰花协议版本中使用空间证明。 例如，这将允许用户在家中安装旧的智能电话作为中继和代理。有关这个想法的更多信息，请参见 15.1 节。

空闲证明基于额外的假设，即周期性同步工作证明足以证明用户对全局计算能力的分享。

遗憾的是，在网络尚处于起步阶段（少于 1,000 万传播者）的情况下，任何一家控制超级计算中心的公司都只要牺牲 1％的计算能力就能控制网络。正如我们所指出的，在我们没有拥有足够数量的传播者之前，我们不会使用空闲证明。

6. 奖章式工作证明

奖章构成了我们核心安全假设与整个网络之间的桥梁。由于我们的基本安全目标是限制激励攻击者获得对兰花网络的控制权，因此我们选择的奖章创作必须符合以下条件：

1. 对于非恶意节点来说，创建奖章必须非常容易。
2. 奖章必须很容易验证。
3. 奖章必须难以批量创建。

有了这些条件，我们将难度定义为在时间和金钱上的高度可伸缩性。 简而言之，我们需要一个工作证明系统，在这个系统中，一个正常的节点很容易进入网络，但攻击者难以进入网络。 我们将讨论我们选择的工作证明，而不是其他方法，例如权益证明或者空间证明。

目前存在两种主要的方法来满足上述要求：质询–响应协议和加密难题。不幸的是，质询– 响应协议可能不能在兰花模型中提供足够的安全性，因为攻击者可能通过合谋预先计算质询并作出响应。这留下了今天有很多存在的加密难题[51,76]，每个难题都有自己的权衡。同样，为了满足兰花的要求，只有这些密码拼图的一个子集是合适的。也就是说，平行化的密码拼图，制作成 ASIC，或平凡地缩放这样的手段是非常困难的。最近，研究人员发现了一些算法，可以产生易于验证的结果，这些结果具有可调的创建难度[51]。这些算法集合利用了内存和总硅片面积昂贵的趋势[46,62]。这类算法被称为不对称记忆硬功能，我们用它们来创建奖章。这些功能有几种类型[51,73,82]，但我们选择使用 Equihash。Equihash 基于 k-XOR 生日问题，通过时空折衷方案提供记忆硬度。由于 Equihash 是可调的，简单的，基于 NP 问题，并且已经在加密货币社区中被接受，所以我们相信使用这样的函数作为我们的工作证明基础提供了可接受的安全性和面向未来的水平。

为了产生奖章，一个同伴拿一个公钥 K，以前的以太坊块散列 E，然后进行一系列的计算，以便定位一个盐 S，使得 F（K，E，S，...）≥N 其中 N 是一些难度缩放因子。 当一个新的以太坊块被添加到链中时，必须计算一个新的 S 来保持当前的奖章。

7. 支付

7.1. 兰花支付需求

支付带宽是一个相当独特的挑战。 在大多数其他支付系统中，物品的成本远远高于发送包的成本，因此联网成本可能被安全成本忽略，被包括在交易成本中。然而，在兰花网络中，一个包的成本就是支付的价格，所以即使支付的交易成本低于一个包，它们的购买成本也是相同的。

因此，我们要求交易费用足够低，以便用户可以（自动通过 Orchid 客户端）支付任意数量的中继流量，直至单个数据包。除了低交易费用之外，支付机制必须足够精细，以至于可以进行微支付甚至纳支付。这不仅要求支付机制的效率，而且要求支付基础物品或可核实记录的可分性。

由于兰花网络的目的是为了摆脱网络监控和审查制度，支付机制的其他要求还包括不可靠，匿名，不依赖于可信任的第三方。即使底层网络不受监视和审查，如果付款机制不是，那么它就将使得用户可以被审查和跟踪。同样，可信赖的第三方可能在影响支付提供商的国家行为体和其他强大实体的干涉之下暴露兰花网络暴。

因此，兰花网络的支付需求包括：

1. 不可伪造性，只有拥有付款接受的抵押物品或可核实记录的所有者才能够使用它进行付款。 2. 可用性，意味着没有人可以阻止用户发送兰花付款，也没有人可以阻止收款人收到付款。

3. 不可逆转性，即使是付款方，也不可能扭转过去的付款。 4. 匿名，定义为发件人和收件人的不可链接性，无论是按帐户地址，金额还是时间，都找不到用户信息走过的痕迹。 理想情况下，匿名会达到如下的效果： 不仅是在恶意的观察者的攻击下，还是发送者或接收者的恶意攻击下，都能是匿名的。

在下面的章节中，我们将讨论支付的潜在解决方案，牢记这些要求。我们会辩证的去讨论，兰花支付（7.12 节）完成除匿名要求以外的所有内容。下面，我们继续讨论支付匿名，效率的扩展和改进。

7.2. 数据包的成本

为了讨论的目的，假设一个数据包长度为 1×103 个字节。 为了计算上限，我们观察到亚马逊网络服务的新加坡 CloudFront 是最昂贵的云服务之一，每 1×109 字节收费 0.14 美元。 这产生了每个数据包的成本为 1.4×10−5（$ 0.00000014）。由于带宽是一种浪费（任何未售出的带宽永远丢失），实际价格可能会明显低于这个上限。

7.3. 传统支付

在目前的金融支付系统中，交易是通过两个或两个以上的实体（如银行或支付服务提供商[2]）之间的谈判来解决的，支付卡采用 ISO / IEC 7816 [3]，银行支付采用 EBICS [4]。这些协议运行在 SWIFT [5]和 NYCE [6]等网络上，以支持国内和国际交易。组成这些网络的实体每个都保留自己的分类账，并不断从电子支付收据和手动调解中更新它们[7]。

连接到传统的支付网络通常需要大多数司法管辖区的特殊许可以及连接实体之间的逐案业务协议。由此产生的全球金融网络可以被看作是连接企业和协议和网络混合的一个许可的临时网络。每一个分类账都代表一个单一的失败点，缺乏密码的完整性，并可以随意的控制业务实体。

虽然传统的支付协议通常不会自行定义交易费用，但是运行协议的实体却增加了收费。 每笔交易费用可以从支付卡交易的几美分[8]到国际电汇[75]的 75 美元不等。 许多系统反过来又收取了一笔交易金额的百分比费用，对于银行转账支付可能高达 13％[10]，对支付卡支付则为 3.5％[11]。

由于传统付款取决于可信任的各方，因此在不牺牲我们所需的性能的情况下，实际上不可能使用兰花网络。特别是，可逆性是以逆转交易的形式设计出来的[75]。 交易通常很难伪造，但信用卡欺诈是常见的，身份盗窃或黑客攻击可能导致用户帐户被盗用。此外，这些支付系统仅提供部分可用性，因为它们往往在不方便的时候发生故障并定期遭受停机。由于管理支付的可信方通常不仅有发送者，接收者，支付金额和时间的记录，而且经常还有关于发送者的身份信息，所以缺乏匿名性。 最后，我们将在下面的章节中看到，传统支付的交易费用相对于兰花市场来说，将非常昂贵。

7.4. 区块链支付

比特币彻底改变了传统支付系统的现状，并继续扰乱全球支付和国际转移市场。比特币是一个全球性的网络和协议，不知道地理边界。 应用公钥密码术，交易在用户自己生成的地址之间转移比特币金额，而不需要任何可信任的方。 用户生成密钥对，其中公钥的散列可以用作支付地址，要求私钥签名从地址传输[12]。 比特币支付是不可伪造的，不可逆转的[77]（在合理的时间内以计算区块确认）。 比特币网络自成立以来停机时间最短，而矿工（除了第 7.6 节进一步讨论的）不太可能出现活跃的审查情况，因此可以将其视为普遍可用。比特币支付是伪匿名的，匿名程度在很大程度上取决于 如何使用网络[68]。

一般来说，分散式加密货币允许人类和计算机系统在历史上第一次在没有可信赖的第三方的情况下进行价值交易 – 激励的分布式覆盖网络（如兰花）。

比特币的交易费用不是由交易金额决定的，而是由交易数据结构的大小乘以发件人配置的因子决定的。 直到 2017 年，平均交易费仍远低于 1 美元，但随着比特币网络达到最大交易容量，2017 年 2 月费用迅速上涨。 平均费用上涨了 13 美元，高达 8 美元，使得依靠比特币网络上的低费用的应用程序成为可能。

以太坊网络也植根于公钥密码学，并通过像比特币这样的工作证明进行保护，从而获得了不可伪造性，可用性和（非经典）不可逆性的相同属性。 以太坊拥有更高的动态可调整的交易容量，自 2015 年推出以来，网络的收费水平一直很低。但是，由于交易数量增加以及以太坊基础本地代币 Ether 的交易费用（称为 gas ）已经增长[14]，平均为 0.20 美元，最高达到 1.00 美元。 执行智能合同代码的交易成本更高，与执行多少计算成比例。

流行的公共区块链网络中的交易费用的增长，阻碍了他们直接处理小额支付的潜力，将小额支付推向支付通道等第二层解决方案。

7.5. 以太坊交易成本

以太坊智能合约允许创建复杂的支付机制，利用以太坊虚拟机[85]（EVM）的能力和灵活性，在经济范围内提供图灵完整的执行环境。 以太坊智能合约执行的每条指令都会增加原始交易的交易费用。

每个 EVM 指令花费一定量的 gas，以太坊交易费用定义为交易所花费的总 gas 量乘以发送方设置的 gas 价格。 矿工选择任何有效的交易纳入其开采块，可以包括交易与任何 gas 价格，包括零。 选择 gas 价较高的交易可能导致更多的利润，因为每个区块都有可以包含多少交易的限制。 同样，接受较低的 gas 价格也可能导致更多的利润，因为如果网络没有以最大容量运行，它可以允许矿工填满他们的区块。 这种机制创造了一个不断变化但稳定的博弈理论平衡，这个平衡被以太坊加油站这样的网站跟踪[15]。

截至 2017 年 10 月，在几个区块内获得高概率的交易成本为 0.026 美元。 要在 15 分钟内确认，$ 0.006 就足够了。 这些估计是交易的基本成本 – 21,000gas，无需执行任何明智的合同代码即可进行简单的以太币转移交易。 如果交易执行智能合约代码，则每个 EVM 指令都会增加额外的 gas 成本。 例如，在智能合约存储中永久存储新的 256 位值需要花费 20,000 个 gas，更新现有值需要花费 5,000 个 gas。

作为 Ethereum ERC20 分类账本仅仅是账户地址到余额的映射，ERC20 代币转账的成本应该是：新账户需要 21,000 + 20,000gas，随后老账户转账需要 21,000 + 5,000gas（因为收款人账户已经有了 代币分类帐本）。 观察现场[16] ERC20 交易，我们看到 gas 成本稍高，约 52,000 和 37,000gas 转移到新的和现有的帐户。 不同之处在于智能合同代码执行不变式验证，例如发件人是否具有足够的余额以及其他实现细节（如付款收据的记录）。 50,000gas 需要交易费用 0.014 美元至 0.062 美元之间，这取决于我们希望交易确认的速度。

7.6. 兰花代币

兰花网络使用基于以太坊的 ERC20 代币，以满足不可伪造性，可用性和不可逆性的支付要求。 以下部分将讨论我们如何降低 ERC20 转帐的交易费用以实现任意小的代币数额。 匿名在 7.17 节中讨论。

兰花代币（OCT）用于兰花网络内的支付。 兰花代币是基于以太坊发行的固定供应量的 ERC20 代币。 供给固定在 1×10^8 个， 每个代币有 1×10^18 个不可分割的子单元（与 Ether 相同的可分解性）。

乍一看，以下部分详细介绍的兰花支付系统可以配置为使用 Ether 或任何 ERC20 代币。 事实上，使用以太币将简化票务合同，稍微降低交易费成本，提高可用性，因为用户只需要 Ether，而不必同时购买兰花代币和 Ether（交易费用）。

然而，以太坊计划未来的协议升级，允许交易费用由任意机制支付，包括 ERC20 代币[17] [18]。 这将消除使用新代币的大部分缺点; gas 成本没有任何差别，用户只需要购买一个代币。 也可以将 gas 价格设置为零，并在合约执行中向采矿者添加 ERC20 代币付款（使用 EVM COINBASE [85]操作码）[19]。 这需要矿工的明确支持，因为他们需要将其采矿策略配置为接受零 gas 价格，并验证交易执行包括将 ERC20 代币转移到

coinbase 地址。

但是，引入新的代币而不是简单地使用 Ether 的决定是出于社会经济而非技术原因。 通过创建一个新的代币，并使其成为兰花网络中唯一有效的支付选项，我们设计出了我们认为足够重要的社会经济效应，以保证增加的复杂性。

7.6.1. 激励

激励是通过赋予人们对网络的部分所有权来引导新的协议和网络的一种方式[20]。

像兰花这样的新型去中心化网络受到鸡和鸡蛋的困扰。 代理和中继节点越多，网络为用户提供的实用程序就越多。 而用户越多，运行代理或中继节点就越有价值。 通过部署一个新的网络代币，可以加速网络效应，因为所有潜在的用户都被激励来尽早使用网络。

7.6.2. 解耦

在去中心化系统中做去中心化系统，新的代币将新系统的市场价值从基础系统中分离出来。 例如，截至 2017 年 10 月，Ether 的市值约为 300 亿美元，日均全球交易量为 5 亿美元[21]。 以太网的价格受多种因素影响，如加密货币的整体猜测，以太坊矿工的哈希能力以及以太坊建立的数百个项目的成败。 然而，单个项目的失败或成功可能不会对以太网的价格产生重大影响，但会对所涉及的项目具有显着的影响。 使用新的代币解耦市场价值，创造了一个更好的项目和系统的规模和健康的指标，有效地预测该市场的未来。

7.6.3. 流动市场

对于系统特定的代币而言，流动性市场可以使严重依赖系统的用户通过做空仓位来对冲潜在的系统故障。 如果这看起来很远，我们应该注意到，金融衍生工具的初衷是允许企业对不幸的未来事件进行对冲。随着 0x [22]和 etherdelta [23]等去中心化交易以及 Augur [24]和 Gnosis [25]等预测市场的出现，基于以太坊的代币和系统的衍生品也不算太远。 事实上，这样的衍生工具可能比传统的金融衍生工具更有效[26]，因为前者没有信任方，没有权限，甚至可能是匿名的。

7.6.4. 新代币

新的代币也可以更容易地为利益相关者设计具体的激励措施; 因为代币完全是从新系统中获得价值的，所以它们对任何为了系统成功而努力的人都是强有力的激励。 以太坊智能合约可以实现代币的自主锁定，以确保代币持有者只能根据定义的时间表访问其代币。 这种激励措施随着时间的推移而调整，并将代币持有者的重点放在系统的长期成功上，而不是像特定团队或相关公司这样的社会结构。 如果兰花网络使用了 Ether，并且利益相关者被锁定了 Ether，他们实际上会更加激励地为以太坊的整体成功而努力，而不是使用以太坊的任何特定系统。 可以认为，这样的结果将不是一个兰花网络和项目的最佳激励调整。

7.7. 以太坊审查制度的抵制

与大多数公链类似，除非验证者（以太坊网络中的矿工）选择不将兰花交易打包，否则一定会对以太坊交易进行审查验证。由于所有矿工都是随机打包出块，与哈希能力成正比，这就要求大部分矿工主动审查兰花付款来保护兰花网络。例如，即使 90％的算力选择不打包兰花相关的交易，兰花网络仍然会运作，只是交易所需的平均时间是正常的十倍。如果一大群 51％的矿工选择通过拒绝包括他们在内的区块来审查兰花相关交易，那么更严格的审查形式就是如此[71]。根据以太坊协议规则，这是有效的，并有效地创建一个软分叉。但是，组织大规模的矿工勾结制造这样一个软叉有很大的利润损失风险;如果软叉未能获得足够的哈希能力，那么勾结的矿工就会错过他们的区块奖励。除了利润风险之外，考虑到以太坊矿工的去中心化性和对区块链采矿策略的法律和法规限制，我们认为这种可能性极小。

7.8. 在宏支付上建立微支付

现在讨论交易成本和支付令牌的选择，现在让我们看看可行的支付方式。 以区块链为基础的小额支付面临的一个根本性挑战是如何避免交易费用。 想象一下，如果我们发送一分钱作为一个简单的以太坊 ERC20 交易，我们会支付 1.4 美分 – 140%的交易费每次支付！ 有效的小额支付要求降低交易费几个数量级。

在 MojoNation [27]中采用的一个潜在有趣的方法是在每对节点之间建立一个“贸易平衡”。 当带宽在它们之间流动时，等交易费从 0 达到了一定的值时，它们会定期结算。 但是，正如我们所看到的，使用以太坊交易结算付款的交易成本至少会导致 0.014 美元的交易费用。 根据之前讨论的上限，我们可以看到这个价格大约等于 140 兆字节的带宽。 这种方法的第二个问题是，恶意邻近节点会知道这个事实，并试图断开连接并创建一个新的身份，而不是支付费用。

7.9. 支付通道

在比特币网络上首次出现的区块链应用中流行的技术是支付通道[28]。由中本聪[66]部分描述，后来由 Hearn 和 Spilman [29]定义和实施，支付通道后来由 Poon 和 Dryja [30]把它应用在比特币闪电网络。支付通道允许发送者和接收者在彼此之间发送任意数量的交易，并且仅支付两笔交易的交易费 – 一个用于设置支付通道，另一个用于关闭它。这是通过首先让发送者发送一笔交易来锁定一些代币，这些代币可以被发送给收件人或发回给发送者。通常，代币只能在将来的某个时间 T 被发回给发送者。同时，代币可以（递增或全部）发送给接受者。发送者持续签署的交易花费越来越多的代币给收件人，并直接发送给收款人，无需发布在区块链上。收款人可以在任何时候，向区块链发布他们最后收到的交易要求汇总的金额， 直到时间 T 后，发送者没有意义，正式确定了他们多次协商后的交易。

支付通道为发款人提供一个有效的方式，为收款人提供连续付款的密码证据。 由于中间付款不会产生任何交易费用，所以可以随意小额付款，任意发送。 在实践中，瓶颈成为验证交易的计算开销以及发送它们的带宽要求。

尽管支付通道有效地为任意数量的中间支付提供了不变的交易费用复杂性，但不是在所有的情况下，这些支付通道效率有这么高。 特别是在有大量发送者和接收者的系统中，他们经常与他们互动，他们不断创建新的支付渠道可能太昂贵了。 同样，对于提供的非常小的或短期的服务（如单个 HTTP 请求或 10 秒的视频流），所需链上交易的交易费用可能太高。

7.10. 概率支付

如果我们无法避免必须在区块链上做支付结算并产生交易手续费，那么理论上的最低成本就是单个交易的成本，因为区块链需要至少一个交易来执行状态转换。 为了解决一些（微）支付，我们至少需要一笔交易。

如果我们可以取消支付通道所需的设置交易（也就是为了建立通道而产生的第一笔交易），并且仍然能够向收件人证明他们正在获得付款，那该怎么办？

幸运的是，在区块链行业有一个类似的解决问题：矿池算力共享[31]。 随着像比特币这样的网络工作的工作难度的增加，矿工们开始将他们的计算能力集中在一起，以避免单个矿工花费数年的时间寻找块解决方案（也就是挖矿合适的 nonce 查找）来产生块。 矿池按矿工算力占矿池比率给予奖励，个别矿工通过哈希证明他们的哈希能力不断发送解决方[32]，在相同的区块上做哈希计算，但会选择一个较低的难度。 该技术使得矿池能够以密码方式验证每个池成员的哈希能力，而不管该池成员是否找到满足实际工作证明目标的解决方案。

如果我们将相同的思想应用于支付通道，我们可以构建概率支付方案，发件人不断向受方证明他们是平均支付的，而不管实际支付是否发生。 这使我们能够创建概率性的微支付，而不需要设置交易（闪电网络和雷电网络需要在建立通道时向区块链发送第一笔交易用于锁定定量的代币），接收者只需在“兑现”时支付交易费用。

在我们研究如何使用以太坊智能合约来构建这样的概率微支付之前，让我们退一步观察一下，概率支付的最初概念早于区块链技术，并于 1996 年由 David Wheeler [84]首次发表。Wheeler 描述了核心 概率支付的概念以及如何将其应用于使用随机数字承诺的电子协议，使得发件人和收件人（论文的术语中的买方和卖方）都不能操纵概率事件的结果，同时也证明他们之间的获胜的概率是多少。

有几篇论文跟随惠勒的想法，1997 年 Ronald Rivest [79]发表了一篇论文，描述了如何在电子微支付中应用概率支付。 2015 年，Pass 和 Shelat [78]描述了如何将可能性微支付应用于比特币等去中心化货币，并指出先前的方案都依赖于可信的第三方。 第二年，Chiesa，Green，

Liu，Miao，Miers 和 Mishra [59]将这项研究扩展到零知识证明，提供适用于加密货币协议的去中心化和匿名微支付。

鉴于最近在以太坊系统中支付通道的兴趣和普遍性，从支付通道的角度来看待可能的支付可能是有价值的。 为了省略第一次设置交易，我们失去了保证发送确切金额的能力，取而代之的只是一个概率保证。 然而，我们将通过调整支付概率，支付金额和支付频率来展示可能的小额支付，以便它们能够取代几类基于区块链的应用的支付通道，而不存在明显的缺陷。

从本质上讲，我们可以避免初始设置交易，我们可以从同一个发送者帐户中为任意数量的收件者支付任意小的服务会话，同时还向每个人证明支付金额的确切概率。假设服务提供商（兰花网络中的一个中继或代理节点）提供了足够的服务量，那么概率支出的变化就会很快达到平衡。

7.11. 基于区块链的概率微支付

为了更容易地传达如何将概率支付应用于区块链协议的核心思想，我们将在这里详述几个细节。 在引用的原文中提供了对 MICROPAY1 方案的正式描述，而兰花的概率支付方案在 7.12 节。

Pass 和 Shelat 描述了 MICROPAY1 [78]，组合了数字签名和承诺方案，来达到发布精确概率的随机结果的条件。发件人首先通过将比特币转移到新生成的密钥的托管地址来进行“存款”。 然后，收款人（MICROPAY1 条款中的商家）挑选一个随机数字，并将此号码的承诺发送给发款人。除了承诺，收款人还提供了一个新的比特币地址。发款人也挑选一个随机数字，并对这个数字（明文的形式），收款人的承诺和其他付款数据（如收件人提供的付款目的地地址）等一系列数据签名。

验证所产生的票据包括检查收款人的承诺是否符合他所披露的数字，以及验证发款人的签名是否与比特币存款的地址相符。如果来自发款人和收款人的随机数字的 XOR 的最后两位数字是 00，那么票据通过验证，并且可以由收款人使用。

直觉上，我们可以把这个方案中的“抛硬币”看作是没有偏见的，除非发送方可以打破承诺的约束性（或伪造签名），或者用户可以打破承诺的隐藏性。

请注意，发款人可以通过向多个收款人并行的发行票据来“双花”其存款，通过在收款人看到票据声明前广播花费，达到在收款人之前发费该存款的目的。MICROPAY1 的作者讨论了如何通过“惩罚托管”来解决这个问题，由发款人存入的第二笔金额做抵押，可以在未来某个时间点返回给发款人，但这笔抵押可以被任何可以为同一付款托管提交两张有效票据的人“削减”或“烧毁”。这可以防止发款人与收款人勾结或作为自己的收件人。

MICROPAY1 的作者在 MICROPAY2 和 MICROPAY3 中构造了迭代改进，其中引入了一个可信方，在票据上执行一些计算验证步骤，并在计算正确的情况下释放签名。

7.12. 兰花支付方案

现在我们已经为我们的支付找到了合适的抽象， 接下来是该如何去实施？

除了第 7.1 节中讨论的要求外，我们还要满足：

- 可重用性，构建每张新票据时不需要每一张票据都产生交易费或必须上链交易，否则交易费将再次成为问题。
- 必须防止双花， 否则会给收款人造成损失。
- 就计算成本而言，系统必须具有足够的性能，以免超出数据包的成本。

在这些要求中，最后一个要素可能是最麻烦的。 据我们所知，不存在不需要按照验证 ECDSA 签名的顺序进行计算的方法在以太坊上构建彩票的 dapp。正如本节所详细描述的，从发件人的要求来看，不仅要证明收件人的票据金额和获胜的可能性，而且还要求发件人的以太坊账户有足够数量被锁定的兰花代币作为发送的门票。

出于这个原因，虽然单靠使用还不够，但我们不得不采用与上述类似的贸易平衡方法。这反过来导致了一个新的要求，即“贸易平衡必须保持足够小，以免在贸易过程中造成激励断开”。由于这是一个由实现实际问题而引起的机制设计问题，现在让我们通过假设一个解决方案来关注实现，并推迟到 7.15 节进一步的讨论。

兰花支付方案受 MICROPAY1 和相关构造的启发，是一个伪造的匿名，概率微支付方案，。通过利用以太坊智能合约和代币锁定抵押惩罚机制，它可以减少运行前和并行（包括双花）支出攻击，而不需要可信任的第三方。兰花支付的伪匿名等同于在以太坊定期交易中可以实现的功能（尽管兰花客户使用额外的隐私技术，例如一次性地址和节点身份与支付地址之间的关键分离以实现有限的匿名）。

MICROPAY2 和 MICROPAY3 中引入的信任方可以被 Ethereum 智能合约代码有效替代。EVM 允许在验证微支付票据时实现任意逻辑（在计算的经济范围内），并为 ECDSA [70]恢复操作以及加密哈希函数提供原语[85]。

7.12.1. 支付票据的定义

兰花支付票据有如下字段：

```
H(function) --- 哈希函数（更多细节在 7.12.2）
timestamp(uint32) --- Unix 时间表示票据的值何时开始呈指数下降
rand(uint256) --- 收件人选择的随机整数
nonce(uint256) --- 票据发送者的随机整数
faceValue(uint256) --- 赢得票据的值
minValueMarket(uint256) --- 基于带宽市场的票据值
minValueAccepted(uint256) ---  基于收件人所接受的内容的预期值
winProb(uint256) --- 特定票据赢取发送者面值的概率
recipient(uint160) --- 票据接收者在以太坊上 160 位以太坊账户地址
randHash(uint256) --- 随机数哈希摘要
ticketHash(uint256) ---
randHash,recipient,faceValue,winProb,nonce 的哈希摘要
(v1,r1,s1)(tuple) --- 票据发送者的 ECDSA 签名元素
(v2,r2,s2)(tuple) --- 接收者的 ECDSA 签名元素
```

7.12.2. 支付票据加密库选择

为了降低兰花小额支付的成本，我们选择了某些密码功能，因为与其他任意函数相比，以太坊的 gas 成本降低了。

H — – Keccak-256 – 由于具有最低的 gas 成本（用于哈希 32 个字节需要花费 36 个 gas[85]），所以在 EVM 中可用的所有哈希函数中花费是最低的。

ECDSA — secp256k1 与 Keccak-256，由于 EVM 支持 ECDSA 恢复此曲线以及兼容现有区块链软件库和工具。

7.12.3. 生成支付票据

Alice 作为接收者，Bob 做为发送者。

-1. Alice 挑选一个随机的 256 位数 rand，计算 randHash，并将摘要发送给 Bob

-2. Bob 初始化参数（nonce, faceValue, winProb,recipient）

-3. Bob 计算票据哈希

-4. Bob 产生签名(私钥，票据 hash)

-5. 票据定义后产生的内容：

```
(a) randHash
(b) recipient
(c) faceValue
(d) winProb
(e) nonce
(f) ticketHash
```

```
(g) creator (签名 ticketHash 的发件人密钥的地址)
(h) creatorSig (发件人对 tickerHash 的签名)
```

请注意，虽然此票据只要收件人可以完全验证就是有效的，但收件人需要对其签名（请参见下文），以便能够在以太坊上的兰花支付合约中声明。

7.12.4. 支付合约的验证

Alice (带宽销售者) 可以执行下面的操作，

```
Verify:
(a) randHash == H(rand)
(b) faceValue >= minValueMarket
(c) winProb >= minValueAccepted
(d) recipient == {接收者公布的以太坊帐号地址}
(e) creator == {发送者公布的以太坊帐号地址}
Validate:
(a) Validate: 用公钥验证是否是该公钥对应的私钥所做的签名
Check:
(a)Validate: 交易发送者在兰花支付合约中锁定了足够的兰花代币
```

现在票据被证明是有效的，并且可能是一张中奖票

7.12.5. 领取票据中的付款

虽然接收者可以在本地充分验证票据是否有效，并且如果它是中奖票据，中奖票据中代币的实际支付是通过兰花支付智能合约完成的。

这个智能合约公开了一个以输入为参数的 Solidity API：

```
1. rand
2. nonce
3. faceValue
4. winProb
5. receipient
6. recipientSig (接收者对票据 hash 的签名)
7. creatorSig (发送者对 ticketHash 的签名)
```

7.12.6. 合约的执行

假设 Alice 是希望购买带宽的用户。 Alice 必须有一个以太坊帐户地址 addressAlice 和兰花代币。 请注意，该地址将具有关联的公钥 PubKeyAlice。Alice 还必须将兰花代币锁定在以前部分定义的以太坊智能合约中，并使用 PubKeyAlice 锁定。 在上一节中，Alice 的地址就是以太坊帐户地址，等于从 ticketHash 上 creatorSig 恢复的公钥。

假设 SLASH 是一个临时布尔值，它被设置为 FALSE，PubKey 是通过 ticketHash 从 recipientSig 恢复的公钥，

```
计算：(a) ticketHash
验证：
(a) randHash；如果不是， 结束执行。
(b) PubKey == recipient address；如果不是，结束执行。
(c) addressAlice 将代币锁在罚金托管账户。如果不是，结束执行。
(d) addressAlice 已经有足够的兰花代币被锁定在门票帐户中以支付门票。 如果不是，则将 SLASH 设置为 TRUE 并继续执行。
(e) H(ticketHash, rand) <= winProb。 如果不是，结束执行。
判断：
(a) 如果 SLASH == FALSE，则支付票款：faceValue 从创建者的票据转移到收件人。
(b) 如果 SLASH = TRUE，则创建者锁定的代币中扣减。
结算：
(a) 发送创建者的票据资金（如果有的话）给收件人（这是来自之前的验证，保证小于 faceValue）。
(b) 将创建者的惩罚托管帐户设置为零（销毁/删除这些代币）。
```

请注意，虽然惩罚抵押机制消除了发送者潜在的双花作恶，但票据发送者仍然有大规模超支的危险。为了解决这个问题，获胜票据的额度应该在时间戳上呈指数下降，从而为获胜者立即兑现提供了强有力的激励。接收者可以使用这种直接性来计算发送者的兰花代币余额的"浪费率"。

7.13. 兰花 GAS 成本

我们已经从上述方案的可靠性原型实施中测量了约 87,000 的 gas 成本。这个成本消耗在对获胜票据声明的 API 调用中。对票据合约的执行过程包括对兰花代币转账 API 的子调用。所有兰花智能合约在密码审查和外部安全审计之后， 会部署到以太坊主网并开源。

7.14. 可验证的随机函数

通过用可验证的随机函数（VRF）替换接收者的随机数字承诺，可以减少前面部分中描述的支付票据的交互性。 由 Micali，Rabin 和 Vadhan [80]于 1999 年首次发表，IETF VRF 草案最近由 Goldberg 和 Papadopoulos [33]提出。该草案规定了两种对 VRF 的构造，一个使用 RSA，一个使用椭圆曲线（EC-VRF）。

使用 VRF，兰花支付票据的发件人将能够在无需得到每个票据接收者的承诺。 相反，发件人只需要知道收件人的公钥。 发件人将用此公钥替换之前描述的票据方案中的随机数哈希。 为了提高效率，这可以是接收票据中已经存在的资金的接收方公钥，但要坚持密钥分离的密码原理，可能需要第二个密钥。

然而，验证兰花支付合约中的 EC-VRF 将需要明确的 EVM 加速椭圆曲线操作，因为直接以 solidity 语言或 EVM 代码组装方式实施它们在 gas 成本方面将是非常昂贵的。

幸运的是，在以太坊 Byzantium [34]版本中，以太坊网络增加了对椭圆曲线标量加法和乘法的 EVM 支持[35]以及 alt bn128 曲线的配对检查[36]。 EC-VRF 结构是针对任何椭圆曲线而定义的，IETF 草案特别将 EC-VRFP256-SHA256 定义为 EC-VRF 密码组（其中 P256 是 NIST-P256 曲线[54]）。 但是，似乎没有理由 在同样足够安全的级别下，不去使用 alt bn128 曲线。 而且，SHA256 可以用 Keccak-256 来代替。这将允许在以太坊智能合约中进行 VRF 验证，从而与兰花智能合约进行整合。

然而，尽管在 zcash 中使用了 alt bn128 曲线，但是与 P256 相比，这是一个更年轻的曲线，并且没有被研究。 也许更重要的是，EC-VRF 的构建是一个早期草案，正在等待审查，EVM 拜占庭升级发生在撰写本文时，并且尚未在现场系统中证明具有重大价值。 因此，在兰花概率微支付中使用 EC-VRF 并不是立即可行的，兰花项目的目标是进一步研究使用，例如 可以先试用 EVM 验证的 EC-VRF-ALTBN128-KECCAK256 结构。

7.15. 贸易平衡

如前所述，对称加密性能的现实阻碍了我们每个数据包的支付，所以我们需要很好地理解采用“贸易平衡”方法所固有的风险。 我们这样做的一般情况是：想象一下 Alice 和 Bob 希望以完全匿名的方式进行交易。 Bob 要执行一些他所要求的任务，而 Alice 每工作一次就要付一次钱。 不幸的是，匿名的性质是这样的，没有事先交易，Alice 和 Bob 没有机制互相信任。 他们能合作吗？

如果 Alice 和 Bob 的关系有一些启动成本（SAlice，SBob s.t. SAlice> xy，SBob> xy），答案是肯定的：逃避金钱或工作不再是经济上的理性，除非（1）Alice 所要求的总工作量≤xy 或（2）Bob 能够完成的工作总量≤xy。正如我们在讨论兰花市场时（第 11 节）所看到的那样，兰花网上存在的启动成本超过 1×103 数据包就会贸易不平衡。因为兰花市场的卖家一般比买家支付更高的启动成本，而且由于客户不对称地知道需要多少工作，兰花网络有客户预付款。

7.16. 改良

7.17. 匿名

前面几节讨论的兰花支付与普通以太坊交易一样是伪匿名的; 所有交易都是公开的，包括金额以及发件人和收件人帐户。兰花客户旨在通过现代钱包技术（如一次性地址[37]和使用分层钱包[38]）来改进公共区块链交易的默认伪匿名性，并使用分层钱包[38]来提供尽管使用单个根密钥的支付地址的不可链接性。

随着以太坊拜占廷版本的发布，现在可以通过利用新的 EVM 椭圆曲线原语操作码来实现具有合理 gas 成本的可链接环签名[39]。 将以太坊智能合约与隐藏地址（如 HD 钱包和可链接环签名提供的地址）相结合，可实现一类混合技术，如 M¨obius [74]混合服务。如 M¨obius 提供了强大的匿名性保证，通过使用基于博弈的安全模型对混合服务进行密码验证。然而，与以前的混合技术不同的是，它提供了针对恶意观察者和发送者的匿名，而不是针对恶意接收者。 将 M¨obius 等服务与兰花概率小额支付相结合，使我们更接近于我们对付款的最终要求 – 匿名。

为了实现完全匿名保证，防止任何恶意行为者，无论是观察员，发送者还是接收者，我们都要依靠零知识证明技术。

在 zcash 网络[40]中应用的 zk-SNARK [47]技术与环签名相比可以提供更强的匿名性保证。 在 zcash 中，屏蔽地址之间的交易提供了发送者，接收者和金额的完全匿名。

7.18. 非交互式

在 7.14 节中，我们表明，通过用 VRF 替换兰花付款方案中的随机数承诺，通过去除与随机数承诺相关联的通信步骤使得该方案更加非互动。 接收方不必在发送者构造票据之前将承诺传达给发送者，而只需从公开的接收方信息中立即构建票据单。

每个接收者将生成一个专门用于 VRF 的新密钥对，并将公钥与其他公共接收者信息一起发布，详见 11.1 节。发送者只需在票据中配置该公钥，而收件人将使用相应的私钥对收到的票据进行签名。在 7.12.4 节中定义的票据验证逻辑将把接收者 VRF 签名解释为它与获胜概率阈值进行比较的值。

正如 7.14 节所讨论的那样，虽然这将是对支付方案的相对简单的修改，但在 EVM 中验证 VRF 的可行性需要进一步的研究。

7.19. 性能

虽然兰花智能合约是不可变的，但是他们可以通过部署新的合约和升级兰花客户端软件来指向他们（如果有需要的话也可以向后兼容旧合约）来有效地进行升级。以太坊智能合约支持多层优化以降低 gas 成本，我们预计兰花支付智能合约的未来版本将使用 例如 EVM 原语[41]（也就是预编译的代码）来优化 gas 成本，类似于常规软件系统常常用内联组装代替昂贵的子程序。

然而，兰花支付票据的验证瓶颈是诸如 ECDSA recovery 等密码学相关操作的执行以及兰花代币发送者和接收者的以太坊账户的状态更新。在这里，一个改进可能是两次使用接收者对有效数据交易的签名。 目前，由于兰花客户灵活性的原因，兰花计划在那里定义了两个签名，并且使得在

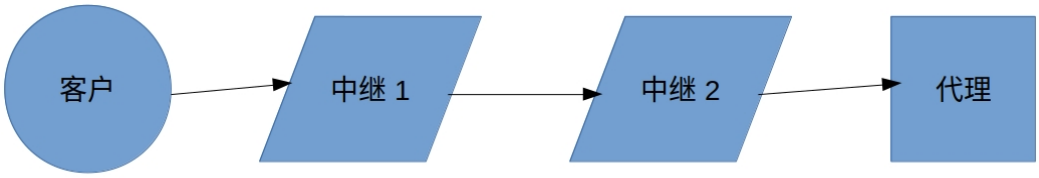
不依赖以太坊细节的情况下更容易指定和推理付款方案。 更简单的优化包括紧密包装票据字段并将单个 256 位字中的多个内部变量进行编码，以与 EVM 堆栈字和永久契约存储插槽（均为 256 位）对齐。

另一方面，为了实现更大的匿名性，可选或甚至强制使用混合技术可能会大大增加兰花支付合约的 gas 成本。 使用基于可链接环签名的混合服务很容易导致大约高出一个数量级的交易费用[39]。但是，这能提供给用户强有力的匿名保证，用户可能会觉得这是值得的。 因为我们可以很容易地调整兰花支付的概率变量 – 票据频率，获胜概率和获胜金额 – 我们可以调整票证之间的平均时间来减少交易费用（特别是对于长期运行的节点，可以隔个几天才做结算一次）。

最后，零知识技术如 zk-SNARKs 的一个非常有趣的属性是大大减少任意计算的计算开销，如以太坊智能合约执行[42]。虽然生成 zk-SNARK 证明是昂贵的，但验证更便宜 – 甚至与原始代码相比。 由于只有验证需要在链上执行，所以在兰花票据中采用零知识证明比原始验证码更便宜。

进一步来说，递归 SNARKs [52]有可能将一组 SNAK 证明集合成一个证明。虽然它们可能更适用于区块链共识协议[43]，但它们也可能对兰花协议有用，例如 将多个票据索赔批量化为单个智能合约交易，同时避免线性 gas 成本叠加。

8. 带宽挖矿



在这章我们将描述中继的规则和代理的行为，以及讨论这些无法审查的，匿名网页浏览的节点如何“连接在一起的“

8.1. 带宽销售规范

中继节点实现相对简单的行为模式如下：

- 保持一个或多个连接，每个都有自己的加密密钥。
- 检查收到的任何票据和获得的入账。
- 监控交易的平衡，如果超过预先确定的额度，则断开连接。
- 从任何打开的连接接收数据，并在消息边界执行解密。
- 处理解密消息如下:

  —将任何非控制段转发到消息中指定的连接。

  —处理控制字段如下：

```
* 空数据。 指示中继丢弃这个字段。
* 数据转发速率。指示中继以固定速率通过链接发送数据，根据需要对数据包进行排队并生成数据以保持速率。
* 棘轮票据。指示中继将一个票据传递给一个对等节， 该对等节点是原来数据来源点。
* 初始化链接。 指示当建立和断开链接的过程时，中继建立一个新链接。
* 初始化网页链接。（仅限代理）指示代理打开指定主机的 SSL 连接。为了支持白名单，这不能是一个原生 IP 地址。
```

上述行为中的一个重要考虑是需要中继节点持续不断地的工作证明。当与我们所有的 WebRTC 链接结合起来时，网站就可能通过运行纯粹的 JavaScript 中继代码来获利。

有关通过控制字段扩展应用程序的讨论，请参见第 15 节。

8.2. 节点保护和“带宽燃烧“

客户连接的中继有一个非常重要的信息：客户的 IP 地址。 我们假设客户希望尽可能保持私密性，所以默认的客户端会长时间作为第一跳对等节点。

另一个关于第一跳节点的问题，我们在讨论由共谋（9.2 节）引起的信息攻击方面进行了深入的讨论，他们坐在一个理想的位置来执行定时攻击。

为了防止这些攻击，我们建议有隐私意识的用户采用一种称为“带宽燃烧”的方法，向第二跳节点支付金额来获得固定数量的带宽。由于这种方法使用的数据会和网络使用的数据完全不相关，从而可以防止攻击者无法看到中继 3 的入站流量。

为了给寻求逃避的用户提供帮助（第 12 节），带宽燃烧也将支持由流行的非兰花 WebRTC 协议的统计特性确定的非固定速率。

8.3. 链路

对使用中继进行匿名访问感兴趣的客户将使用上述规范来创建中继的“链路”。

9. 链路串谋攻击

在本节中，我们将探讨攻击者可以通过控制或监视多个中继和/或互联网服务提供商（ISP）来推断或推导哪些类型的信息。假定中继和代理是随机选择（ISP 也同样做此假定），我们建立了一个给定攻击可能以不同链长进行的概率模型。

9.1. 个人中继和代理有用的信息

由于基于 IP 的网络固有结构以及兰花协议使用基于以太坊的支付，中继和代理节点及其入侵防御系统可以访问以下信息

- 他们所连接的所有计算机的 IP 地址。
- 他们转发的数据包的大小，时间和数量。
- 控制支付代币的公钥。
- 针对它们的数据包控制字段的内容。

  此外，代理节点及入侵防御系统可以访问以下信息：
- Web 服务器的主机名和 SSL / TLS 会话协商的明文部分。

9.2. 潜在的合谋者

在兰花网络中， 以下的角色可能被攻击者监视或者串谋：

- 互联网服务提供商以客户，中继，代理，web 服务器的角色加入兰花网络。不可靠概率用 s 表示。
- 网站。 链接到代理的网站服务器。 不可靠概率用 w 表示。
- Relayn。 链路中的第 n 个中继节点。 不可靠概率用 r/n 表示。

- 代理。 有流量通过中继到网站服务器的代理。不可靠概率用 x/n 表示。

我们已经分离出上面的 r 和 x，原因是：虽然攻击者无法控制他们可用于计算工作量计算的总计算量，但他们可以控制如何在中继和代理节点之间分配计算。

9.3. 攻击类型

共谋攻击的核心目标是将特定的兰花客户与特定的 SSL 连接相链接。 主要可以采用如下方法：

- 关系。攻击者可以通过观察消息路由过程中的节点来推断出一个客户正在和一个给定的网站通话。
- 记时。 攻击者可以通过控制并观察数据包的时间规律来推断客户正在与给定网站通话。
- 未燃烧的带宽花费。 尽管客户可以使用带宽燃烧的方式掩盖自己的流量，但攻击者仍然可以执行定时攻击。

9.4. “常规”互联网访问：零中继，零代理

虽然当客户直接连接网站时，兰花系统不被使用，但我们在接下来的类容会分析在客户启动中存在的风险。

ISP	Website	P(Relate)	P(Timing)	P(Unburn)
x		s		
	x	w		

在上表中，“X”表示参与共谋，互联网服务商 有 x 的概率采用关系攻击类型。网站有 w 的概率采用关系攻击类型。

9.5. VPN: 零中继， 零代理

为了进行分析，我们也提出了 VPN 访问固有的合谋风险

g 是 VPN 提供商被监控或者与对手勾结的可能性。 请注意，g 可能会随着时间的推移而变化导致难以建模，例如，由于您的 VPN 使用情况。

9.6. 零中继，一个代理

毫无疑问，这种情况下的风险与 VPN 使用的风险很相似。 不使用中继的链相当于在每个浏览会话之前随机选择新的 VPN 提供商的 VPN，并且 VPN 提供商不存储个人信息。

9.7. 一个中继，一个代理

如果在这种配置中使用带宽燃烧，所有的定时攻击都会被缓解。

15. 未来的工作

本节中的内容分为两类：好的方面和我们像公众展示的内部矛盾特征。不够我们相信这种矛盾都是很普遍的——虽然几乎所有人都有一个自己最喜欢的将权利用作歧途的例子，但是同样也有无数的将权利用作正途的例子。兰花这种协议是没有自己的判断力，无法告知它被自由战士，还是被恐怖分子，坏人还是英雄所用。

15.1. 空间证明

如第 5 节所述，我们非常有兴趣探索其他证明类型。这是一个重要的问题，因为工作证明系统的环境影响，以及我们目前的工作证明算法要求全面的计算机充当网络路由器。

我们很高兴能够探索使用磁盘空间成为我们安全核心的稀缺资源的可能性，这可能会让旧手机或类似硬件有利地参与到兰花网络中。

15.2. 内容主机的保护

许多先前的方法（第 3 节）发现内容主机寻求与网络用户类似的保护。我们在这方面内部是有矛盾的，因为我们确实认为有一些内容不符合公共利益（例如有关制造核武器的信息）。但是，如果情况不符合要求，兰花可以扩展到支持这种“无限制的，未受到威胁的主机”，如下图所示：

15.3. 以太坊支付模块的安全保证

正如我们在防火墙规避部分（第 12 节）中所讨论的，客户端的以太坊网络流量很可能是安全的薄弱环节。因为所有的节点都必须维护这个信息，所以使用兰花协议来

分发以太坊信息似乎是天作之合。

不幸的是，依靠那些你正在付钱的信息会导致棘手的问题。我们希望在不久的将来添加处理这个问题的功能，但不会在我们最初发布的版本加入此功能。

15.4. 兰花平台

尽管我们预计核心系统的设计将在不久的将来占用我们大部分的时间，我们也会添加以下用例的相关的功能， 因为这样做， 可以为兰花网络增加大量的流量。

- 1.直接访问网络的 API 接口加入代币服务。
- 2.网络上的文件存储和静态网站托管。
- 3.文件共享。
- 4.电子邮件/消息服务。
- 5.仲裁/调解服务。