オーキッド: 分散型ネットワーク形成と確率的マイクロペイメントの実現

デイビッド・L・サラモン、グスタフ・シモンソン、ジェイ・フリーマン、ブライアン・J・フォックス、ブライアン・ヴォハスカ、スティーブン・F・ベル、スティーブン・ウォーターハウス博士

2018 年 5 月 28 日
バージョン 0.9.2

要約

ブラウジング検閲やプライベートなブラウジング情報の検索が行われるようになるにつれて、匿名化の方法への関心が高まっています。残念ながら、I2P や Tor など、制限がなく監視されないインターネット アクセスに対する既存のアプローチは広く採用されていません。実際、わずか数千人の無給のボランディアだけがリレーと出口ノードをホストしている現状では、巧妙な攻撃者はある程度の数のノードを監視し、さらには侵害することができます。私たちは、「帯域幅マイニング」に基づく市場ベースの完全分散型の匿名ピアツーピア システムを提唱します。このシステムでは、参加者に直接インセンティブを与えることで、リレーと出口ノードの欠如に対処できると考えています。

このペーパーにはまだ開発中のシステムに関する説明が記載されています。そのため、実装において生じる違いに対処するために、将来的に内容を変更したり、新しい内容を追加したりすることがあります。ライブラリ コンポーネントと特定の暗号化アルゴリズムの使用については柔軟に対処できます。ただし、本システムの本質、目的、目標が変わることはありません。

以下のような提案事項があります。

- パケットと類似したトランザクション コストを伴うブロックチェーン ベースの確率的支払いメカニズム
- 帯域幅販売のための商品仕様
- エクリプス攻撃を恣意的に困難にする、ピアツーピア システムにおける分散帰納的証明の方法
- 攻撃者が攻撃の一環として自身の入札単価を変更する可能性がある状況下での帯域幅販売に適した、セキュリティ強化された効率的な入札メカニズム
- 完全分散型の匿名帯域幅市場

目次

1. イントロダクション

オーキッド プロトコルは、帯域幅販売者をオーキッド マーケットと呼ばれる構造化されたピアツーピア (P2P) ネットワークに編成します。カスタマーはオーキッド マーケットに接続し、インターネット上の特定のリソースに*プロキシ チェーン*を形成するために帯域幅の売り手に支払いを行います。

グローバル インターネットからデータを送受信する一般的な方法とは異なり、オーキッド マーケットのプロキシ チェーンはデータの発信元に関する情報とその宛先に関する情報を分離するため、単一のリレーまたはプロキシで両方の情報を保持したり、保持する人の身元が判明したりすることはありません。オーキッド マーケットの構造は、*共謀攻撃* (帯域幅の複数の売り手が情報の分離を克服する能力) に対する強力な抵抗手段を提供することで、情報の分離をさらにサポートします。

プロキシ チェーンの参加者の役割は次のとおりです。

- ***ソース ノード*または*カスタマー*** — トランザクションを開始する参加者。
- ***リレー ノード*** — ネットワーク トラフィックを転送する中間参加者。
- ***プロキシ*または*出口ノード*** — 要求されたグローバル インターネット サイトに接続する参加者。
- **帯域幅の売り手** — *リレー*または *プロキシ*。

発信元と宛先の情報を区分してグローバル インターネットからデータを送受信するあまり一般的ではない方法とは異なり、オーキッド マーケットはトラフィック分析を防止する*固定レート中継*と、情報の隠蔽または発見に関連しない参加を促すインセンティブを提供します。それがトークンによる支払いです。

詳細な説明に移る前に、本システムが解決する主要な問題と、本システムの基盤として私たちが選択した一般的なソリューションについて簡単に説明します。

*トラフィック分析*問題

問題: あなたが数学者でいっぱいのカフェテリアにいると想像してください。あなたは部屋の中にいる誰にも知られずに、部屋の向こう側にいる友人にメッセージを送りたいと考えています。メッセージを渡す手順についてはまだ交渉していないため、実装のすべての詳細を部屋にいる全員に公開する必要があります。何ができるでしょうか?

1981 年に Chaum [56] によって提示されたこの問題に対する見事な解決法は、すべての人をリレーと受信者の両方として行動させることでした。このスキームでは、参加者はデジタル的に「封筒が入った封筒」に相当する暗号化されたメッセージを準備します。アリス (Alice) にメッセージを送信するには、次のように計算します

$$Enc(\text{“}ToBob\text{”} \,||\, Enc(\text{“}ToAlice\text{”} \,||\, Enc(message, Alice), Bob), Carol)$$

そのメッセージ (message) をキャロル (Carol) に送信し、キャロルはそれを解読してボブ (Bob) に送信し、ボブはそれを解読してアリス (Alice) に送信します。トラフィック分析を防止するために、全員が一定数のメッセージをサイクルごとに送信します。返信アドレスの処理には、ボブとキャロルに一意のメッセージ識別子を記憶させ、そのチェーンに沿ってメッセージを送り返せるようにします。

上記の方法を使用するシステムにおいて特に重要視する必要があるのは、*共謀*の可能性です。ボブとキャロルが協力すれば、所与のメッセージの送信者と受信者を特定できる可能性があります。

*シビル*問題

上記のカフェテリアの問題では、物理的な身体を使用して*シビル攻撃* (1 人の参加者が恣意的に多数のユーザーのふりをする可能性がある状況) を防ぎました。残念ながら、デジタル システムではこのアプローチは使用できません。

問題: デジタルのコンテキストにおいて、誰が「本物」であるかはどうすればわかるでしょうか?

この問題の解決法は Hashcash [85] にあります。「本物」であることを証明するために計算リソースを消費しなければならない環境であれば、膨大な数のネットワーク参加者を装うためには膨大な計算リソースを占有しなければならない、という状況にシビル攻撃者を追い込むことができます。

*無作為抽出*問題

上記のカフェテリアの問題は、システムの他のすべてのユーザーにメッセージを送信する簡単な方法を想定しています (たとえば、カフェテリアで叫ぶなど)。*共謀攻撃*に対して最高度の耐性を持つ Chaumian mix (チャウミアン ミックス) を実装するには、「本物」のリレーから無作為に抽出できる必要があります。簡単に言うと、そのためには誰かがネットワークに参加したりネットワークから離脱したりするたびに全員が通知を受ける必要があります。残念ながら、実世界の P2P ネットワークでは、すべてのユーザーにそのようなリストを維持させると許容量を超えるネットワーク トラフィックが発生します ($O(n^2)$ 件の通知)。

問題: ネットワークのオーバーヘッドを最小化し、ピアの効率的な無作為抽出をサポートする、現在「本物」のすべてのリレーの分散リストはどのようにしたら維持できるでしょうか?

この問題に対する優れた解決法は Chord [85] の分散ハッシュ テーブル (DHT) にあります。このスキームでは、各ピアに大きなスペースで一意のアドレスを割り当てられ、その後 $O(log(n))$ 時間内に検索を実行できる方法で接続されます。ユーザーの追加または削除に必要なのは $O(log(n))$ のピアに通知することのみです。

システムの概要

オーキッド プロトコルは、本質的に上記の解決法の組み合わせです。私たちのアプローチでは、ピアは自らが「実在」することを実証するために、*メダリオン*を作成するよう要求され、その後、*オーキッド マーケット*と呼ばれる分散型 P2P ネットワークに編成されます。オーキッド マーケット参加者の誠実性を保つために、すべてのピアはその隣人の行動の正当性をチェックします。その後、カスタマーはオーキッド マーケットを使用して、チャウミアン メッセージ転送用の無作為のピアを選択します。オーキッド マーケットでは、参加を奨励するために、カスタマーが転送バイトごとにリレーとプロキシに支払いを行うようにします。

これは単純なアイデアですが、もちろん細部には予想外の面倒もあります。システムは完全に分散化され、完全に自律的で、完全に匿名であり、支払いを処理できる必要があります。したがって、この設計文書の多くは、カスタマーのセキュリティ、システムのパフォーマンス、およびシステムの経済的な健全性に対する攻撃を防ぐことに重点を置いています。攻撃分析は重要であり、私たちは多くの時間をそれに費やしていますが、結局のところ市場の運営に対する必要な自負心にすぎません。「森の中で迷子になった」ことに気付いたら、先ほどの説明を道しるべとして使っていただければ幸いです。システム設計の詳細はすべて、上記の 3 つの問題に対する現実世界のソリューションを実現するためのものです。

2. 代替アプローチ

保護されていないインターネット アクセス

保護なしでインターネットにアクセスしているユーザーは、完全な閲覧履歴と Web サイトの使用履歴を ISP に提供していることになり、さらに ISP はそのデータを共有したり販売したりしているかもしれません。

仮想プライベート ネットワーク (VPN) サービス

仮想プライベート ネットワーク (VPN) では、暗号化を使用して、VPN サブスクライバーのトラフィックをより大規模でセキュアではないネットワークに安全に転送します。VPN がトラフィックを受信するとトラフィックは復号化され、別の大規模なセキュアではないネットワークに再送信されます。再送信はユーザーが Web サイトによって課せられたアクセス制限を回避するのに役立つだけでなく、少しではあるものの Web サイトのブラウジング習慣の追跡を減らすこともできます。また、暗号化はユーザーの ISP がトラフィックを見ることを防ぎ、それにより監視攻撃を防止できます。これは、VPN をユーザーの新しい ISP にすることで可能になります。ISP が以前に実行できた攻撃は、いずれも VPN プロバイダーによって簡単に実行できます。

VPN ユーザーは、VPN プロバイダーが常に信頼できるとは想定すべきではありません。VPN サービス プロバイダーは ISP よりも多くの競争に直面していますが、最終的には同じ人材源から優れた才能を募り、同様の帯域幅販売ビジネス モデルを採用するでしょう。VPN プロバイダーが、ユーザーが ISP を信頼しないようになった原因であるインセンティブの餌食にならないことはまずありません。さらに、VPN 環境でトラフィックを中継するために IP アドレスを再利用することにより、商用 Web サイトによるそれらの使用を比較的簡単にブロックできます [13]。

Tor (トーア)

Tor [60] は、*オニオン ルーティング*のアイデアをより多くの聴衆に紹介することで有名なフリー ソフトウェア プロジェクトです。このシステムでは、ユーザーはリレーと出口ノードのグローバル リストをダウンロードしてそのリストから無作為に選択を行い、選択したものから*オニオン ルート*を形成します。オニオン ルートは順番を付けたリレーのリストです。オニオン ルートに沿って送信されるパケットは各ピアに対して順番に暗号化されるため、出口ノードが確実にパケットを理解するためには、各ノードがルートの途中でパケットを受信する必要があります。その結果、複数のノードが侵害されるか、同じユーザーが複数のノードを実行することがない限り、誰がパケットを送信したのか、そしてパケットがどこに行ったのかの両方を知るリレーは 1 つのみとなります。

3. 攻撃

オーキッド プロトコルの多くは攻撃防止を中心に設計されているため、ここではまず、P2P ネットワークに対する一般的な攻撃に関する文献を確認していきます。

推論攻撃

オーキッド プロトコルが最も重点を置いて防御しなければならないのはユーザーに関する情報を流出させる攻撃です。オーキッドは既存のインターネット上のオーバーレイとして実装されているため、一部の情報は*不可避的に*一部のピアと共有されます。さらに、オーキッドの基盤となる支払いシステムは ERC20 トークンを利用しているため、いくつかのトランザクション情報も同様に、イーサリアム ネットワークで入手可能となる場合があります。以下のリストでは、そのような情報に「*」のマークが付いています。このドキュメントで*不可避的に*共有されていると具体的に記載されていないものの、その情報を明らかにする方法が発見されている情報は*情報攻撃*と呼ばれ、オーキッドの White Hat Bug Bounty で説明されています。共有されるものの詳細については、セクション 7 のプロトコル仕様、セクション B.1 の共謀の説明、およびネットワークのリファレンス実装 [1] を参照してください。

攻撃者が興味を持っていると想定されるデータのタイプ (タイムレス)。

- 現実世界の個人識別情報。ユーザーの名前、SSN、住所など。
- Web サイトのアカウント情報。特定の Web サイトのユーザー アカウント。これは、現実世界の個人識別情報とは異なる場合があることに注意してください。
- *IP 情報。ユーザーがオーキッド ネットワークにアクセスしている IP アドレス。これは、場合によっては現実世界の個人識別情報の流出に等しいことに注意してください。
- *イーサリアム情報。ユーザーのウォレットに関連付けられた鍵 (*公開鍵または秘密鍵)。これは、場合によっては現実世界の個人識別情報の流出に等しいことに注意してください。

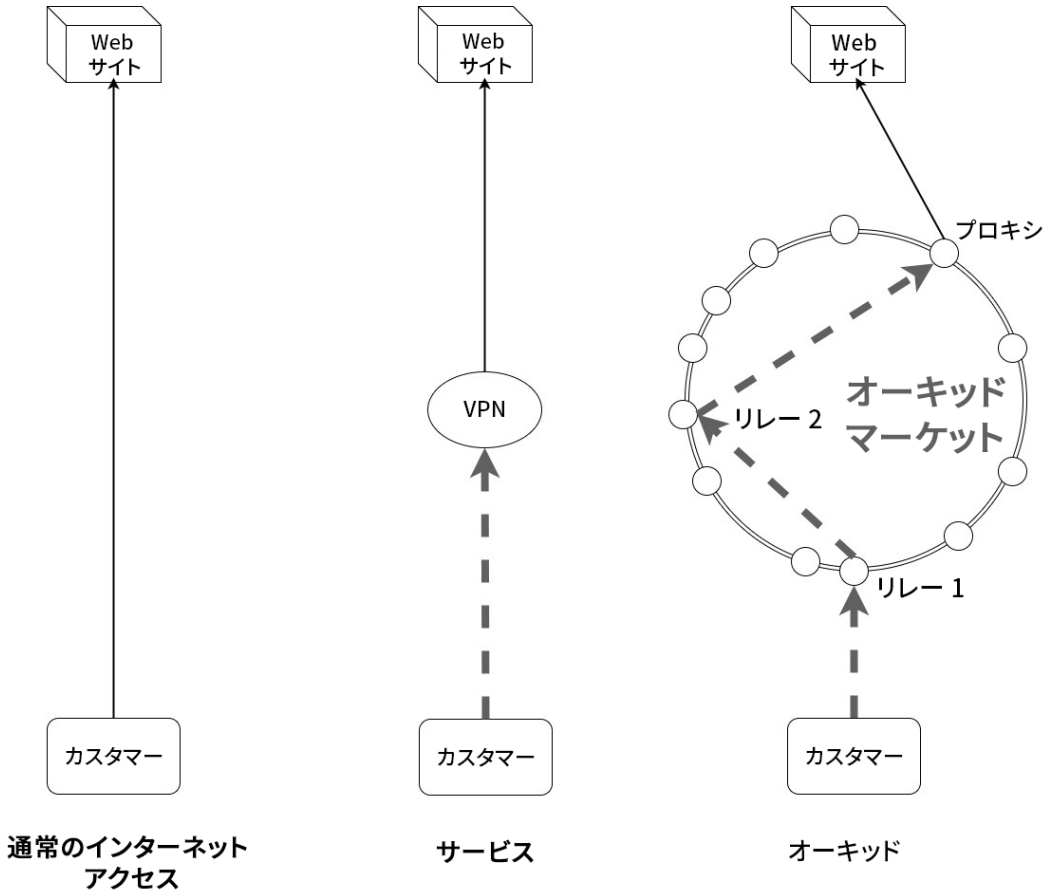

図 1: 直接接続、VPN 接続、オーキッド接続

- *オーキッド ネットワーク情報。オーキッド ネットワーク上のノードの現在のビジネスに関連付けられている鍵 (*公開鍵または秘密鍵)。

攻撃者が関心を持っていると想定される、以下のような種類の挙動情報 (時間およびチェーン関連データ)

- *カスタマーの識別情報。攻撃者はカスタマーの IP アドレスを取得します。
- *リレー識別情報。攻撃者はリレーの IP アドレスを取得します。
- *プロキシ識別情報。攻撃者はプロキシの IP アドレスを取得します。
- *リンク識別情報。攻撃者は、1 つのチェーンで 2 つの IP アドレスが使用されたことを知ります。
- *Web サイトへのアクセス。攻撃者は、オーキッド ネットワークから特定の Web サイトへのアウトバウンド接続が行われたことを知ります。
- *Web サーバー アクセス。攻撃者は、オーキッド ネットワークから特定の Web サーバー (複数の Web サイトをホストしている可能性あり) へのアウトバウンド接続が行われたことを知ります。
- *イーサリアム リンク。攻撃者は、イーサリアムの公開鍵がオーキッド ユーザーによって保持されていることを知ります。
- *購入リンク。攻撃者は、2 つのトランザクションの支払者が同じであることを知ります。
- *購入情報。攻撃者は、チェーンを介して送信される帯域幅の量とタイミングを知ります。

以下で説明するように、上記のすべての挙動情報は通常の運用中にオーキッド ネットワーク上の他のノードと共有されていますが、ほとんどの状況では、ユーザーに直接危害が及ぶのは、行動情報集約によって攻撃者が一度に複数の情報を取得できる場合です。たとえば、ユーザー X が Web サイト Y にアクセスしたことを確認するためには、攻撃者は、購入者の識別情報、Web サイトへのアクセス情報、いくつかのリンク識別情報が必要です。このため、参照仕様に準拠しているピアであれば、カスタマーが購入したサービスを提供するために必要な場合を除き、上記の情報を保存または共有することはありません。

システム動作の統計的モデリングによる匿名化解除は、推論攻撃または監視攻撃と呼ばれます。これらはしばしば、慎重に仕組まれたリクエストまたは時間制限されたリクエストなどの「様子見」や、ネットワークから特定のピアを DoS 攻撃し、トラフィック パターンがどのように応答するかを観察するなど、他の攻撃と組み合わせて実行されます。

- SSL で暗号化された Web トラフィックからエンドユーザーの病歴、家族の収入、投資状況を推測 [57]。
- グローバル トラフィック ログから Tor、I2P、およびオーキッド トラフィックの匿名化を解除 [59]。
- タイミング分析による OpenSSL サーバーの秘密鍵の取得 [54]。

経済的攻撃

同様のシステムとは異なり、オーキッド プロトコルは支払いメカニズムに対する攻撃にも関与する必要があります。本ペーパーで使用される分類法は次のとおりです。

1. **経済的悪用。**ユーザーが「無料サンプル」帯域幅を提供して、無料サンプル帯域幅のみの使用を可能にするなど、利益をもたらす望ましくない行動。
2. **経済的サービス妨害 (EDoS 攻撃)。**支払いを利用してオーキッド ネットワークの別のノードを大量の購入で圧倒し、これらのノードをオフラインにする。

シビル攻撃

複数のユーザーになりすまして実行される悪意のあるアクションは、複数の人格障害に苦しむ患者の症状に似ていることから、その名前を取り「シビル攻撃」と呼ばれます。このタイプの攻撃の利用例には以下のようなものがあります。

- Yelp、Amazon などに複数のレビューを送信する。
- 複数のリーチャーになりすますことにより、BitTorrent で高速ダウンロードを行う [74]。

エクリプス攻撃

エクリプス攻撃の攻撃者の目標は、システムの一部をシステム自体から隠すことです。使用される方法は一般に、ネットワークでの権限昇格攻撃に相当するものです。これは、ネットワーク上でネットワークに対するより多くの制御を獲得し、それを利用してさらに多くの制御を得ようとするものです。

- ビットコイン マイニング P2P ネットワークをセグメント化し、攻撃者が計算能力の実質的に 51% 未満しか制御していない場合でも、いわゆる「51% 攻撃」を可能にする [65]。
- マグネット リンクに関連付けられたアドレス スペースを乗っ取ることにより、BitTorrent DHT からファイルを削除する [87]。

中間者攻撃

相互作用する 2 当事者の間に自身を割り込ませた場合にのみ実行できるアクションを総称して中間者攻撃と呼びます。暗号化された情報はメタデータの分析のために記録される場合がありますが（セクション 3)、暗号化されていないデータはさらに変更されて動作を支配する可能性があります。鍵の交換がセキュリティで保護されていない場合、中間者は、攻撃者の鍵が相手の鍵であると誤って信じるように、双方の当事者をだますこともできます。

サービス品質攻撃

一部の敵対者は、オーキッド ネットワーク ユーザーのシステム パフォーマンスを全般的に低下させ、それにより潜在的に使用を減らすことで満足する場合があります。

サービス拒否攻撃

特定のリソースをオフラインにすることを中心とした攻撃は、DoS（サービスの可用性侵害）攻撃と呼ばれます。「予期しない」状況でのシステムの動作は、多くの場合、十分に特定されておらずテストもされていません。DoS 攻撃は、P2P ネットワークのノード匿名化の解除につながります。以下は、その注目すべき例です。

- Tor トラフィックの匿名化を解除するために、シビル攻撃ベースの監視と連携して使用される標的型 DoS 攻撃 [52]。
- I2P のフラッドフィル データベースを完全に制御するための DoS 攻撃によるオフライン化には 20 のシビル ノードしか必要としないため、ネットワーク上のすべてのトラフィックの匿名化解除が可能 [64]。

ハッキング

歴史的に信頼できるピアを攻撃ベクトルに変換することにより、やる気のある攻撃者がネットワーク上のノードを直接侵害する可能性があります。チェーンを使用して帯域幅を展開すると、最終的に攻撃者が接続を「バックトレース」できるようになる可能性があります。このような攻撃はセキュリティ上重要な意味を持ちますが、オーキッド ネットワークの対象範囲外です。オーキッド ネットワークでその設計どおりの目標を達成した後は、これがシステム ユーザーに対する主要な攻撃・脅威となります。

4. オーキッド マーケット

オーキッド マーケットは、オーキッド プロトコルを構築する基盤となります。オーキッド マーケットは基本的には、リレー、プロキシ、およびユーザー間の帯域幅の売買を促進する分散型 P2P ネットワークです。メダリオンと呼ばれるプルーフオブワークを提示することでこのマーケットに参入し、参加を継続することができます。オーキッド マーケットのネットワーク構造は分散型ハッシュ テーブル (DHT) に似ており、修正されたコード [83、85] の拡張と考えることができます。

4.1. 基本的なマーケット オペレーション

大まかに言うと、オーキッド マーケットが提供するオペレーションは次のとおりです。

- ペドラーがオーキッド マーケットに参加する方法。
- 販売しているサービスをペドラーに尋ねる方法。
- 計算リソースによって無作為に加重された、すべてのピアのサブセットを選択する方法。*ルックアップ プロパティ*が保持するリソースなど。

$$lookup(random\_address) \Rightarrow random(Peddler)$$

ペドラーはコードのノードと見なすことができ、フィンガー テーブル アナログ (署名付きルーティング表と呼ばれる) を介して、近隣の情報を追跡します。オーキッド マーケットでは、ペドラーはリレーまたはプロキシの場合もあり、帯域幅の買い手および売り手として機能します。ユーザーはオーキッド マーケット内でペドラーとなる必要はありませんが、すべてのリレーとプロキシはペドラーになる必要があります。*ルックアップ プロパティ*は重要です。なぜなら、一連の n ペドラー (a アタッシェを含む) から無作為にペドラーが抽出された場合、次の確率で無作為のペドラーが攻撃者ではないことをカスタマーが知ることができるためです。

$$P(Attacker|random(Peddler)) = 1 - \frac{a}{n}$$

オーキッド ホワイトペーパー [90] の各セクションでは、このプロパティがエクリプス攻撃やその他の攻撃に対する保護となることを示しています。これらのオペレーションを実装するために、オーキッド マーケットは鍵と値のない DHT の構造を採用しています。無作為抽出を実行するには、ユーザーは単に無作為なアドレスを生成し、そのポイントに最も近いペドラーを見つけるだけです。オーキッド マーケットは、オーダー 2^{256} のコードのようなリングとして表すことができるため、任意の無作為なアドレスは、$\{1, 0\}^{256}$ から無作為な整数を選択する必要があります。

4.2. ペドラーの基本的なオペレーション

オーキッド マーケットでペドラーがサポートするオペレーションは次のとおりです。

- *リスト サービス*。ペドラーに対して、ペドラーが販売するサービスのリストを要求します。
- *ルーティング表とメダリオンの入手*。ペドラーのメダリオン、署名されたルーティング表およびルーティング表のメンバーにトラフィックを中継するコストを返します。
- *中継トラフィック*。ルーティング表内のピアの 1 つにトラフィックを転送するために、ペドラーに支払いをします。

- *購入サービス*。ペドラーをサービス プロバイダーとして採用します。

メダリオンはプルーフオブワークのためのオーキッド マーケットのトークンであり、マーケットに*参加するためのライセンス*です。メダリオンを持たないペドラーは、現在のイーサリアム ブロック ダイジェストの TTL のオーダーで、時間内にマーケットから削除されます。署名済みのルーティング表については、オーキッド ホワイトペーパー [90] でさらに説明します。上記のオペレーションの最初の 2 つはカスタマーが目的のペドラーに移動するために使用され、次の 2 つは、ペドラーが見つかった場合にサービスの購入を交渉するために使用されます。

オーキッド マーケットのナビゲーションはチェーンで使用されるものと似ています。カスタマーは既知のペドラー (ブートストラップで発見、4.6 を参照) に接続してそのルーティング表を調べ、選択したポイントに最も近いペドラーにトラフィックを転送するために支払いをします。ルーティング表のセクションで説明するように、これによりカスタマーは IP アドレスを秘密に保ちながら、$O(log^2 n)$ パケットのペドラーに対して、比較的効率的な無作為のアクセスを提供できます。

帯域幅を必要とするオーキッド マーケットでのすべてのオペレーションには、他のエンティティと同じ帯域幅コストがかかることに注意してください。各ペドラーは、少なくとも帯域幅を購入するのと同程度に頻繁にマーケット オペレーションのために帯域幅を販売するため、各ペドラーのこれらのコストは最小化されます。ペドラーに対するこの帯域幅コストは、付録 E で言及されている攻撃を防止します。

ペドラーは、修正された Chord DHT で使用されるのと同じスキームを使用して、オーキッド プロトコルで接続されます。より成熟した文献と、マシンでチェックされた正確性の証明が存在するため、私たちは、カデムリア (Kademlia) よりもコード (Chord) を選択しました [83]。

一連のペドラー アドレスは、サイズ 2^{256} のコードリングにおける整数として表されます。ここで距離 d (ピアのアドレス a と b の間の距離) は次のように定義されます。

$$a,\ b,\ d \in (0,\ 2^{256})$$

$$a + d \equiv b \pmod{2^{256}}$$

A はオーキッド マーケットにおけるペドラーの集合、e は単一の特定のペドラーとします。コードでは、任意のノードのピアの最大予想数は $\log_2(n)$ であることを思い出してください。e の一連の強制接続は次のように定義されます。

$$L = \{f : \min_{log_2(n)}\{dist(e + t, f)\}\}$$

ここで、$t \in \{1,\ 2,\ 4,\ ..2^{255}\}$ は任意のペドラーで、min は、dist(...) から最小の $\log_2(n)$ 要素のセット (f に対しては最小の距離) を返します。

私たちがこのルーティング構造を選択したのは、その成熟度、展開されたシステムでの成功した実績、および正当性の証明のためです。詳細について興味をお持ちであれば、[85] をお読みください。ここでは、このルーティング スキームによって次の 2 つの特性が提供されることに注意するだけで十分です。

1. *有限の確定的な接続*。すべてのペドラーは ≤ 256 の強制接続を有することが期待されます。
2. *対数横断距離*。無作為なアドレス t、無作為に接続されたペドラー e (接続 C を保持)、$dist(e,t) \approx 2 * \min_{f \in C} dist(f,\ t)$ を想定します。ホップごとに距離が半分になるため、ネットワーク上の予想されるトラバースの長さは $log_2(n)$ となります。ここで n はネットワークのサイズです。

4.3. オーキッド マーケットのメダリオン

メダリオンはプルーフオブワークのトークンであり、イーサリアム ブロック ダイジェスト、所有者の公開

鍵、および付録 D で説明されているその他の数量と密接に関連しています。オーキッド マーケット内では、メダリオンは 2 つの方法で使用されます。

- 攻撃の原因となる市場への些細な侵入を防ぐため
- 攻撃者が市場で場所を選択するのを防ぐため

攻撃者がオーキッド マーケットの合計計算能力のシェアに比例するよりも多くのペドラーを実行するのを防ぐために、すべてのペドラーは、メダリオン サイクルごとに自身のメダリオンへのすべての接続の有効性をチェックします。有効なメダリオンが提供されない場合はネットワークから切断されます。ペドラーの場所は、付録 C で定義されているメダリオンの暗号化ハッシュになるように定義されています。それはすなわち、以下のように言い換えることができます。

$$\text{ペドラーのアドレス} = H(\text{メダリオン}, \ldots)$$

これにより、オーキッド マーケットの各メンバーは、メダリオン所有者のマーケットでの位置を簡単に評価および検証できます。さらに、ペドラーのマーケット アドレスをメダリオンに紐付けることにより、エクリプス攻撃を実行することがはるかに困難になります。

4.4. 署名付きルーティングとエクリプス攻撃

分散型ネットワークで発生する問題の 1 つは、誰も（おそらく攻撃者を除く）がネットワークをグローバルに把握していないため、ペドラーが、悪意のある完全に隔離されたサブネットワークに接続される*エクリプス*攻撃を受けているかどうかの判断が難しいことです。たとえば、上記のルーティング スキームで、有する接続について攻撃者が嘘をついた場合を想像してください。買い手がこれを検出する方法がない場合、すべての「参加者」が攻撃者によって所有されている偽のオーキッド マーケットに誘導される可能性があります。この状況の悪用を緩和するために、ペドラーのルーティング表はアルゴリズム的に選択され、ルーティング表に含まれるピアによって検証されます。

ノードが強制接続を確立しようとする場合、そのノードは、強制接続リスト上の各ノードに対して、そのリスト上の他のノードがすべて同じオーキッド マーケットのメンバーであることを証明する必要があります。これを行うには、まず、無作為なペドラー G を選択します。これは、ルーティング表 $H(C_i)$ 内のすべての接続のハッシュに最も近いアドレスを持つペドラーを見つけることによって行います。その後、以下を提供します。

1. リスト上のすべてのペドラーが G にルーティングできることの証明。
2. G が各ペドラーにルーティングできることの証明。
3. リスト上の各ペドラーが実際に強制接続であることの証明。

これらの証明はすべて、C_i から G に至る署名されたルーティング表の形態をとります。または (3) の場合には、エントリ ペドラーから各 C_i に至る署名されたルーティング表のチェーンの形態をとります。そのような証明が提供されると、新しいルーティング表のすべてのピアがテーブルに署名し、接続するペドラーがそれらに署名します。新しいペドラーが強制接続される C_i のこれらの要素については、署名取得のためにそれぞれの接続に同じプルーフが送信されます。

これがオーキッド マーケットにペドラーを追加する唯一の方法であるため、これらの要件はオーキッド マーケットの健全性の帰納的証明となります。C_i のノードの 1 つが偽のルーティング表を提供しようとした場合、G へのルーティングは C_i の他のペドラーと同じになりません。C_i のノードのいずれかがオーキッド マーケットのメンバーではない場合、G はそれらにルーティングすることはできません。接続しようとするペドラーが C_i が自身の強制接続ポイントに最も近いノードであると嘘をついた場合、(3) はそれが偽であることを示します。

これらの特性から、攻撃者に残される道は以下のとおりになります。

- 攻撃者がすべての C_i を制御できるようなメダリオンのアドレスを生成できた場合には、上記のシステムは機能しなくなります。これが発生する確率は $\left(\frac{a}{n}\right)^{\log(n)}$ です。そのようなコリジョンが発生した場合、すべてのクエリの $(1 - \frac{n-\log(n)}{n})^{\log(n)}\%$ が損なわれます。これらの数値を概観すると、攻撃者がネットワークの 10% を制御している場合、100 万ノードでは、そのようなコリジョンが発生する確率は $1 \times 10^{-8}\%$ です。そしてそれが発生した場合、すべてのシステム クエリの $1 \times 10^{3}\%$ が影響を受けます。1 億ノードでは、確率は 1×10^{-12}% に低下し、クエリの 1e-5% に中断を引き起こします。この損傷は再生成中に修復されます (セクション 4.5 を参照)。
- 攻撃者がネットワークに参加したものの、有効なルーティング表の使用を余儀なくされた場合、実行できる攻撃は、オーキッド マーケットでのトラフィックのルーティングではなく、サービスの販売に関連するもののみとなります。これは当社のその他の攻撃モデルで予想される状況であるため (攻撃者は計算リソースに比例して多くのペドラーを制御する)、ここでは攻撃とみなしません。

4.5. エクリプス攻撃と再生成

寿命が長い P2P ネットワークはエクリプス攻撃の標的となります。上記の署名付きルーティング スキームは検証のためのピアの数を増やし続けることでこれらを困難にすることができますが、もう一つのアプローチは単純にピアの寿命を制限することです。このため、オーキッド マーケットのペドラーはイーサリアム ブロック 100 個ごとに鍵を変更する必要があります。

4.6. エントリ ノードを探す

エントリ ノードの配布は難しいトピックです。抑圧的な政府がこのリストにアクセスできた場合は、ユーザーがこのリストにアクセスする能力をブロックすることでしょう。そのため、ブロックされた場合にはインターネット中断を引き起こす重要なサービスを見つけ、それらに含まれるデータにエントリ ノード情報を追加する方法を考案しました。

4.7. そのオーキッド マーケットの識別

セキュリティに関する上記の説明は、新しいマシンで「正しいオーキッド マーケット」を見つける方法がなければ最終的には無意味となります。*エントリー ペドラー*のための配布方法はいずれも、攻撃者によって操作されたエントリー ペドラーの潜入を免れるものとみなすことはできません。そのために、私たちは所与のオーキッド マーケットの計算力を推定し、最も総合的な計算力を持つ市場を選択しています。

- 密度の推定。ペドラーの強制接続は、2^{256} のアドレス スペースの一連のポイントに最も近いペドラーと定義されているため、実際の状況では、理想的な接続と実際の接続の間に測定可能なギャップがあります。このスペースの密度を推定するために、これらの接続を無作為な二項プロセスの結果として見ることができます。すなわち、理想のポイントと実際のポイントの間のすべてのポイントは失敗であり、実際のポイントが成功です。したがって、所与の数の欠落ノード M と、実現された接続の所与の数 C について、事前の統一的な MAP 推定によるネットワーク密度は以下のようになります。

$$\frac{C}{C+M} * 2^{256}$$

- 横断距離。オーキッド マーケットでは $O(log_2(n))$ のホップにおいて、アドレスのルックアップを提供しています。これを逆に使用すれば、ネットワーク密度を推定することができます。

密度の推定は十分であると考えがちですが、控え目で適度な規模のシビル ネットワークを所有している賢明な攻撃者であれば、偽のネットワークのエントリ ペドラーとして自由に使用できるノードを持つことになります。一方で、「本物のオーキッド マーケット」のペドラーは、ネットワークからの無作為なサンプルである密度を持つことになります。さらに悪いことには、横断距離が測定基準として選択された場合、攻撃者がそれを予測し、$O(log_2(n))$ よりも長い、横断にとって最適ではないルーティング表を作成することが想像できます。ありがたいことに、準最適に接続されたオーキッド マーケットでは密度指標でのパフォーマンスが低下します。オーキッド システムで使用される検証方法は、無作為のアドレスに移動し、その途中でルーティング表を保存して、最初の 2 ホップを除くすべてのルーティング表を使用して密度推定を実行することです。

4.8. プロキシ ホワイトリスト

プロキシ サービスの提供を希望する一部のユーザーは、「オープン アクセス」を提供することに抵抗がある場合があります。たとえば、ユーザーに facebook.com へのアクセスを許可すると、リレーとして機能するのと同様のリスクを負うことになり、インターネットへの任意の接続を許可すると、地元の法執行機関からの要請につながる可能性があります。そのため、オーキッド マーケットのペドラーは、プロキシとして使用するときにユーザーがコンタクトできる Web サイトのホワイトリストを設定し、*Get Offers (オファーを取得)* に対するその応答で、それらのホワイトリストを指定することができます。

5. メダリオン

完全に分散化された完全匿名のデジタル システムは、1 人の悪意のあるユーザーが数千人のユーザーになりすます攻撃 (シビル攻撃) に悩まされます。シビルの生成およびこのクラスの攻撃のその他の影響を軽減するために、オーキッド プロトコルはプルーフオブワークのスキームを採用しています。このスキームのトークンはメダリオンと呼ばれます。各メダリオンには、ジェネレーターが所与の時間で*かなりの規模*の計算リソースを占有したことを暗号で示すデータが含まれています。計算は高価なリソースであるため、メダリオンを使用することで、特定の攻撃者が複数のユーザーになりすまそうとする際にコスト上の制限を課すことができます。

5.1. メダリオンのプルーフオブワーク

メダリオンは、私たちのコア セキュリティの前提とネットワーク全体の間をつなぐ重要な役割を担っています。基本的なセキュリティ目標は、意欲的な攻撃者によるオーキッド ネットワークの制御を阻止することであるため、メダリオンの作成には次の条件を満たす必要があります。

1. 悪意のないノードであればメダリオンを*簡単*に作成できること
2. メダリオンが*簡単に*検証可能であること
3. メダリオンの大量作成は*困難*であること

これらの条件において「*困難*」とは、膨大な時間と資金を要することと定義します。要するに、私たちが求めるのは、通常のノードがネットワークへのエントリを取得するのは*簡単*な一方で、攻撃者がネットワークへのエントリをスケーリングするのは困難なプルーフオブワーク システムです。セクション 5.2 では、プルーフオブステーク [46、70、72] やプルーフオブスペース [63、80] などの他の方法ではなく、プルーフオブワークを選択した理由について説明します。

上記の要件を満たす主要な方法は現在 2 つ存在します。「チャレンジ-レスポンス プロトコル」と「暗号パズル」です。残念ながら、攻撃者は共謀によってチャレンジとレスポンスを事前に計算できるため、チャレンジ-レスポンス プロトコルはオーキッド モデル内では十分なセキュリティにはなり得ません。残された選択肢は暗号パズル [50、78] ですが、暗号パズルは今日数多く存在しており、それぞれ独自のトレードオフがあります。繰り返しますが、オーキッドの要件を満たすために適切なのは、このような各種暗号パズルの一部のみです。つまり、簡単に並列化したり、ASIC にしたり、簡単にスケーリングしたりできない暗号パズルです。最近、研究者は、調整可能な作成難易度を含む、検証の容易な結果を生成するアルゴリズムを発見しました [50]。これらのアルゴリズムのコレクションでは、メモリと全シリコン領域のスケーリングに高コストがかかるという傾向を利用しています [45、61]。これらのクラスのアルゴリズムは非対称メモリハード関数と呼ばれ、私たちはこれをメダリオンの作成に使用しています。これらの関数にはいくつかのバリエーションがあります [50、75、86] が、私たちは Equihash (エクイハッシュ) を使用することにしました。Equihash は k-XOR バースデイ問題に基づくもので、時間と空間のトレードオフを介してメモリ ハードネスを提供します [1]。Equihash は調整可能かつシンプルであり、NP 問題に基づいており、暗号通貨コミュニティで受け入れられているため、このような関数をプルーフオブワークの基盤として使用することで、許容できるレベルのセキュリティと将来の保証が得られると考えられます。

メダリオンを作成するには、ピアは公開鍵 K と以前のイーサリアム ブロック ハッシュ E を取得し、その後、ソルト S を見つけるための一連の計算を実行します。ソルトは、$F(K, E, S, ...) \geq N$ などで、この N は難易度のスケーリング係数です。新しいイーサリアム ブロックがチェーンに追加されると、メダリオンを最新の状態に保つために新しい S を計算する必要があります。メダリオンの仕様については、付録 C でさらに詳しく定義されています。

5.2. プルーフタイプの選択

他の市場ベースの分散型ネットワークに精通している読者には、私たちのメダリオンの使用が前提において他のプルーフオブワーク システム（ビットコインなど）と類似していることがわかるでしょう。さらに、オーキッド プロトコル、具体的にはオーキッド マーケットへの承認を得るために、プルーフオブステーク、プルーフオブアイドル、またはその他の方法を使用しないのはなぜかと疑問に思われるかもしれません。このセクションでは、他の方法ではなくプルーフオブワークを選択した理由を説明します。

プルーフオブステーク

プルーフオブステークは、攻撃者がトークンの大部分を制御することはないという前提に基づいています。私たちが想定する攻撃モデルには、意欲が高く、リソースが豊富で抑圧的な政府も含まれる可能性があるため、プルーフオブステークの前提が常に真であると見なすことはできません。ビットコインの驚くべき時価総額でさえ、控え目な規模の国の GDP をはるかに下回っています。事態をより複雑にしているのは、私たちが近い将来、匿名の支払いをサポートするようにシステムを拡張する予定であることです。これにより、「敵対的買収」の検出がはるかに困難になります。したがって、システム内の十分

[1] [67] によって最初に発見されたように、この時間と空間のトレードオフが時間とメモリのトレードオフを連想させるのも偶然ではありません。

なステークによって、オーキッド プロトコルの匿名性とセキュリティが恒久的かつ不可逆的に侵害される可能性があるため、メダリオンのベースをプルーフオブステーク モデルとすることはできませんでした。要するに、プルーフオブステークを使用しなかったのは、ユーザーのプライバシー権が最高額の入札者に販売される可能性のあるシステムを設計したくなかったためです。

プルーフオブスペース

プルーフオブワーク システムでは計算リソースが使用されますが、プルーフオブスペースでは代わりにストレージ スペースが使用されます。要するに、プルーフオブスペースは、2 名の参加者、すなわちプルーバー (prover) とベリファイヤー (verifier) の間で、プルーバーがある程度のストレージ スペースを保有していることを、ベリファイヤー主導の計算を実行することによって検証する*インタラクティブ*なプロトコルです。その前提は、プルーバーがこれらの計算を保存して呼び出した場合にのみ使用可能になるというものです [63]。適切な方法が見つかるかどうかはわかりませんが、私たちは今後のバージョンのオーキッド プロトコルにプルーフオブスペースを使用する可能性を検討しています。

プルーフオブアイドル

プルーフオブアイドルは、定期的かつ同期されたプルーフオブワークがユーザーのグローバルな計算能力のシェアを示すのに十分であるという追加の仮定に基づいています。残念ながら、ネットワークはまだ初期段階 ($\leq$1,000 万ペドラー) ではありますが、スーパー コンピューティング センターを有する企業が、その計算能力の 1% を犠牲にするだけで、ネットワーク全体を制御できるようになる可能性があります。この攻撃が壊滅的でなくなるのに十分な数のペドラーを擁するにはかなり時間がかかると思われるため、今回のリリースではプルーフオブアイドルを使用していません。

5.3. メダリオンの仕様

大まかに言うと、メダリオンの生成には、(1) 公開/秘密鍵ペア K の生成と最新のイーサリアム ブロック ダイジェスト E の取得、および (2) (反復的または並行して) ソルト S を見つけ、$F_N(K, E, S)$ がなんらかの当選条件を満たして目標達成できるようにするという 2 つのステップがあります。ここで N は、難易度のスケーリング係数です。メダリオンの目標は、特定のエンティティにプルーフオブワークを提供することであることを思い出してください。したがって、メダリオンを使用して複数のピアを偽装できないように、各メダリオンを特定の公開鍵に暗号でリンクする必要があります。さらに、任意のエンティティが活用できる計算前の利点を制限したいと考えています。したがって、メダリオンは、数十秒単位で変化するイーサリアム ブロック ダイジェストに暗号で紐付けられています。以下は、メダリオンの仕様の定義です。

pk_m はピア m に属する一意の公開鍵

sk_m は、pk_m に関連付けられた一意の秘密鍵

e_t は t 時点でのイーサリアム ブロック ダイジェスト

$h(y)$ は、値 y を入力した暗号化ハッシュ関数のダイジェスト

$sig(sk, r)$ は r の基本的な署名で、秘密鍵 sk を使用している

$F_{n,k}(x_j)$ は Equihash の出力値で、2開始カウンター x_j と難易度 (n, k) を伴っている

シードは $h(e_t / sig(sk, e_t))$

$h(y)$ は暗号化ハッシュ関数でもかまいませんが、オーキッド プロトコルでは Keccak を使用します。このハッシュ関数の選択に関する説明については、付録 D.1 を参照してください。基本的な署名は、適切なサイズのデータの秘密鍵によるべき乗と定義します。

これらの定義を使用して、メダリオンを次のように定めます。

$$M = \{t, e_t, pk_m, sig(sk, e_t), F_{n,k}(seed)\}$$

これは、世界的に合意された Equihash 難易度パラメータ (n, k) に関連して定めています。これらのパラメータの詳細については、[50] を参照してください。シード *(seed)* を F への入力値として使用することで、ピアの秘密鍵をメダリオンに暗号的にリンクします。メダリオンはオーキッド マーケットでピアのコード アドレスを決定するため、メダリオンを所有するエンティティは、特定のピアに関連付けられた *pkm* を使用することで検証できます。さらに、エンティティは、特定のコード アドレスから公開鍵の所有権の証明を要求できます。メダリオンのエンジニアリングの詳細については、付録 C を参照してください。

6. 支払い

6.1. オーキッドの支払い要件

ほとんどの支払いシステムでは、ターゲット アイテムのコストはトランザクション コストよりも大幅に高くなっています。特に、ターゲット アイテムのコストは、ある当事者から別の当事者への資金移動に関連するコストよりもはるかに大きくなります。これはほとんどのインターネット購入の場合であり、ネットワーク コストはほとんど些細なコストとして無視される場合があります。ただし、オーキッドのネットワークではターゲット アイテムのコストは帯域幅になります。つまり、回線を介して送信される各パケットには関連するコストが存在します。したがって、支払いを送信するためのトランザクション コストが 1 つのパケットのコストと同じくらい低い場合、これらのコストは等しくなります。もちろん、これはオーキッドのプロトコルの経済的前提を破ることになるでしょう。

任意の精度で帯域幅を販売したいので、**要求する**トランザクションの手数料を任意に低くするために、ユーザーが最小限のトランザクション コストで任意の量の中継トラフィックを支払うことができる新しい形式の支払いシステムが必要です。今、私たちは、任意に低いトランザクション コストと任意の帯域幅の分割性を備えた支払いシステムを必要としています。さらに、オーキッドのプロトコルの目的は、インターネットの監視と検閲を大幅に削減することです。したがって、支払いメカニズムの追加要件には無検閲性、匿名性、信頼できる第三者への非依存性を含める必要があります。つまり、基礎となるネットワークには監視と検閲に耐性があり、支払いメカニズムには耐性がない場合でも、システムは悪用可能であり、ユーザーは検閲または追跡される可能性があります。同様に、信頼できる第三者に依存すると、オーキッドのネットワークは、支払いプロバイダーに影響を与える可能性のある十分な動機を持つ、または強力なエンティティからの干渉にさらされます。

したがって、オーキッドの支払いの要件は次のとおりになります。

1. *経済的実行可能性*。支払いは任意に安くする必要があります。

[2] $F(x_j)$ の出力が一連のカウンター j であることに注意してください。ただし、出力のすべての j について、$XOR_j = h(j)$、$\Sigma XOR_j = 0$。

2. *偽造不可能性*。所有者のみが支払いトークンを支払いに使用できるようにする必要があります。
3. *可用性。オーキッドによる支払いの送信を禁止したり、支払いの受け取りを禁止したりすることはできません。*
4. *不可逆性*。どのようなエンティティも**過去の**支払いを取り消せないようにすべきです。
5. *匿名性*。参加者は、アカウントの住所、支払い金額、または時間とは関連付けられていない必要があります。3

以下のセクションでは、支払いの潜在的なソリューションについて説明します。オーキッドの支払い（セクション 6.5）は匿名性要件を除くすべてを満たすと主張します。

6.2. 従来の支払い

現在の金融支払いシステムでは、トランザクションは支払いカード用の ISO / IEC 7816 [3] や銀行支払い用の EBICS [4] などのプロトコルを使用して、銀行や支払いサービス プロバイダー [2] などの複数のエンティティ間の交渉を通じて決済されます。このようなプロトコルは、SWIFT [5] や NYCE [6] などのネットワークで実行され、国内および国際トランザクションの両方をサポートしています。これらのネットワークを形成するエンティティはそれぞれ独自の元帳を保持し、それらを電子支払い領収書および手動の照合から継続的に更新します [7]。

従来の支払いネットワークに接続するには、通常、ほとんどの司法管轄区で特別な認可が必要であり、接続するエンティティ間の個別のビジネス契約も必要です。結果として生じるグローバルな金融ネットワークは、接続するビジネスの許可されたアドホックなメッシュならびにプロトコルとネットワークの混在と見なすことができます。それぞれの元帳にはある一つの障害が存在し、暗号の整合性に欠けており、支配する事業体の気まぐれで任意に変更することができます。

通常、従来の支払いプロトコルはそれ自体ではトランザクション料金を定義しませんが、プロトコルを実行するエンティティは手数料を上乗せします。トランザクションごとの手数料の範囲は、支払いカード取引の場合の数セントから [8]、国際電信送金の場合の $75 まで [9] あります。多くのシステムでは、代わりに、またはそれに加えて、取引金額のパーセンテージ料金を課金します。これは、銀行振込の場合は 13% [10]、支払いカードの場合は 3.5%[11] です。

従来の支払いは信頼できる当事者に依存しているため、必要な特性を犠牲にせずにオーキッドのネットワークで使用することは事実上不可能です。特に、可逆性が、反転トランザクションの形の設計により存在します [77]。トランザクションは一般的に偽造が困難ですが、クレジットカード詐欺はよくあり、個人情報の盗難やハッキングはユーザー アカウントの侵害につながる可能性があります。さらに、これらの支払いシステムは、不便な時間に誤動作し、定期的にダウンタイムを被る傾向があるため、部分的な可用性しか提供しません。支払いを管理する信頼のおける当事者は通常、送信者、受信者、支払いの金額と時間の記録だけでなく、多くの場合、送信者に関する身元情報も持っているため、匿名性を欠きます。最後に、次のセクションで説明するように、オーキッドのネットワークでは、従来型の支払いオーダーのトランザクション料金は法外に高くなります。

6.3. ブロックチェーン支払い

ビットコインは、従来の支払いシステムの現状に革命をもたらし、支払いと国際送金において世界市場

[3]理想的には、匿名性は悪意のあるオブザーバーに対してだけでなく、送信者または受信者に悪意がある場合にも保持されるべきです。

を揺るがし続けています。ビットコインは、地理的な境界にとらわれないグローバルなネットワークおよびプロトコルです。トランザクションにおいて公開鍵暗号方式を適用することで、信頼できる仲介者を必要とせずに、ユーザー同士が生成したアドレス間で当該額のビットコインを転送します。ユーザーは、公開鍵のハッシュを支払いアドレスとして使用できるキーペアを生成します。そのため、アドレスからの転送に署名するには秘密鍵が必要になります [12]。ビットコインの支払いは偽造不可であり、不可逆的です [79]（ブロック確認を考慮する妥当な時間内で）。ビットコインのネットワークは運用開始以来、最小限のダウンタイムが発生したのみで、マイナーによる積極的な検閲（セクション 6.6 でさらに説明）以外は、一般的に利用可能であると見なすことができます。ビットコインの支払いは疑似匿名であり、匿名性の度合いはネットワークの使用方法に大きく依存します [68]。
一般に、分散型暗号通貨により、人間とコンピューター システムは歴史上初めて、信頼できる第三者なしで価値を取引することができています。これは、オーキッドなどの、動機を持つ分散オーバーレイネットワークにとっては重要な要件です。

ビットコインのトランザクション料金は、トランザクションの額ではなく、トランザクション データ構造のサイズに送信者が設定した係数を掛けたものによって決まります。2017 年まで平均トランザクション料金は 1 ドルを大きく下回っていましたが、2017 年 2 月にビットコイン ネットワークが最大トランザクション容量に達したため、手数料が急上昇しました。平均手数料は 8 ドルにまで上昇し [13]、低料金に依存するアプリケーションのビットコイン ネットワークでの運用は成り立たなくなりました。

イーサリアム ネットワークは公開鍵暗号方式にも根付いており、ビットコインのように、動作検証によって保護されており、偽造不可能性、可用性、そして（非古典的な）不可逆性という同じ特性を導出しています。イーサリアムはより高く動的に調整可能なトランザクション容量を持ち、ネットワークは 2015 年の開始以来低料金になっています。ただし、トランザクション数の増加とイーサリアムの基になるネイティブ トークンであるイーサ、トランザクション料金（*ガス*として知られる）は平均で 0.20 ドル、ピーク時には 1.00 ドルにまで上がりました [14]。スマート コントラクト コードを実行するトランザクションは、実行される計算の量に比例してさらにコストがかかります。
一般的なパブリック ブロックチェーン ネットワークでのトランザクション料金の増加はマイクロペイメントを直接処理する可能性を抑制し、マイクロペイメントを、支払いチャネルなどの第 2 層ソリューションに押し上げています。

6.4. ブロックチェーンベースの確率的マイクロペイメント

確率的支払いをブロックチェーン プロトコルにどのように適用できるかという中核をなすアイデアを簡単に伝えるために、ここでいくつかの詳細を説明します。MICROPAY1 スキームの正式な説明は引用元の論文にあり、オーキッドの確率的支払いスキームについてはセクション 6.5 で定式化されています。

MICROPAY1 [81] は Pass and Shelat により説明することができます。デジタル署名とコミットメントスキームを組み合わせて、正確な確率の無作為な結果を含むリリース条件を設計します。送信者は、まずビットコインを新しく生成されたキーのエスクロー アドレスに転送することにより「デポジット」を行います。次に、受信者（MICROPAY1 用語では「商人」）が乱数を選択し、この番号を介してコミットメントを送信者に送信します。コミットメントに加えて、受信者は新しいビットコイン アドレスも提供します。また、送信者は乱数を選択し、この番号（プレーンテキスト）、受信者からのコミットメント、および受信者が提供した支払い先アドレスなどの他の支払いデータの連結に署名します。

結果のチケットの検証には、受信者のコミットメントが明らかにした番号と一致することの確認と、送信者からの署名がビットコイン入金のアドレスと一致することの確認が含まれます。送信者と受信者からの乱数の XOR の最後の 2 桁が「00」である場合、チケットは有効であり、受信者はそれを使用することができます。

このスキームでは、送信者がコミットメントの拘束属性を破る（または署名を偽造する）ことができる、またはユーザーがコミットメントの隠匿属性を破ることができる場合を除き、偏りのない「コイントス」と考えることができます。

送信者がチケットを複数の受信者に並行して発行することで預金を「二重に使用」するか、受信者からチケットの請求が確認されたときに支出をブロードキャストして受信者を出し抜くことができることに留意してください。MICROPAY1 の作成者は、「ペナルティ エスクロー」によってこれを解決する方法について議論します。送信者によって預けられた第 2 の額が将来のいつかの時点で使われて送信者に戻ることができ、その時まで同じ支払いエスクローに対して 2 つの有効なチケットを送信できる誰かによって「削減」されるか「バーン（焼却）」されます。これにより、送信者が受信者と共謀したり、自身の受信者として動作したりするのを防ぎます。

MICROPAY1 の作成者は、MICROPAY2 および MICROPAY3 の反復的な改善を構築します。ここでは、チケットに対していくつかの計算検証手順を実行し、計算が正しい場合に署名をリリースするために信頼できる当事者が導入されます。

6.5. オーキッドの支払いスキーム

支払いに適した抽象的概念を特定したので、次はどのように実装するかを考える必要があります。

セクション 6.1 で説明した要件に加えて、次の要件も満たします。

- *再利用性*。新しいチケットを作成する方法では、チケットごとに新しいトランザクション料金や新しいオンチェーン トランザクションを必要としてはなりません。そうでないと、トランザクション料金料が再び問題となります。
- 二重支出を防止しなければならず、さもなくば利益をもたらさず、失敗となります。
- システムは、パケットのコストを圧倒しないように、計算コストの点で十分なパフォーマンスを発揮する必要があります。

これらの要件のうち、最後の要素がおそらく最も面倒なものです。私たちの知る限り、イーサリアム トークンに基づいて宝くじチケットを構築する方法は存在せず、ECDSA 署名を検証する順序で計算する必要はありません。このセクションで詳しく説明するように、これは、チケットの量と当選の可能性だけでなく、送信者のイーサリアム アカウントにチケット送信目的で十分な量のオーキッド トークンがロックされていることを、送信者が受信者に暗号的に証明するという要件に基づくものです。

このため、それだけで使用するには不十分でしたが、上述のアプローチに類似した「取引のバランス」アプローチを採用せざるを得ません。これは、新しい要件、すなわち「取引のバランスを十分に小さく維持して、貿易を切断するインセンティブを起こさないようにしなければならない」ということにつながります。これは実装の現実に起因するメカニズム設計の問題であるため、ここではソリューションが存在すると想定して実装に焦点を当て、これ以上の議論はセクション 6.9 で行います。

オーキッドの支払いスキームは、MICROPAY1 や関連する構成要素に触発された、疑似匿名の確率的マイクロペイメント スキームです。イーサリアム スマート コントラクトと大幅なペナルティ デポジットを活用することで、信頼できる当事者を必要とせずに、フロントランニングおよび並行（ダブルを含む）支出攻撃を軽減します。オーキッド ペイメントの疑似匿名性は、通常のイーサリアム トランザクションで達成できるものと同等です（ただし、オーキッドのクライアントは、ワンタイム アドレスやノード ID と支払いアドレス間の鍵分離など、追加のプライバシー技術を使用して限定的な匿名性を実現できています）。

MICROPAY2 および MICROPAY3 で導入された「信頼ある関係者」は、イーサリアム スマート コン

トラク トコードにちょうど置き換えることができます。EVM は、マイクロペイメント チケットを検証するための任意のロジック（計算の経済的範囲内）の実装を可能にし、ECDSA [71] 回復操作と暗号ハッシュ関数のプリミティブ [89] を提供します。支払いスキームの詳細な説明については、付録 D で説明します。

6.6. オーキッド トークン

オーキッドのネットワークでは、偽造不可能性、可用性、不可逆性の支払い要件を満たすために、イーサリアム ベースの ERC20 トークンを使用しています。次のセクションでは、ERC20 転送のトランザクション料金を下げて、任意の小額のトークンを送信できるようにする方法について説明します。支払いの匿名性については、セクション 6.10 で説明します。

オーキッド トークン (OCT) は、オーキッド ネットワーク内での支払いに使用します。オーキッド トークンは、新しいイーサリアムベースの ERC20 互換の固定供給トークンです。供給は 1×10^9 (10 億) トークンに固定されており、各トークンには 1×10^{18} の非分割サブユニット（イーサと同じ分割可能性）があります。

一見すると、以下のセクションで詳しく説明するオーキッド ペイメント システムでは、イーサまたは ERC20 トークンを使用するように構成できます。実際、イーサを使用するとチケット コントラクトが簡素化されてガス コストがわずかに削減され、ユーザーは（トランザクション料金を払って）オーキッド トークンとイーサの両方を取得する必要がなく、イーサのみが必要になるため使いやすさが向上します。

ただし、イーサリアムは、ERC20 トークン [15] [16] を含む任意のメカニズムによってトランザクション料金を支払うことができるように、将来のプロトコルのアップグレードを計画しています。これにより、新しいトークンを使用する場合のほとんどの欠点がなくなります。ガス コストに違いはなく、ユーザーは 1 つのトークンを取得するだけで済みます。コントラクトを実行するうえで [17]、ガス価格をゼロに設定して、ERC20 トークンの支払いをマイナーに追加することもできます (EVM COINBASE [89] オペコードを使用)。これには、ゼロのガス価格を受け入れ、トランザクション実行にコインベース アドレスへの ERC20 トークン転送が含まれることを検証するためにマイニング戦略を構成する必要があるため、マイナーからの明示的な支援が必要になります。

ただし、単にイーサを使用する代わりに新しいトークンを導入する決定は、技術的ではなく社会経済的な理由によるものです。新しいトークンを作成し、オーキッドのネットワークで唯一の有効な支払いオプションにすることで、複雑さの増大を正当化するのに十分重要であると考えられる社会経済効果を作り出します。

6.7. オーキッドのガス コスト

上記のスキームの堅牢性プロトタイプ実装から約 87,000 のガス コストを測定しました。このコストは、入力として当選チケットを使用して呼び出された場合に、チケット請求用の API を完全に実行するためのものです。実行を主張するチケットには、Orchid ERC20 元帳転送 API への *サブコール*が含まれます。すべてのオーキッド スマート コントラクトの堅牢な実装は、暗号レビューと最小限の外部セキュリティ監査の後にオープンソース化されます。

6.8. 検閲抵抗

ほとんどのパブリック ブロックチェーン ネットワークと同様に、イーサリアムのトランザクションは、バリ

データー（イーサリアム ネットワークのマイナー）が作成したブロック内にトランザクションを含めないよう選択しない限り、検閲されることはありません。ブロックは、ハッシュ パワーに比例してすべてのマイナー間で無作為にマイニングされるため、オーキッド ネットワークを大幅に混乱させるには、圧倒的多数のマイナーがオーキッドの支払いを積極的に検閲する必要があります。たとえば、ハッシュ パワーの 90% がオーキッド関連のトランザクションをブロックに含めないことを選択した場合でも、トランザクションの確認に平均で 10 倍の時間がかかることにはなりますが、オーキッドのネットワークは依然として機能します。マイナーの大規模なグループ（51%など）が、オーキッド関連のトランザクションを含むブロックを拒否することで、その検閲を選択する場合には、さらに厳しい形式の検閲になります。これはイーサリアムのプロトコルのルールに従って有効であり、実質的にソフト フォークを形成します。ただし、このようなソフト フォークを作成するために大規模にマイナーを共謀させることは、共謀者にとって重大な利益喪失のリスクを伴います。ソフト フォークが十分なハッシュ パワーを達成できない場合、共謀しているマイナーたちはブロック報酬を得られないことになるからです。利益喪失のリスクがなくても、イーサリアムのマイナーの分散化された性質と、ブロックチェーン マイニング戦略に関する法的および規制上の制限がないことから、この可能性は極めて低いと考えています。

6.9. 取引のバランス

2 人のオーキッド参加者であるアリス (Alice) とボブ (Bob) が、完全に匿名でのトランザクションを望んでいるとします。ボブは x 課金するタスクを実行し、アリスは彼に y タスクごとに支払うことになっています。あいにく、匿名性の性質上、事前のトランザクションがなければ、アリスとボブにはお互いを信頼するメカニズムがありません。この 2 人は協力できるでしょうか?

アリスとボブの関係にセットアップ コストがあるとすれば (S_{Alice}, S_{Bob} s.t. $S_{Alice} > xy, S_{Bob} > xy$)、答えはイエスです。(1) アリスが求めていた仕事の総量が $\leq xy$ であるか、または (2) ボブが実行できる仕事の総量が $\leq xy$ でない限り、資金を持って逃げたり、仕事を終わらせたりすることが経済的には合理的となります。オーキッド マーケットの説明 (セクション 4) にあるように、オーキッドのネットワークには、1×10^3 パケットを超える取引の不均衡をサポートするセットアップ コストが存在します。オーキッド マーケットの売り手は通常、買い手よりも高いセットアップ コストを支払うため、またカスタマーは必要な作業量を非対称的に知っているため、オーキッドのネットワークにはカスタマーの前払いがあります。

6.10. 匿名性

前のセクションで説明したオーキッドの支払いは、通常のイーサリアム トランザクションと同様に疑似匿名で、金額、送信者および受信者のアカウントを含むすべてのトランザクションは公開されています。オーキッド クライアントは、ワンタイム アドレス [18] や HD ウォレット [19] などの最新のウォレット技術により、パブリック ブロックチェーン トランザクションのデフォルトの疑似匿名性を改善し、単一のルートキーを使用しているにもかかわらず、支払いアドレスのリンク不能性を提供することを目指しています。

イーサリアム ビザンチウム リリースでは、楕円曲線演算に新しい EVM プリミティブを活用することで、合理的なガス コストでリンク可能なリング署名を実装できるようになりました [20]。イーサリアム スマート コントラクトと HD ウォレットやリンク可能なリング署名が提供するステルス アドレスを組み合わせることで、メビウス [76] ミキシング サービスなどのミキシング テクノロジーが有効になります。メビウスは、サービスをミキシングするためのゲームベースのセキュリティ モデルを使用して暗号的に証明された、強力な匿名性保証を提供します。ただし、以前のミキシング テクノロジーとは異なり、悪意のあるオブザーバーおよび送信者に対して匿名性を提供しますが、悪意のある受信者に対しては提供しません。メビウスなどのサービスとオーキッドの確率的マイクロペイメントを組み合わせると、支払いに関する最

終要件である匿名性に近づくことができます。

支払いのオブザーバーか、送信者か受信者かにかかわらず、悪意のある行為者に対する完全な匿名性の保証を実現するには、ゼロ知識技術を検討する必要があります。

7. 帯域幅マイニング

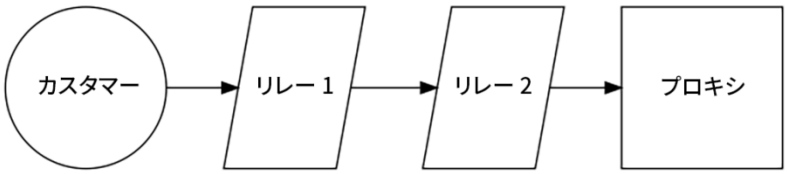

図 2: カスタマー向けの 3 ペドラー チェーン ルーティング トラフィック

このセクションでは、リレーとプロキシ　の動作の仕様について説明し、検閲不能の匿名 Web ブラウジングをサポートするために、これらのノードの「連鎖」について説明します。

帯域幅販売のための仕様

リレー ノードは比較的単純な動作パターンを実装します。

- それぞれが独自の暗号化キーを持つ 1 つ以上の接続を維持する。
- 受け取ったチケットと当選者を確認する。
- 取引のバランスを監視し、事前に宣言された金額を超えた場合は切断する。
- 開いている接続からデータを受信し、メッセージ境界で復号化を実行する。
- 復号化されたメッセージを次のように処理する。
  - 非制御セグメントをメッセージで指定された接続に転送する。
  - 制御セグメントを処理する。
    - *ダミー データ。*このセグメントを破棄するようにリレーに指示する。
    - *レートで消費。*固定レートで接続を介してデータを送信し、レートを維持するために必要に応じてパケットをキューに入れてデータを生成するようにリレーに指示します。
    - *ラチェット チケット。*このパケットを受信したピアにチケットを渡すようにリレーに指示します。
    - *接続の開始。*新しい接続を確立するようにリレーに指示します。セットアップ中および切断の処理に使用されます。
    - *初期 Web 接続。*(プロキシのみ。)指定されたホストへの SSL 接続を開くようにプロキシに指示します。ホワイトリストのサポートのため、これを生の IP アドレスにすることはできません。

上記の動作における重要な考慮事項は、継続的にリレーの作業証明が不要なことです。WebRTC 接続であるすべての接続と組み合わせると、純粋な javascript リレー コードを実行することで、訪問者を収益化する可能性がある Web サイトへの扉が開かれたままになります。

アプリケーション固有の制御セグメントを介した可能な拡張の説明については、セクション 10 を参照してください。

ガード ノードと「帯域幅の消費」

カスタマーが接続されているリレーには非常に重要な情報が含まれています。それはカスタマーの IP アドレスです。カスタマーはこれを可能な限りプライベートに保ちたいと考えていると想定し、デフォルト クライアントは最初のホップとして長寿命のピアを望むことを表明します。

共謀 (セクション B.1) に起因する情報攻撃の説明で詳細に説明されている、最初のホップのノードに関するもう 1 つの懸念は、タイミング攻撃を実行するのに最適なポジションに置かれていることです。

これらの攻撃を防ぐために、プライバシーを重視するユーザーは、*帯域幅の消費* (一定量の帯域幅をカスタマーに送信するために 2 番目のホップを支払う) と呼ばれる方法を推奨します。このアプローチでは、ネットワークの使用とはまったく関係のないデータ使用が発生するため、このアプローチでは、リレー 3 のインバウンド トラフィックを確認できない敵によるタイミング攻撃が防止されます。

回避を求めるユーザーに支援を提供するために (セクション B.3)、帯域幅の消費は、オーキッド以外の広く使われている WebRTC プロトコルの統計的特性によって決定される非固定レートもサポートします。

連鎖 (チェーン)

匿名インターネット アクセスにリレーを使用することに関心のあるカスタマーは、上記の仕様を使用してリレーの「チェーン」を作成します。

8. パフォーマンスのスケーリング

このセクションでは、ユーザー数が増えるにつれてシステムがどのように機能するかを調べます。

アルゴリズム性能

大まかに言って、オーキッドのプロトコルは 3 つの部分に分けることができます。イーサリアムベースの支払い、マニホールド、そしてオーキッド マーケットの 3 つです。

イーサリアムベースの支払いは、イーサリアムを通常のトランザクションとしてスケーリングします。イーサリアムのシステム設計を検討した結果、オーキッドのネットワークが非常に成功し、イーサリアムの総トランザクション量のかなりの割合になったとしても、このコンポーネントは設計許容範囲内で機能すると確信しています。

マニホールドは帯域幅の売り手のチェーン (リレーとプロキシ) であり、それらはすべて、オーキッドのネットワークの参加者総数に依存しないパフォーマンス特性を持っています。

オーキッドのマーケットの中核事業は、よく研究されたコード DHT に基づいています。ペドラーが維持する必要がある接続の数は、最大 256 接続まで $O(log(n))$ の割合で増加します。ネットワーク上のクエリには $O(log(n))$ ホップが必要です。これらの操作は、ネットワークのサイズが大きくなるにつれて負担が大きくなりますが、私たちはパフォーマンスに大きな影響が生じるとは考えていません。

乏しいリソースの割り当て

オーキッドのプロトコルはトークンを中心に構築されています。これらのトークンにより、価格発見を通じて、買い手と売り手の間のバランスの変化を適切に処理できます。

たとえば、リレーの供給が不足している場合、すべてのカスタマーにスローな経験を提供するのではなく、カスタマーは不足が修正されるまで誰がシステムを使用できるかを決める入札に参加します。逆に、リレーが十分に供給されている場合、一部のリレーは価格が上昇するまでシステムを離れることがあります。

実世界の性能

ソフトウェアはまだ完成していないため、ここで提供する具体的な数字はありません。リリース時には、本文書に次のグラフを記載する予定です。

1. オーキッド マーケットの規模に応じたチェーン設定時間。
2. オーキッド マーケットの規模に応じたオーキッド マーケット参加時間。
3. 価格が希少性、豊富さにどれだけ早く適応するか。
4. 皆様の面白いアイデアを追加してください!

9. 外部ライブラリ

オーキッドの機能は、いくつかの重要なプリミティブに基づいています。読者全員がこれらのプリミティブ、もしくはオーキッドのネットワークで使用される特定のプロパティに精通しているとは限らないため、ここで簡単にまとめて説明します。

WebRTC

WebRTC [47] は、もともと Web ブラウザ間のリアルタイム通信を容易にするために設計されたシステムです。STUN、ICE、TURN、および RTP-over-TCP を含む、NAT およびファイアウォール トラバーサル方式の優れた実装を提供します。カスタムコードの TCP および UDP ネットワーク コードではなく、WebRTC をネットワーク プロトコルの基盤として選択することにより、これらのテクノロジーの世界クラスの実装を得て、(ある程度) ユーザーのトラフィックを一般的な Web トラフィックとしてマスク化します。

NACL

NaCL [48] (「ソルト」と発音) は Daniel J. Bernstein et al. による暗号化ライブラリで、高レベルの暗号化ツールの構築に必要なコア操作の構築に焦点を当てています。NaCL とその開発者の評判に基づいて、このプロジェクトの暗号プリミティブのソースとして選ばれました。以下で説明するすべての暗号化操作は、イーサリアム スマート コントラクト暗号化コードを除き、NaCL を使用して実装されます。

イーサリアム

イーサリアム [55] は、ネイティブ通貨 (ETH) とチューリング完全なスマート コントラクトを含む分

散型ブロックチェーンおよびプラットフォームです。スマート コントラクトがオーキッドの設計に非常に有用であることが判明したため、支払い残高の追跡およびオーキッドの支払いチケットの検証と公平性に関連する多くの設計上の懸念を取り払うことができます。

10. 今後の作業

このセクションの項目は、次の 2 つのカテゴリに分類されます。あればよい機能と、一般に公開することに関して内部で議論している機能の 2 つです。私たちは、この矛盾性が普遍的であると信じています。誰でも力によって抑圧されている例はすぐに挙げることはできると思いますが、善のために使用されている力の例も無数にあります。オーキッドのようなプロトコルには独自の判断がないため、トラフィック ルーティングを誰のために行っているか、自由の戦士なのかテロリストなのか、悪者なのかヒーローなのか、判断することができません。

プルーフオブスペース

セクション 5 でも述べたように、私たちは代替となるプルーフタイプの調査に多くの関心を寄せています。これは、プルーフオブワーク システム環境への影響と、現在のプルーフオブワーク アルゴリズムがネットワーク ルーターとして機能する完全なコンピューターを必要とするため、重要な問題です。
私たちは、古い電話や同様のハードウェアがオーキッドのネットワークに有益に参加できるようにするために、セキュリティの中核で不足しているリソースとしてディスク スペースを使用する可能性を追求することにワクワクしています。

コンテンツ ホストの保護

多くの従来のアプローチ（セクション 2）により、コンテンツ ホストが Web ユーザーと同様の保護を求めていることが判明しています。私たちは（たとえば、核兵器の製造に関連する情報など）自由に配布できないことが公共の利益につながるコンテンツがあると考えているため、社内で双方の意見が対立しています。ただし、予期していない状況で要求された場合、オーキッドを拡張して、次の図に示すような「制限のない監視されていないホスト」をサポートすることができます。

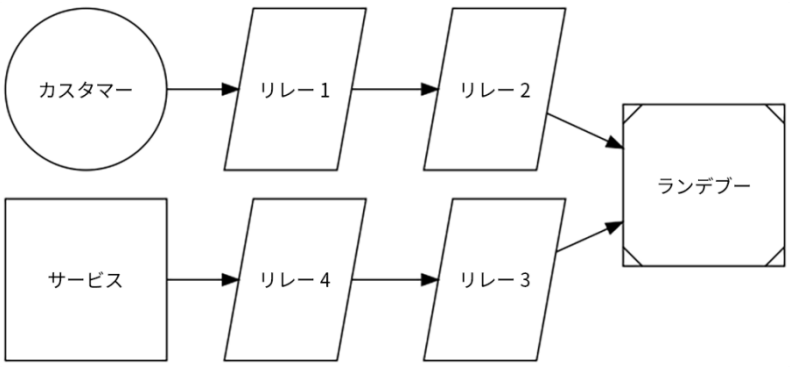

図 3: サービスとカスタマーの間のリレーとして機能するランデブー ノード

イーサリアム トラフィックの保護

ファイアウォールの回避に関するセクション（セクション B.3）で説明しているように、クライアントのイーサリアム ネットワーク トラフィックは弱いリンクである可能性があります。すべてのノードがこの情報を維持する必要があるため、オーキッド プロトコルを使用してイーサリアム情報を配布するのは自然なことのように思えます。

残念ながら、支払いに関する情報の対価を支払う相手に依存すると、厄介な問題につながります。近い将来これを追加する予定ですが、最初のリリースには含めません。

プラットフォームとしてのオーキッド

コア システムの設計に当面の時間の大部分を費やすと予想していますが、次の利用事例をサポートする機能を追加すると、オーキッドのネットワークを介してルーティングされる帯域幅の量が大幅に増加する可能性に非常に興味があります。

1. Web サイトがネットワークに直接接続し、サービスにトークンを組み込むための API。
2. ネットワーク上のファイル ストレージと静的な Web サイト ホスティング。
3. ファイル共有。
4. メール/メッセージング サービス。
5. 仲裁/調停サービス。

参考文献

[1] URL: http://www.meshlabs.org.
[2] URL: https://en.wikipedia.org/wiki/Payment_service_provider.
[3] URL: https://en.wikipedia.org/wiki/ISO/IEC_7816.
[4] URL: http://www.ebics.org/home-page.
[5] URL: https://www.swift.com.
[6] URL: http://www.nyce.net/about.
[7] URL: http://www.investopedia.com/terms/r/reconciliation.asp.
[8] URL: https://www.quora.com/What-are-common-credit-card-processing-fees.
[9] URL: https://www.nerdwallet.com/blog/banking/wire-transfers-what-banks-charge.
[10] URL: https://www.economist.com/blogs/dailychart/2010/12/remittances.
[11] URL: https://www.valuepenguin.com/what-credit-card-processing-fees-costs.
[12] URL: https://bitcoin.org/en/developer-guide#transactions.
[13] URL: https://bitinfocharts.com/comparison/bitcoin-transactionfees.html.
[14] URL: https://bitinfocharts.com/comparison/ethereum-transactionfees.html.
[15] URL: https://blog.ethereum.org/2015/07/05/on-abstraction.
[16] URL: https://blog.ethereum.org/2015/12/24/understanding-serenity-part-i-abstraction.
[17] URL: https://github.com/ethereum/EIPs/issues/662#issuecomment-312709604.
[18] URL: https://en.bitcoin.it/wiki/Address_reuse.
[19] URL: https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki.
[20] URL: https://ropsten.etherscan.io/address/0x5e10d764314040b04ac7d96610b9851c8bc02815# code.
[21] URL: https://p16.praetorian.com/blog/man- in- the- middle- tls- ssl- protocol-downgrade- attack.
[22] URL: http://ethgasstation.info.
[23] URL: https://etherscan.io/token/OmiseGo.
[24] URL: https://solidity.readthedocs.io/en/develop/assembly.html.
[25] URL: https://hackernoon.com/zksnarks-and-blockchain-scalability-af85e350a93a.
[26] URL: https://hackernoon.com/scaling-tezo-8de241dd91bd.
[27] URL: http://mojonation.net.
[28] URL: https://en.bitcoin.it/wiki/Payment_channels.
[29] URL: https://en.bitcoin.it/wiki/Contract.
[30] URL: https://lightning.network/lightning-network-paper.pdf.
[31] URL: https://en.wikipedia.org/wiki/Mining_pool#Mining_pool_methods.
[32] URL: https://bitcoin.stackexchange.com/questions/1505/what-is-a-share-can-i-find-it- while-mining-solo-or-only-when-pool-mining.
[33] URL: https://blog.coinbase.com/app-coins-and-the-dawn-of-the-decentralized-business-model-8b8c951e734f.
[34] URL: http://onchainfx.com.
[35] URL: https://0xproject.com/.
[36] URL: https://etherdelta.com/#REQ-ETH.
[37] URL: https://augur.net.
[38] URL: https://gnosis.pm.
[39] URL: https://medium.com/@vishakh/a- deeper- look- into- a- financial- derivative-on- the- ethereum-blockchain-47497bd64744.
[40] URL: https://tools.ietf.org/html/draft-goldbe-vrf-00.
[41] URL: https://blog.ethereum.org/2017/10/12/byzantium-hf-announcement.
[42] URL: https://github.com/ethereum/EIPs/pull/213.
[43] URL: https://github.com/ethereum/EIPs/pull/212.
[44] URL: https://theethereum.wiki/w/index.php/ERC20_Token_Standard.

[45] Martin Abadi et al. “Moderately hard, memory-bound functions”.In:*ACM Transactions on Internet Technology (TOIT)* 5.2 (2005), pp. 299–327.
[46] Iddo Bentov, Rafael Pass, and Elaine Shi.“Snow White:Provably Secure Proofs of Stake.” *In:IACR Cryptology ePrint Archive* 2016 (2016), p. 919.
[47] Adam Bergkvist et al. “Webrtc 1.0:Real-time communication between browsers”.In:*Working draft, W3C* 91 (2012).
[48] Daniel Bernstein, Tanja Lange, and Peter Schwabe.“The security impact of a new cryptographic library”.In:*Progress in Cryptology–LATINCRYPT 2012* (2012), pp. 159–176.
[49] Jean-Luc Beuchat et al. “High-Speed Software Implementation of the Optimal Ate Pairing over Barreto–Naehrig Curves”.In:*4th International Conference on Pairing-Based Cryptography*.Springer.2010.URL: https://eprint.iacr.org/2010/354.pdf.
[50] Alex Biryukov and Dmitry Khovratovich.“Equihash:Asymmetric proof-of-work based on the gener- alized birthday problem”.In:*Ledger* 2 (2017).
[51] N. Bitansky et al. “Recursive composition and bootstrapping for SNARKs and proof-carrying data”.In:*In Proceedings of the forty-fifth annual ACM symposium on Theory of computing*.ACM.2013, pp. 111–120. URL: https://eprint.iacr.org/2012/095.pdf.
[52] Nikita Borisov et al. “Denial of service or denial of security?” In:*Proceedings of the 14th ACM con- ference on Computer and communications security*.ACM.2007, pp. 92–102.
[53] Michael Brown et al. “Software implementation of the NIST elliptic curves over prime fields”.*In:Topics in Cryptology—CT-RSA 2001*.Springer.2001, pp. 250–265. URL: https://pdfs.semanticscholar.org/ac3c/28ebf9a40319202b3c4f64cc81cdaf193da5.pdf.
[54] David Brumley and Dan Boneh.“Remote timing attacks are practical”.In:*Computer Networks* 48.5 (2005), pp. 701–716.
[55] Vitalik Buterin.*Ethereum:A next-generation smart contract and decentralized application platform.* https://github.com/ethereum/wiki/wiki/White-Paper.Accessed:2016-08-22.2014. URL:https://github.com/ethereum/wiki/wiki/White-Paper.
[56] David L Chaum.“Untraceable electronic mail, return addresses, and digital pseudonyms”.In:*Communications of the ACM* 24.2 (1981), pp. 84–90.
[57] Shuo Chen et al. “Side-channel leaks in web applications:A reality today, a challenge tomorrow”.*In:Security and Privacy (SP), 2010 IEEE Symposium on*.IEEE.2010, pp. 191–206.
[58] Alessandro Chiesa et al. “Decentralized Anonymous Micropayments”.In:*Annual International Conference on the Theory and Applications of Cryptographic Techniques*.Springer.2017, pp. 609–642.
[59] George Danezis.“The Traffic Analysis of Continuous-Time Mixes”.In:*Proceedings of Privacy Enhanc- ing Technologies workshop (PET 2004)*.Vol. 3424. LNCS.May 2004, pp. 35–50.
[60] Roger Dingledine, Nick Mathewson, and Paul Syverson.Tor:*The second-generation onion router*.Tech. rep. Naval Research Lab Washington DC, 2004.
[61] Cynthia Dwork, Moni Naor, and Hoeteck Wee.“Pebbling and proofs of work”.In:*CRYPTO*.Vol. 5. Springer.2005, pp. 37–54.
[62] Kevin P Dyer et al. “Peek-a-boo, i still see you:Why efficient traffic analysis countermeasures fail”.In:*Security and Privacy (SP), 2012 IEEE Symposium on*.IEEE.2012, pp. 332–346.
[63] Stefan Dziembowski et al. “Proofs of space”.In:*Annual Cryptology Conference*.Springer.2015, pp. 585–605.
[64] Christoph Egger et al. “Practical attacks against the I2P network”.In:*International Workshop on Recent Advances in Intrusion Detection*.Springer.2013, pp. 432–451.
[65] Ittay Eyal and Emin Gu¨n Sirer.“Majority is not enough:Bitcoin mining is vulnerable”.In:*International conference on financial cryptography and data security*.Springer.2014, pp. 436–454.
[66] Mike Hearn.[*Bitcoin-development] Anti DoS for tx replacement*.Bitcoin development mailing list.2013.URL:https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2013-April/002417.html.
[67] Martin Hellman.“A cryptanalytic time-memory trade-off”.In:*IEEE transactions on Information Theory* 26.4 (1980), pp. 401–406.

[68] J Herrera-Joancomart´ı."Research and challenges on bitcoin anonymity".In:*In Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance*.Springer.2015, pp. 3-16. URL: https://www.researchgate.net/profile/Jordi_Herrera-Joancomarti/publication/281773799_Research_and_Challenges_on_Bitcoin_Anonymity/links/55f7c7d408ae07629dcbc471.pdf.

[69] Douglas R. Hofstadter.*Metamagical Themas:Questing for the Essence of Mind and Pattern*.New York, NY, USA:Basic Books, Inc., 1985. isbn:0465045405.

[70] Nicolas Houy."It Will Cost You Nothing to'Kill'a Proof-of-Stake Crypto-Currency".In:*Browser Download This Paper* (2014).

[71] Don Johnson, Alfred Menezes, and Scott Vanstone."The elliptic curve digital signature algorithm (ECDSA)".In:*International Journal of Information Security 1, no. 1 (2001)*.Springer, pp. 3–63. URL:http://residentrf.ucoz.ru/_ld/0/34_Digital_Signatu.pdf.

[72] Aggelos Kiayias et al. "Ouroboros:A provably secure proof-of-stake blockchain protocol".In:*Annual International Cryptology Conference*.Springer.2017, pp. 357–388.

[73] Joshua A. Kroll, Ian C. Davey, and Edward W. Felten."The economics of Bitcoin mining, or Bitcoin in the presence of adversaries".In:*Proceedings of WEIS (Vol. 2013)*.WEIS.2013, p. 11. URL:http://www.thebitcoin.fr/wp-content/uploads/2014/01/The-Economics-of-Bitcoin-Mining-or-Bitcoin-in-the-Presence-of-Adversaries.pdf.

[74] Thomas Locher et al. "Free riding in BitTorrent is cheap".In:*Proc.Workshop on Hot Topics in Networks (HotNets)*.2006, pp. 85–90.

[75] Daniel Lorimer.*Momentum–a memory-hard proof-of-work via finding birthday collisions, 2014*. URL:http://www.hashcash.org/papers/momentum.pdf.

[76] Sarah Meiklejohn and Rebekah Mercer."Möbius:Trustless Tumbling for Transaction Privacy".In:2017.URL:https://allquantor.at/blockchainbib/pdf/meiklejohn2017moebius.pdf.

[77] Adnan Noor Mian et al. "Enhancing communication adaptability between payment card processing networks".In:*IEEE Communications Magazine, 53(3)*.IEEE.2015, pp. 58–64.

[78] Satoshi Nakamoto.*Bitcoin:A peer-to-peer electronic cash system*.2008.

[79] Arvind Narayanan et al. *Bitcoin and Cryptocurrency Technologies:A Comprehensive Introduction*.Princeton University Press, 2016.

[80] Sunoo Park et al. *Spacecoin:A cryptocurrency based on proofs of space*.Tech. rep. IACR Cryptology ePrint Archive 2015, 2015.

[81] Rafael Pass and Abhi Shelat."Micropayments for Decentralized Currencies".In:*Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*.ACM.2015.URL:https://pdfs.semanticscholar.org/bca9/92b35d844160b30edbbf1809e17551d867ea.pdf.

[82] Ronald L Rivest."Electronic Lottery Tickets as Micropayments".In:*International Conference on Financial Cryptography*.Springer.1997. URL:http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.135.2668&rep=rep1&type=pdf.

[83] David L. Salamon et al. "How to make Chord correct".In:(2017). URL:https://orchidprotocol. com/whitepaper.pdf.

[84] Rabin Silvio and Vadhan."Verifiable Random Functions".In:*Foundations of Computer Science, 1999. 40th Annual Symposium on*.IEEE.1999. URL:https://dash.harvard.edu/bitstream/handle/1/5028196/Vadhan_VerifRandomFunction.pdf.

[85] Ion Stoica et al. "Chord:A scalable peer-to-peer lookup service for internet applications".In:*ACM SIGCOMM Computer Communication Review* 31.4 (2001), pp. 149–160.

[86] John Tromp."Cuckoo Cycle:a memory-hard proof-of-work system." In:*IACR Cryptology ePrint Archive* 2014 (2014), p. 59.

[87] Liang Wang and Jussi Kangasharju."Real-world sybil attacks in BitTorrent mainline DHT".In:*Global Communications Conference (GLOBECOM), 2012* IEEE.IEEE.2012, pp. 826–832.

[88] David Wheeler."Transactions using bets".In:*International Workshop on Security Protocols.Springer*.1996, pp. 89–92.

[89] Gavin Wood.ETHEREUM:A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER.
URL:https://ethereum.github.io/yellowpaper/paper.pdf.

[90] Pamela Zave."Orchid:A Fully Distributed, Anonymous Proxy Network Incentivized Through Band- width Mining".In:*arXiv preprint arXiv:1502.06461* (2015).

A. オークション

帯域幅を購入するとき、価格に敏感なカスタマーは、大幅なに低価格を提供する攻撃者に利用される恐れがあります。たとえば、最低入札価格で長さ 4 のチェーンを購入しようとしているカスタマーがいるとします。これを知っている攻撃者は、価格を可能な限り最小額に設定することにより、$\frac{a}{n}$より高い確率でチェーン内の各ノードで選択される可能性を実現します。

これに対処するために、オーキッドのマーケットを利用しているカスタマーは、帯域幅の売り手人口から無作為にその一部を選択し、その一部から作成可能な必要な長さの一連のチェーンからプロバイダーを無作為に選択します。

これにより、攻撃者は割り当てられた場所と、無作為に選択された他の売り手により設定された価格に関連して選択した価格の両方で、「運を天に任せるしかない」状況に陥ります。この制約がある場合でも、売り手が価格変更を通じて需要に影響を与えられることなど、従来の市場の特性は維持されます。

A.1. 付録の概要

この付録では、プライバシーを重視する帯域幅の買い手が利用できる戦略と、さまざまな戦略の効果について説明します。

この問題の分析に適したオーキッド マーケットの単純化されたモデルが導入され、いくつかのアプローチについて理論と具体例を分析しています。このセクションの読者が、帯域幅購入用のアルゴリズムを選択する際のトレード オフについて理解し、別の方法を採用するためのクライアントのハードコード化に進まないことを願っています。

A.2. 分析用の簡易モデル

オークション戦略を評価するために、オーキッド マーケットの完全な複雑さを考慮する必要はありません。分析を簡略にするために、ここでは参加者の目標に関する一連の仮定を立てます。

1. **売り手。**売り手はアクセス販売用の r の「帯域幅スロット」を所有しています。売り手の唯一の目標は、この収入源からの収益を最大化することです。
2. **攻撃者。**攻撃者は $a \geq 2$ 人の売り手を所有しています。攻撃者の唯一の目標は単一の買い手に売り手から複数のスロットを購入させることであり、これは攻撃の成功を意味します。
3. **買い手。**買い手は、3 人の異なる売り手（つまり攻撃者を阻止）から 3 つの帯域幅スロットをできるだけ安い価格で購入しようとします。

オーキッド マーケットの詳細にこだわる代わりに、すべての買い手が、現在のすべてのメダリオン所有者とその現在の帯域幅価格の最新リストを所有していると仮定します。また、リレーとプロキシの区別や、ホワイトリストやその他の機能フィルタリングによって加わる複雑さについてもここでは取り上げません。

基本の構造と参加者の目標を設定したところで、ゲームの構造を具体化していきましょう。

1. **状況。**買い手の戦略は、すべての売り手とすべての攻撃者に伝えられます。売り手の戦略も攻撃者に伝えられます（買い手の戦略に応じて変わる可能性があります）。
2. **フェーズ 1。**すべての売り手が価格を選択します。次に、売り手の価格が攻撃者に伝わり、攻撃者が自分の価格を選択します。

3. **フェーズ 2。**すべての価格が買い手に伝わります。買い手は、無作為の順序で、リストから最大 3 つのオファーを選択するように求められます。
4. **フェーズ 3。**利益が分配されます。
   (a) 売り手と攻撃者は、彼らを選択した購入者から提供価格と等しい金額を受け取ります。
   (b) 買い手は、攻撃を受けずに 3 つのスロットを購入した場合、それらのスロットに支払った額を差し引いた利益 (買い手によって額は異なる) を受け取ります。
   (c) 買い手への攻撃に成功した攻撃者は、その報奨金 U_b (買い手によって額は異なる) を受け取ります。

このマーケットで「N プレーヤー ゲーム」が形成されたので、ここで買い手がすべきこと、売り手がすべきこと、そして攻撃者がすべきことについて考えていきます。買い手はニーズにおいて最も経済的に正常ではない立場にあるため、上記の設定では買い手が最初に動くようにしました。

上記のゲームで買い手の戦略が攻撃者と共有されているという考えに異論を唱える読者もいるかもしれません。私たちがこのように設定したのは、熱心な攻撃者であれば最初の攻撃に失敗したとしても、特定の買い手が支払った大まかな価格に関する情報を得ることができる可能性があるためです (たとえば、すべての売り手の IP アドレスと価格を把握し、その買い手のインターネット接続を監視することで、どの帯域幅の売り手が最初のホップとして使用されたかを判断する)。時間の経過とともにこのような情報漏えいが戦略推測の過程に組み込まれていく可能性を考慮して、この分析においては単に攻撃者が戦略情報を得ていると推測するのが適切であると考えました。

成功基準

ソリューションの成功は、次の各基準のパフォーマンスによって決まります。

1. **セキュリティ。**予算とチェーンの長さが固定されていれば、カスタマーは攻撃から最大限に保護されます。
2. **安定性。**3 つのグループのいずれのメンバーも、戦略を変更することで自らの満足度を向上することはできません。
3. **経済的適合。**売り手は、価格を上げ下げすることで受け取るオーダーの数を調整することができ、売り手の利益を最大化するために使い慣れた方法を選んで採用することができます。

最適化に精通しているがゲーム理論には精通していない読者であれば、特に安定性について注目してください。グループのメンバーが全体の利益を守るために集結し、グループ レベルの「最適戦略」を提案する誘惑にかられることがあるためです。そのような行動は表面的には合理的に見えるかもしれませんが、完全に匿名の分散市場に存在するインセンティブ構造では、それが安定的に行われることは難しくなっています。たとえば、オファー中の合計スロットが需要を超える状況にある複数の売り手がまとまり、最低価格を設定する可能性があります (経済用語では「*トラスト*」、ゲーム理論用語では「超合理性」 [69]、マルクス主義用語では「階級革命」と呼ばれます)。残念ながら、価格をトラスト価格未満に引き下げた売り手は追加の利益を得ることができるため、そのような取り決めは安定しません。このため、この分析ではそれらを考慮しません。ただし、この考えを将来的に完全に除外するものではありません。ブロックチェーン技術の将来の進歩により、そのような合意の遵守に対する分散検証が可能になる時代がくるでしょう。

考えられるもう一つのサプライズは、買い手を最大限に利用する攻撃者の行動を明示的に決定しようとし、そのような戦略の安定性を「成功」の基準として含めることです。これを行うのは、所与のアプロー

チのセキュリティに境界を設ける方法が他にないためです。

A.3. 選択攻撃

買い手戦略の議論に深く入り込む前に、まず攻撃者の目標と、攻撃の構成要素について考えてみましょう。無限の予算を持ち、売り手リストにある以外の情報を持たない偏執的な買い手がいるとしましょう。このような買い手にとって無作為抽出より良い戦略はないことは明白です。これにより、攻撃者の攻撃成功率は $\left(\frac{a}{n}\right)^2$ *となります。*これが可能な限りの最良の結果であるため、この確率の攻撃に対処可能な戦略を検討します。

この点への攻撃は、攻撃者が攻撃の成功率を $\left(\frac{a}{n}\right)^2$ より高める可能性を含む方法となります。

A.4. 戦略の候補

次に、買い手の戦略の評価に進みます。

最低価格

買い手が最低コストのプロバイダーを選択した場合、コンポーネントの攻撃者は価格を最低許容値（0 トークン）に設定します。これにより、$\frac{a}{z}$の攻撃が成功します。ここで z は、ゼロ課金するノードの数です。

たとえば、3 人の純粋な売り手が、$\{2, 4, 6\}$ のそれぞれの価格で単一のスロットを提供している市場を考えてみましょう。また、最大合計価格が 12 の買い手が一人いるとします。賢明な攻撃者であれば $\{0, 0\}$ の価格でマーケットに参入するため、販売が確実に成立し、その場合の攻撃の成功率は「1」になります。

価格加重無作為

買い手が帯域幅のコスト差の単調な機能を購入する機会を設けることを選択した場合、これにより次のパフォーマンスがやや向上します。

$$\frac{af(0)}{\sum_{e \in S} f(e)}$$

上記の例に戻り、コストの逆二乗増加を関数として使用すると、$\frac{228}{337} \approx 85\%$ の確率で攻撃が成功します。85% の方が 100% よりもはるかにましですが、それでも満足のいくものではありません。

手頃な価格のリレーからの無作為抽出

買い手が、$\frac{1}{3}$以下の金額を請求する売り手から無作為に選択することを選んだ場合、コンポーネントの攻撃者は、その最大価格以下に価格を設定します。疑わしい場合、攻撃者は再び価格を 0 に設定できます。

例に戻ると、コンポーネントの攻撃者は 0 および 1 の価格、または同等の価格で市場に参入し、2 つの非攻撃の組み合わせ $\{(0, 2, 4), (1, 2, 4)\}$、および 2 つの攻撃の組み合わせ $\{(0, 1, 2)$ と $(0, 1, 4)\}$

につながります。これにより、攻撃が成功する可能性が $\frac{1}{2}$ になります。

コスト次第の無作為抽出

買い手が 3 人の売り手 (S_i, S_j, S_k) からの合計コストが最大コスト以下になるように無作為に選択することを選んだ場合、攻撃者は (1) 3 つの組み合わせの数 (A_i, A_j, S_k) を最大化するか (買い手が購入可能な範囲)、(2) 3 つの組み合わせの数 (A_i, S_j, S_k) を最小化するよう (買い手が購入可能な範囲) に価格を選択できます。

例に戻ると、賢明な攻撃者は 1 および 4.1 の価格、または同等の価格で市場に参入し、5 つの非攻撃の組み合わせ $\{(1, 2, 4), (1, 2, 6), (1, 4, 6), (2, 4, 4.1), (2, 4, 6)\}$、そして 3 つの攻撃の組み合わせ $\{(1, 2, 4.1), (1, 4, 4.1), (1, 4.1, 6)\}$ につながります。そのため、攻撃が成功する可能性は $\frac{3}{8}$ となります。

攻撃者が 4.1 の価格ポイントを選択することによって達成される効果的な「クラウディング アウト (押し出し)」に注目してください。4.1 を含む 4 つの組み合わせの 1 つだけが*成功しない*攻撃になります。攻撃者の側でそのような行動を考慮した後でも、6 を課金する売り手を除外するよりもコスト次第の無作為抽出が優れていることは注目に値します。

ピアごとに正規化されたコストの無作為抽出

問うべき質問の一つに、ピアが過小評価されないように無作為抽出にバイアスをかけたらどうなるのか、ということがあります。残念ながら答えは、「攻撃者がこれを有効に活用する」になります。

上記の例を続けると、コンポーネントの攻撃者は、価格 1 と 6.1、もしくは同等の価格を選択します。買い手は 1 と組み合わせた場合に 6.1 しか購入できないため、攻撃が成功する確率は $\frac{3}{7}$ になります。

ペアごとに正規化されコストの無作為抽出

上記の質問に似たものとして、無作為抽出にバイアスをかけ、売り手の*ペア*が選択される確率を正規化したらどうなるか、と問うこともできます。どのペアが稀であるかを制御することで、攻撃者が攻撃に成功する可能性がここでも高まります。

上記の例を続けると、コンポーネントの攻撃者は再び 1 と 6.1 を価格、または価格と同等なものとして選択すると、6.1 を含むペアは非常に少ないため、今回は成功率が $\frac{24}{49}$ となります。

A.5. 安定性分析

候補となるアプローチのいくつかの概要を説明し終えたところで、ある疑問が湧いてきます。買い手は所定の戦略から離れる動機を持つのでしょうか? そうだとすると、攻撃者はさまざまな買い手を利用しようとするので、それぞれの戦略のセキュリティ属性にどう影響するのでしょうか?

分析のための分析を避けるため、価格設定とセキュリティの両方の観点から準最適/不安定な戦略については省略しました。

最低価格

最低価格は、選択された価格がすでに可能な限り低いため、経済的インセンティブに対して安定してい

ます。ただし、セキュリティの観点からは準最適であって極めて不安定です。セキュリティを重視する買い手は、異なる戦略、おそらくコスト次第の無作為抽出を選択するでしょう。

両グループのセキュリティ上の利点について次のセクションで説明するように、これによって、攻撃者がこれら 2 種類の買い手間でリソースをどのように割り当てるかを考えざるを得ないという興味深い状況が発生します。

コスト次第の無作為抽出

前の議論の逆として、この戦略はセキュリティの観点からは安定していますが、経済的インセンティブについては安定していません。セキュリティに興味のない買い手は最低コストの選択を採用するでしょう。

中には、この戦略がセキュリティの観点から安定しているという主張に驚く読者もいるかもしれません。おそらく、オーキッド マーケットの買い手の中には、攻撃者がコスト次第の無作為抽出を悪用する知識を利用して、独自に改良した選択方法を作成する人がいるでしょう。前述のように、オーキッド マーケットは購入戦略の推論に対して安全ではなく、さらに厄介なことに、選択した代替方法を攻撃者が推測した可能性を知る方法がありません。したがって、十分に偏執的な買い手にとっては、この方法は最悪の仮定の下で最高のパフォーマンスを発揮するため、安定した方法と言えます。

ただし、オーキッド マーケットの一部の買い手がこのアドバイスを無視したり、単に最低コストの選択方法を採用したりする限り、セキュリティを重視する買い手にとってこれは朗報です。二次的な攻撃の最適化は、攻撃者によるコスト次第の無作為抽出の最適な活用を妨げるだけとなります。

A.6. 経済的適合性分析

ここで、売り手の戦略の問題を見てみましょう。ここでの目標は、売り手が通常の種類の経済的アルゴリズムを採用できる範囲を示すことです。

価格設定とセキュリティの両方の観点から準最適/不安定な戦略については、ここでは省略しました。

最低価格

このアプローチは経済において当然予想されるケースであるため、経済的には完全に適合していると言えます。

コスト次第の無作為抽出

この戦略に経済的適合性があると最初は思えないかもしれませんが、最高価格を共有していない買い手の集団を考慮すると、売り手の商品に対する関心の頻度は価格に対する従来の感応度に従うと考えられます。

売り手は通常の方法で価格を上げ下げすることで受け取るオーダーの数を調整できるため、コスト次第の無作為抽出には経済的適合性があります。

A.7. 結論

ここまで、オーキッド マーケットの買い手に適したオークション方法の分析に一通り目を通し、それによって、全般的な使用に向けてコスト次第の無作為抽出をどのように選定したかを示しました。

「純粋に無作為」なアプローチを選択した理由は、攻撃者が買い手の戦略を十分に理解しているという仮定と、正当な売り手が価格を選択した後に攻撃者が価格を選択するという仮定に基づいています。ここではバイアスのかかったサンプルにできる最良のことは何もしないことですが、最悪の場合、攻撃者は選択の機会を増やすことができます。サンプルにバイアスをかけるのではなく、利用可能なオプションの数を最大限に増やし、そこから均一に選択しました。

従来のオークション モデルに比べて、買い手にとってのこうしたコストの増加を心配している読者には、プレミアムを「セキュリティの価格」と見なすようお勧めします。これまで見てきたように、自らを攻撃に広くさらすことによって低価格を実現することは些細な事柄と言えます。

B. 攻撃とセキュリティ

B.1. チェーンに対する共謀攻撃

このセクションでは、複数のリレーやインターネット サービス プロバイダー (ISP) を制御または監視する攻撃者によって、どういう種類の情報が推論または推測されるのかを検討します。リレーとプロキシが無作為に選択される (したがって ISP も選択される) ことを仮定して、特定の攻撃がさまざまなチェーン長で実行される可能性の確率モデルを構築します。

個別のリレーやプロキシで利用可能な情報

IP ベースのネットワーキングの固有の構造、およびオーキッドのプロトコルがイーサリアムベースの支払いを利用していることにより、リレーとプロキシ ノード、*およびその IPS* が次の情報にアクセスします。

- 接続先のすべてのコンピューターの IP アドレス。
- 転送するパケットのサイズ、タイミングと数。
- それらに支払うトークンを制御する公開鍵。
- それらに向けられた制御セグメントの内容。

さらに、プロキシ ノード*とその ISP* が次の情報にアクセスします。

- Web サーバーのホスト名、および SSL/TLS セッションのネゴシエーションのプランテキスト部分。

潜在的な共謀の当事者

次のロールはカスタマー情報にアクセスできるため、攻撃の一環として意味のある共謀や監視が行われる可能性があります。

- カスタマーリレー、プロキシ、または Web サーバーのインターネット サービス プロバイダ(ISP)。s の確率で信頼できない。
- Web サイト。プロキシが接続されている Web サーバー。w の確率で信頼できない。
- リレー$_n$。チェーン内の n 番目のリレー。$\frac{r}{n}$ の確率で信頼できない。
- プロキシ。帯域幅を Web サーバーに中継するプロキシ。$\frac{x}{n}$ の確率で信頼できない。

私たちは、上記の r と x を分離しました。これは、攻撃者は、プルーフオブワーク計算に利用できる計算の総量を制御することはできませんが、リレーとプロキシ ノード間で計算をどのように割り当てるかは制御できるためです。

攻撃の種類

共謀攻撃の中心的な目標は、オーキッドの特定カスタマーと特定の SSL 接続をリンクすることです。これを行うには 2 つの方法があります。

- リレーション。これが可能な場合、攻撃者はカスタマーが特定の Web サイトと対話していると推測できます。これは、ルートに沿って十分なポイントを監視できるためです。

- タイミング。これが可能な場合、攻撃者は、パケットのタイミングを制御して監視することにより、カスタマーが特定の Web サイト と対話していると推測できます。
- 未消費。これが可能な場合、攻撃者は帯域幅の消費をカスタマーが採用しているにもかかわらず、タイミング攻撃を実行できます。

通常のインターネット アクセス: ゼロ リレーおよびゼロ プロキシ

カスタマーが Web サイトに直接接続する場合、オーキッド システムはもちろん使用されませんが、私たちはこのセットアップにどのような情報リスクが存在するかを検証して、残りの分析の根拠とすることが重要であると感じています。

ISP	Web サイト	P (リレーション)	P (タイミング)	P (未消費)
x		s		
	x	w		

上記の表で、「X」は共謀への参加を示し、P (リレーション) と P (タイミング) の値はこの発生の可能性を示します。攻撃が不可能な行は、余分な「X」のある行と同様に省略され、より単純な攻撃が可能なところでのより洗練された攻撃について言及されています。

VPN: ゼロ リレーおよびゼロ プロキシ

分析の根拠として、VPN アクセスに固有の共謀リスクも示します。

ISP	VPN	Web サイト	P (リレーション)	P (タイミング)	P (未消費)
	x		g		
x		x	sw		

ここで g は、VPN プロバイダーが監視されているか、または敵と共謀している可能性を表します。g が、VPN を使用した結果としてなど、モデル化が困難な方法で時間とともに変化する場合があることに留意してください。

ゼロ リレー、1 つのプロキシ

ISP	プロキシ	Web サイト	P (リレーション)	P (タイミング)	P (未消費)
	x		$\left(\frac{x}{n}\right)$		
x		x	sw		

この場合のリスクが VPN 利用のリスクと非常に似ていることは意外ではありません。リレーを使用しないチェーンは、各ブラウジング セッションの前に新しい VPN プロバイダーに無作為に選択され、VPN プロバイダーによって個人情報が保存されない VPN と同等です。

1 つのリレー、1 つのプロキシ

ISP	リレー 1	プロキシ	Web サイト	P (リレーション)	P (タイミング)	P (未消費)
	x	x		$\left(\frac{rx}{n^2}\right)$		
	x		x	$w\left(\frac{r}{n}\right)$		
x		x		$s\left(\frac{x}{n}\right)$		
x			x		sw	

この構成で帯域幅の消費が採用されている場合、あらゆるタイミング攻撃が軽減されます。タイミングケースにリレー 1 またはプロキシを追加すると、リレーションが許可されることに注意してください。

2 つのリレー、1 つのプロキシ

ISP	リレー 1	リレー 2	プロキシ	Web サイト	P (リレーション)	P (タイミング)	P (未消費)
	x		x		$\left(\frac{rx}{n^2}\right)$		
x		x	x		$s\left(\frac{rx}{n^2}\right)$		
	x	x		x	$s\left(\frac{r}{n}\right)^2$		
x		x		x	$sw\left(\frac{r}{n}\right)$		
	x			x		$s\left(\frac{r}{n}\right)$	
x				x		sw	

この構成で帯域幅の消費が採用されている場合、あらゆるタイミング攻撃が軽減されます。リレー 1 と Web サイトによって実行されるタイミング攻撃の場合、リレー 2 またはプロキシを共謀に追加するとリレーションとなります。カスタマーの ISP が Web サイトと共謀している場合、リレー 2 を追加するとリレーションとなります。

3 つのリレー、1 つのプロキシ

ISP	リレー 1	リレー 2	リレー 3	プロキシ	Web サイト	P (リレーション)	P (タイミング)	P (未消費)
	x	x		x		$\left(\frac{r^2x}{n^3}\right)$		
	x		x	x		$\left(\frac{r^2x}{n^3}\right)$		
x		x		x		$s\left(\frac{rx}{n^2}\right)$		
	x		x		x	$w\left(\frac{r}{n}\right)^2$		
x		x	x		x	$sw\left(\frac{r}{n}\right)^2$		
	x				x		$s\left(\frac{r}{n}\right)$	
x					x		sw	
	x	x			x			$s\left(\frac{r}{n}\right)^2$
x		x			x			$sw\left(\frac{r}{n}\right)$

B.2. SSL および TLS の脆弱性

SSL と TLS は複雑なプロトコルであり、実装の欠陥が発見されるたびに一定のセキュリティ アップデートが適用されます。残念ながら、ユーザーがソフトウェアのアップグレードを遅らせたり、信頼できないソフトウェアや作成が不十分なソフトウェアを使用したり、ソフトウェアの設定を間違えたりすることがあります。ユーザーを可能な限り保護するために、オーキッドのプロトコルは「健全性チェック」機能を提供します。

SSL ダウングレード攻撃

いわゆる *SSL ダウングレード攻撃*では、攻撃者は安全な接続に低品質の暗号化を使用させます([21])。この攻撃を実行するために、攻撃者はクライアントがサポートするより安全な暗号化方式の記述を初期キー ネゴシエーション パケットから単に削除します。この攻撃を防ぐために、オーキッドのクライアントは可能な限り自動的に逆の処理を行います。キー ネゴシエーション パケットから安全でないオプションの記述を削除します (「SSL アップグレード」攻撃)。

旧式のブラウザと電話アプリ

SSL と TLS のセキュリティの脆弱性は Web ブラウザで定期的に検出され、パッチが適用されます。ただし、すべてのユーザーが最新のブラウザを使用すると想定することはできません。開発者が SSL 証明書の検証などを省略するなど、携帯電話アプリでも同様の状況が発生します。

これらの問題に対処するため、オーキッドのクライアントは、Google Chrome で使用されるオープンソース SSL ライブラリである「Boring SSL」の最新コピーを使用して、証明書チェーンを自動的に検証します。

B.3. ファイアウォール迂回機能

上記のシステムは、インターネットへの無料でオープンなアクセスをすでに所有しているユーザーのみが利用できるとすれば、ほとんど役に立たないでしょう。このセクションでは、攻撃者が提供しているインターネット アクセスを利用するユーザーのアクセスを容易にする機能について説明します。

攻撃者がすべてのインターネット アクセスを完全にブロックした場合は、この領域での防御は不可能になることに注意してください。したがって、このセクションのすべての防御分析では、攻撃者がブランケット ブロッキングにある程度のコストを被るため、高コストの攻撃は実行されないことを期待して、このコストの最大化を図ることを想定しています。

ブートストラップ

ファイアウォール プロバイダーがオーキッド ネットワークに対して試みる最初の攻撃の 1 つは、エントリー ペドラーのリストを作成し、それらへのすべてのアクセスをブロックすることです。これは、カスタマーがエントリー ペドラーにアクセスできない場合、ネットワークを使用できないためです。問題を複雑にしているのは、有能な攻撃者はカスタマーが利用できる IP アドレスのリストを持っていると想定せざるを得ないことです。

最初にこれに対処するために、ユーザーはプルーフオブワークと引き換えに新しいリレー IP アドレスを取得できるサービスを提供します。ブートストラップ自体の前後のブロックを妨げるために、Web、メ

ール、および一般的なインスタント メッセージング プラットフォームを介して、このブーストラッピング サービスへのアクセスを提供します。ユーザーは、チャレンジをクライアントのオプション画面から最も便利な通信メカニズムにコピー & ペーストしてから、返信をクライアントにコピー & ペーストします。

DPI、推論、およびアクティブ プローブ

より洗練されたファイアウォールは、ディープ パケット インスペクション (DPI - ヘッダーだけでなくパケットのコンテンツの分析)、タイミングの推測 (パケット サイズ、量、およびタイミングに関する総計の統計指標の使用)、アクティブ プローブなどの方法を採用しています (提供されているサービスを識別するために、ユーザーまたはユーザーが接続しているサーバーとの接続を試みます。)

私たちは、重要な情報を提供するためにディープ パケット インスペクションやアクティブ プローブを使用することは考えていません。WebRTC の使用によってすべての通信は暗号化されており、アクティブな WebRTC オファーが発行されない限り、オープン ポートはありません。これは WebRTC の他のすべての使用法の動作と同じであるため、この動作によってはオーキッド ユーザーが明らかになることはありません。

ただし、暗号化されたストリーム上の Web リクエストのタイミングとサイズは、他の種類の WebRTC トラフィックとは異なって見えるため、タイミング推論を利用すれば、オーキッド ユーザーを効果的に検出できる可能性はあります ([62])。これに対処するため、推論攻撃が発生する可能性が高い状況でオーキッド ネットワークにアクセスするユーザーは、「帯域幅の消費」を使用することをお勧めします (セクション 7)。

開示。イーサリアム トラフィック

現在のクライアントは支払いステータスを追跡するためにイーサリアム クライアントを使用しており、イーサリアムには独自の、強化されていないネットワーキング署名があります。そのため、このイーサリアム トラフィックとの関連が検出されやすい弱点となる可能性があります。ファイアウォールのオペレーターは、単に「イーサリアムを実行しているコンピューターであるか?」と「大量の WebRTC トラフィックを消費しているか?」ということのみを尋ねるでしょう。

プロジェクトの焦点を維持するために、イーサリアムの強化やオーキッド ネットワーク上でのイーサリアム トラフィックの提供は、最初のリリースの機能としては提供していません。この問題については、今後のバージョンで対処する予定です。セクション 10 参照してください。

B.4. 攻撃分析と攻撃者のユーザー事例

抑圧的な Web アプリケーション

攻撃者の目標。すべてのオーキッド リレーおよびプロキシ IP アドレスの特定

オーキッド マーケットにはすべてのリレーとプロキシが含まれているため、これはセクション B.3 で説明したものの逆の攻撃です。

オーキッド マーケットでの強制接続の数は、$O(log(n))$ で増加します。ここで、n はネットワークのサイズです。攻撃者がグローバルな計算の $m\%$ を保持している場合、プルーフオブワークを完了させるたびに $log(n)$ の IP アドレスを知ることになります。したがって攻撃者は、c エポックにおいて、リレーとプロキシ IP アドレスの $1 - (1 - \frac{\log(n)}{n})c$ パーセントを知ることになります。

Tor のトラフィックのブロックがどのように成功するかをよく知っている読者は、上記がシステムの深刻な問題にならないかと心配するかもしれません。幸いなことに、そうはなりません。Tor には約 1,000 の出口ノードがあるため、簡単にフィルタリングできます。私たちの場合、主にホワイトリストのサポートがあるため、数十万の出口ノードがあると予想されます。これによって、上記の方法を使用したマスク解除にはるかに大きな計算課題が必要になることに加え、これらの IP アドレスをブロックすると、抑圧的な Web アプリケーション自身のユーザーをもブロックすることになります。

企業ネットワークと「優れた」ファイアウォール

攻撃者の目標。インターネット アクセスの提供先であるユーザーがオーキッド ネットワークを使用できないようにする

これに関連する機能については、セクション B.3 で詳しく説明します。この攻撃者が成功する見通しは低いと言えます。オーキッド ネットワークの使用量は、一定量のデータを中継する WebRTC 接続として表されます。プローブが可能なオープン ポートはなく、それらを「出力」するために信頼できる IP アドレスもありません。

受動的な監視と推論 (おそらくシビル攻撃によるもの)

攻撃者の目標。カスタマー IP の識別および Web サイトの識別

このクラスの攻撃に関連する分析については、セクション B.1 で詳しく説明します。この攻撃者が成功する見通しは低いと言えます。チェーンのいくつかの位置に自分自身を配置するには、グローバルな計算能力の大部分を所有する (そしてこの攻撃に専念する) 必要があるため、非常に困難です。

短時間のトローリングと QoS 攻撃

攻撃者の目標。可能な限り多くのネットワークで騒乱を引き起こすこと

このドメインでの分析の実行は、セキュリティ志向の読者にとって楽しいものとなるでしょう。ここでのタスクは、限られた予算 (おそらく 10,000 米ドル程度) を与えられ、ネットワークを可能な限り混乱させることです。

攻撃チェーン - 攻撃者はここでさまざまな攻撃を試みることができます。たとえば、パケットを無作為にドロップする、非常に遅いサービスのみを提供する、断続的に遅いサービスを提供する、単純に切断す

るなどです。いずれの場合も、カスタマーが単に問題のノードを交換するだけで、すべてのカスタマーが軽微な不便を被ります。パケットをドロップする場合には、他のリレーにさらなる不便が生じる可能性があります。B がパケットを受信していないことについて嘘をついている場合、A がそのパケットを転送しなかったかどうかを判断する方法がないからです。この場合、カスタマーは両方のリレーを交換します。

オーキッド マーケットへの攻撃 - 攻撃者は、この状況に関しても同様に多くの選択肢を持っています。

- 参加プロトコルを不適切に実装する。この場合、「攻撃者」は単にパケット転送のために他のペドラーに支払いをしているため、これを攻撃とはみなしません。
- オーキッド マーケットに参加するが、ユーザーへのルーティング表情報の提供を拒否する、またはパケットの転送を拒否する。これにより、問題のペドラーがネットワークから切断されるまで、オーキッド マーケットの $\frac{log(n)}{n}$ のクエリに追加のルーティング負荷が発生します。
- オーキッド マーケットに居座ってサービスを提供しない。この場合、カスタマーが実行するオークションの効率が低下します ($\frac{log(n)}{n}$ 時間について 1 人の参加者が失われるため)。

C. メダリオンのエンジニアリング仕様

セクション 5.3 のメダリオンの仕様の概要に基づいて、この付属資料では、メダリオンとその生成に関するより正確な定義を説明します。この付属資料は [50] で使用されている表記法に基づいていることに留意してください。以下のリストはタイプ別に分類されています。

t (uint) – 正確な UNIX 時間 p。精度は 100ms 以上でなければならない

skm (uint) – 秘密鍵 x。無作為に選択されたもの

e_t(uint) – イーサリアム ブロック ダイジェスト (別名ブロック ハッシュ)。t 時点でのもの

$h(y)$ (uint) – y を入力した場合の Keccak のダイジェスト

$seed$ (uint) – $h(e_t / sig(sk, e_t))$

pk_m (tuple) – 公開鍵 $x * G$、楕円曲線 C 上で、ベースポイントは G、順序は N

$sig(sk, r)$ (tuple) – r の ECDSA 署名。秘密鍵 sk を使用したもの

$F_{n,k}(x)$ (struct) – $\{n, k, x, i_0, ..., i_{2^k}\} : n, k, x, i_j \in \mathrm{Z}/h/$

M (struct) – $\{t, e_t, pk_m, sig(sk, e_t), F_{n,k}(seed)\}$

メダリオンのアルゴリズム

メダリオンを生成するには、2 つの方法が提案されています。1 つ目は非対話型です。この場合、メダリオンの生成には、現在のイーサリアム ブロックダ イジェストと公開鍵ペアが必要です。以下に示す Equihash のプルーフオブワークは簡略化されたものです。

アルゴリズム 1: 非対話型のメダリオン生成

準備手順

$sk \leftarrow \mathrm{random} \in \mathrm{Z}/h/$
$pk \leftarrow sk * G$
$sig(sk, e_t) \leftarrow \mathrm{ECDSA}(sk, e_t)$
$seed \leftarrow h(e_t, sig(sk, e_t))$

Equihash のプルーフオブワーク

set difficulty (n, k)
set counter $i_1 = seed$
set $\{i_j\}$ of 2^k items
while $\{h(i1) \oplus h(i_2) \oplus ... \oplus h(i_{2^k}) \neq 0\}${
build bigger list of $\{i_j\}$
find subsets of colliding $\{i_j\}$
sort $\{h(i_j)\}$
}

戻り値: $t, e_t, pk, sig(sk, e_t), \{i_j\}$

2 番目の方法は最初の方法に基づいており、オーキッド マーケット内のペドラーがメダリオンの構築に参加するための要件を追加します。オーキッド マーケットからの参加を要求することにより、チャレンジ チャレンジ-レスポンスタイプのプロトコルが作成されます。これは原始的なプルーフオブタイムに例えることができます。

このスキームには、m 人の登場人物がいます。メダリオンの生成者である *Alice (アリス)* と、オーキッ

ド マーケットのペドラーのコミュニティです。このコミュニティは $pi \in \{Bob, Chris, Dana, ...\}$ と表されます。*アリス*はエントリ ペドラーである *Bob (ボブ)* と対話し、ボブは $sig(skBob, e)$ を計算して、その署名を*アリス*に返します。オーキッド マーケットがさらに多くの参加者を必要とする場合、*ボブ*は m 人の無作為なペドラーに連絡し、それらのペドラーは $sig(sk_{pi}, e)$ を、*ボブ*を通して*アリス*に返します。*アリス*は次に $seed$ を計算します。その際に、$\{sig(sk_{pi}, e)\}$ を追加の入力値として使用し、得た値 M を*ボブ*に返します。

アルゴリズム 2: レスポンス指向のメダリオン生成

準備手順

$sk \leftarrow \text{random} \in Z_{|h|}$
$pk \leftarrow sk * G$
$sig(sk, e_t) \leftarrow \text{ECDSA}(sk, e_t)$

インタラクティブなステップ

プルーフオブタイムを実行する
$seed \leftarrow h(et, \{sig(sk_{pi}, e_t))\}$

Equihash のプルーフオブワーク

set difficulty (n, k)
set counter $i1 = seed$
set $\{i_j\}$ of 2_k items
while $\{h(i_1) \oplus h(i_2) \oplus ... \oplus h(i_2k) \neq 0\}\{$
build bigger list of $\{i_j\}$
find subsets of colliding $\{i_j\}$
sort $\{h(i_j)\}$
$\}$

戻り値: $t, e_t, pk, sig(sk, e_t), \{i_j\}$

D. 支払いプロトコルと定義

D.1. 支払いチケットの暗号化の選択

私たちは、オーキッド マイクロペイメントのコストを削減するために、他の暗号化機能よりもイーサリアム ガスのコストが少ない、特定の暗号化機能を選択しました。

h (**Keccak-256**) – オーキッド プロトコルでは、任意の安全な暗号化ハッシュ関数を使用することができます。ただし、私たちのシステムでは、EVM で使用可能なすべてのハッシュ関数の中でガス コストが最も低い Keccack を選択しました[4]。この選択は、オーキッドのトランザクション コストをさらに最小限に抑えるためです。

sig(*sk, r*) (**ECDSA**) – Keccak-256 の secp256k1 曲線を内部の暗号化ハッシュ関数として使用します。繰り返しますが、この選択が行われたのは、EVM は ECDSA のサポートのために構築されたものであり、オーキッドのガス コストを削減する必要があったからです。さらに、選択された曲線は、既存のブロックチェーン ソフトウェア ライブラリおよびツールと互換性があります。

D.2. 支払いチケットの定義

オーキッドの支払いチケットには以下のフィールドがあります。

h (**function**) – 暗号化ハッシュ関数

timestamp (**uint32**) – 値が減少し始める時刻を示す

recipient (**uint160**) – チケットの受信者の 160 ビット イーサリアム アカウント アドレス

rand (**uint256**) – *受信者*が選択した無作為な整数

nonce (**uint256**) – チケット送信者が選択した無作為な整数

faceValue (**uint256**) – 当選チケットの価値

marketValue (**uint256**) – 帯域幅市場に基づくチケットの期待価格

acceptedValue (**uint256**) – 受信者が受け入れる内容に基づくチケットの期待価格

winProb (**uint256**) – 特定のチケットが送信者から *faceValue* を勝ち取る確率

randHash (**uint256**) – *h*(*rand*)のダイジェスト

ticketHash (**uint256**) – *h*(*randHash, recipient, faceValue, winProb, nonce*)のダイジェスト

(*v*1, *r*1, *s*1) (**tuple**) – チケット送信者の ECDSA 署名要素

(*v*2, *r*2, *s*2) (**tuple**) – *受信者*の ECDSA 署名要素

D.3. 支払いチケットの生成

*アリス*を受信者、*ボブ*を送信者と想定してみましょう。

1. *アリス*は無作為な 256 ビット数と *rand* を選択し、*randHash* を計算して、そのダイジェストを*ボ*

[4] 32 バイトのハッシュに対して 36 ガスのイーサリアム コスト [89]

*ブ*に送信する

2. *ボブ*は (*nonce, faceV alue, winProb, recipient*) 5の値を選択する
3. *ボブ*は *ticketHash* を計算する
4. *ボブ*は Sig(PrivKey, *ticketHash*) を計算する
5. その結果のチケットは、以下のように構成されます。

   (a) *randHash*
   (b) *recipient*
   (c) *faceV alue*
   (d) *winProb*
   (e) *nonce*
   (f) *ticketHash*
   (g) *creator* (*ticketHash* を署名した送信者の鍵のアドレス)
   (h) *creatorSig* (*ticketHash* 上の送信者の署名)

このチケットは受信者が完全に検証できるという意味で有効ですが、オーキッド ペイメントのイーサリアム スマート コントラクトで請求できるようにするには、受信者が署名する必要があります (以下を参照してください)。

D.4. 支払いチケットの検証

アリス (帯域幅販売者) は次の操作を実行します。

検証

(a) $randHash = H(rand)$
(b) $faceV alue \geq marketV alue$
(c) $winProb \geq acceptedV alue$
(d) *recipient* = {受信者によって公開されたイーサリアム アカウント アドレス}
(e) *creator* = {送信者によって公開されたイーサリアム アカウント アドレス}

バリデーション

(a) 次のことをバリデーションする: *creatorSig* は秘密鍵によって署名されており、その秘密鍵の所有者の公開鍵が作成者のアドレスであること

確認

(a) 次のことをバリデーションする: *creator* は、オーキッド ペイメント スマート コントラクトにロックインされた十分なオーキッド トークンを所有している

アサーション

(a) チケットは現在有効であることが証明されており、当選チケットである可能性がある

[5]次のようなこの仕様の情報を使用します。*受信者*によって署名された一般的な帯域幅市場データと公開機能

D.5. チケットから支払いを請求する

受信者は、チケットが有効であるか、および当選チケットであるか否かについてはローカルで完全に確認できますが、当選チケットのトークンの実際の支払いは、オーキッド ペイメントのスマート コントラクトによって行う必要があります。オーキッドのスマート コントラクトは、入力として以下を受け取る Solidity API を公開します。

1. *rand*
2. *nonce*
3. *faceV alue*
4. *winProb*
5. *receipient*
6. *recipientSig*（*ticketHash* 上の*受信者*の署名）
7. *creatorSig*（*ticketHash* 上の送信者の署名）

D.5.1. スマート コントラクトの実行

仮に*アリス*が帯域幅を購入するユーザーだとします。*アリス*には、イーサリアムのアカウント アドレス、*addressAlice* と、オーキッドのトークンが必要です。このアドレスには公開鍵、*PubKeyAlice* が関連付けられていることに注意してください。前のセクションで説明したように、アリスは、オーキッド トークンをイーサリアムのスマート コントラクトに*ロックアップ*し、*PubKeyAlice* でロックする必要もあります。前のセクションでは、*アリスの*アドレスはイーサリアムのアカウント アドレスで、*ticketHash* 上で *creatorSig* から復元された公開鍵と同じとされています。

ここでは、FALSE とされる一時的なブール値として SLASH を設定し、*PubKey* を *ticketHash* 上で *recipientSig* から復元された公開鍵とします。

計算

(a) *ticketHash*

検証

(a) *randHash*、そうでない場合は実行を中止[6]。

(b) *PubKey* = *受信者*のアドレス、そうでない場合は実行を中止。

(c) *addressAlice* には、ペナルティ エスクロー アカウントにロックされているオーキッド トークンがある。そうでない場合は実行を中止。

(d) *addressAlice* には、チケット アカウントにロックされている、チケットの支払いに十分なオーキッド トークンがある。そうでない場合は、SLASH を TRUE に設定して実行を継続。

(e) *H*(*ticketHash, rand*)7 ≤ *winProb*.そうでない場合は実行を中止。

判断

(a) SLASH = FALSE の場合、チケットは支払われる。*faceV alue* が作成者のチケット資金から *recipient* に送金される。

[6]トランザクションが中止されたため、イーサリアムの状態遷移は発生せず、ガスは消費されません
[7]uint256 として解釈されます

(b) SLASH = TRUE の場合、作成者はスラッシュされる。

決済

(a) 作成者のチケット資金がある場合は、*recipient* に送金する (これは、*faceV alue* よりも小さいことを保証する事前の検証によるもの)。

(b) 作成者のペナルティ エスクロー アカウントをゼロに設定する (これらのトークンをバーン/スラッシュする)。

ペナルティ エスクローを大幅に削減することで、チケット送信者が二重支出から得られる以上の損失をもたらすことにより、二重支出の意欲を削ぎ、二重支出を防止できますが、それでもチケット送信者が大規模な過剰支出を行う危険は存在します。これに対処するには、当選チケットの価値が *timestamp (タイムスタンプ)* によって指数関数的に減少し始めるようにします。これは当選者にとって直ちに現金化することを促す強力なインセンティブになります。受信者はこの即時性を利用して、送信者のオーキッド トークンの残高の「浪費率」を計算できます。

E.　支払いに関する追加の詳細

E.1. パケットにかかる費用

議論のために、平均パケットの長さが 1KB であると仮定しましょう。パケットの平均総コストの合理的な上限を計算してみます。クラウド サービス業界で最も高価な帯域幅プロバイダーの 1 つは、Amazon Web Services のシンガポール CloudFront で、1×10^9 バイトあたり 0.14 ドルが請求されます。この場合、パケットごとのコストは 1.4×10^{-5} セント (0.00000014 ドル) になります。

帯域幅はほとんどのユーザーにとって無駄であるため (売れ残りの帯域幅は永久に失われます)、オーキッド マーケットの帯域幅の実際の価格はこの上限よりも大幅に低くなると考えられます。

E.2. イーサリアムのトランザクション コスト

イーサリアム スマート コントラクトは、イーサリアム仮想マシン [89] (EVM) のパワーと柔軟性を利用して、洗練された支払いメカニズムの作成を可能にします。このマシンはチューリングの完全な実行環境を (経済的に可能な範囲内で) 提供します。イーサリアム スマート コントラクトによって実行されるインストラクションのコストは、トランザクションを組成するトランザクション料金に加算されます。

EVM の各インストラクションにはコストとしてある程度の量のガスがかかるため、イーサリアムのトランザクション料金は、トランザクションに費やされるガスの合計に送信者が設定したガス価格を掛けたものとして定義されます。マイナーはマイニングされたブロックに含める有効なトランザクションを選択しますが、その際に任意のガス価格 (ゼロを含む) のトランザクションを含めることができます。各ブロックに含めることができるトランザクション数には制限があるため、ガス価格の高いトランザクションを選択すると、利益が増える可能性があります。同様に、より低いガス価格を受け入れると、ネットワークが最大容量で稼働していない場合でもマイナーがブロックを埋めることができるため、より多くの利益につながる可能性があります。このメカニズムによって、絶えず変化しながらも安定したゲーム理論的平衡が生まれます。この状況は、イーサリアム ガス ステーション [22] などのサイトによって追跡されます。

2017 年 10 月現在、高い確率で数ブロック内にトランザクションを含めるためのコストは 0.026 ドルです。15 分以内に確認するには、0.006 ドルで十分です。これらの見積もりはトランザクションの基本コストです。スマート コントラクト コードを実行しないプレーン イーサ送金のコストは 21,000 ガスです。トランザクションがスマート コントラクト コードを実行する場合、EVM のインストラクションごとに追加のガス コストがかかります。たとえば、新たな 256 ビット値をスマート コントラクト ストレージに永続的に保存するには 20,000 ガス、既存の値を更新するには 5,000 ガスがかかります。

イーサリアムの ERC20 元帳はアカウント アドレスと残高のマッピングにすぎないため、ERC20 トークンの転送には 21,000 + 20,000 ガスのオーダーが必要であり、その後の転送には 21,000 + 5,000 ガスが必要です (受信者のアカウントがその時点ですでにトークン台帳にエントリしているため)。ライブ [23] の ERC20 トランザクションを見ると、新規および既存のアカウントへの転送のガス費用はそれぞれ約 52,000 および 37,000 で、やや高いことがわかります。その差額は、送信者が十分な残高を持っているかどうかなどの不変条件の検証を実行するスマート コントラクト コード、および支払い領収書のログなどの他の実装の詳細によって説明できます。トランザクションを確認する速度に応じて、50,000 ガスで 0.014 ドルから 0.062 ドルまでのトランザクション料金が必要になります。

E.3. パフォーマンス

オーキッド ペイメントのスマート コントラクトは不変ですが、新しいコントラクトを展開してオーキッド クラ

イアント ソフトウェアをアップグレードすることで、オーキッド ペイメントを効果的にアップグレードすることが可能です（必要に応じて古いコントラクトとの下位互換性を維持します）。イーサリアム スマート コントラクトは、ガス コストを削減するためのさまざまな最適化をサポートしています。オーキッド ペイメント スマート コントラクトの将来のバージョンでは、たとえばインライン ソリディティ アセンブリ [24] を使用してガス コストを最適化する予定です。これはしばしば、通常のソフトウェア システムが、高価なサブルーチンをインライン アセンブリで置き換えるのと同様です。

ただし、オーキッドの支払いチケットの検証では、ECDSA によるリカバリなどの暗号化操作と、オーキッド トークン台帳における送信者と受信者のエントリの更新がボトルネックとなります。この改善策としては、スマート コントラクトの API コール ペイロードを運ぶイーサリアム トランザクションの受信者の署名を、チケット データ構造をカバーする署名として二重に使用することが考えられます。現在、オーキッド スキームは、オーキッド クライアントの柔軟性のため、そしてイーサリアムの仕様に依存せずに支払いスキームの指定と推論を容易にするために、2 つの署名を定義しています。より単純な最適化には、チケット フィールドを密にパッキングして、複数の内部変数を単一の 256 ビット ワードにエンコードし、EVM スタック ワードおよび恒久的なコントラクト ストレージ スロット（両方ともサイズは 256 ビット）に一致するよう調整する方法などがあります。

一方、匿名性を高めるには、ミキシング テクノロジーの使用が任意または必須となり、オーキッド ペイメントのガス コストを大幅に増加させる可能性があります。リンク可能なリング署名に基づくミキシング サービスを使用すると、トランザクション料金が簡単に 1 桁高くなる可能性があります [20]。ただし、強力な匿名性の保証が提供される場合、ユーザーはこれに価値があると考えるかもしれません。オーキッドの支払いの確率変数（チケットの頻度、当選確率、および当選額）については簡単に調整できるため、チケット請求間の平均時間を調整してトランザクション料金を削減できます（特に、平均で数日毎に支払いが必要なだけの長時間実行ノードの場合）。

最後に、zk-SNARKs などのゼロ知識テクノロジーには、イーサリアム スマート コントラクトの実行など、任意の計算のオーバーヘッドを大幅に削減するという非常に興味深い特性があります [25]。zk-SNARK 証明の生成は高価ですが、検証は元のコードと比較しても安価です。チェーン上で実行する必要があるのは検証のみであるため、オーキッド チケットを請求するゼロ知識の証明による検証は、元の検証コードを使用するよりも安価になります。

さらに進むと、再帰的な SNARK [51] によって、一連の SNARK 証明を 1 つの証明に集約できる可能性があります。これはブロックチェーン コンセンサス プロトコル [26] の方により適用しやすいかもしれませんが、オーキッドにとっても、たとえば、線形ガス コストの複雑さを回避しながら、複数のチケット請求を単一のスマート コントラクト トランザクションにバッチ処理するのに役立ちます。

E.4. マクロペイメントからマイクロペイメントを構築する

トランザクション コストと支払いトークンの選択について説明したので、次に実行可能な支払い方法を見てみましょう。ブロックチェーンベースのマイクロペイメントの基本的な課題の 1 つは、トランザクション料金を回避する方法です。1 セントを多くの回数にわたって送信する状況を想像してみてください。各 1 セントを単純なイーサリアム ERC20 トランザクションとして送信する場合、各支払いに対して 1.4 セント、すなわち 140% のトランザクション料金を支払うことになります。マイクロペイメントを効果的なものにするためには、トランザクション料金を数桁下げる必要があります。

MojoNation [27] で採用された潜在的に興味深いアプローチの 1 つは、ノードの各ペア間で「取引のバランス」を取ることです。この方法では、各ペア間に帯域幅が流れる際、バランスがゼロから離れすぎると定期的に清算されます。ただし、これまで見てきたように、プレーンなイーサリアム トランザクションを使用して支払いを決済する場合のトランザクション コストは、少なくとも 0.014 ドルになります。前

述の上限に基づくと、この価格は約 140 メガバイトの帯域幅に等しいことがわかります。このアプローチの 2 番目の問題は、調整のしきい値に近づいたピアがその事実を認識し、料金を支払うのではなく、切断して新しい ID を作成しようとすることです。

E.5. 支払いチャネル

ビットコイン ネットワークで最初に見られたブロックチェーン アプリケーションの一般的な手法は、支払いチャネルです [28]。サトシ・ナカモト [66] が部分的に記述し、後にハーン (Hearn) とスピルマン (Spilman) [29] が定義および実装した後、支払いチャネルは、ビットコインのライトニング ネットワークのために、プーン (Poon) とドライジャ (Dryja) [30] によって研究されました。支払いチャネルを使用すると、送信者と受信者は任意の金額のトランザクションを相互に送信し、2 つのトランザクション (トランザクションのセットアップとクローズ) に対してのみトランザクション料金を支払えばよくなります。これは最初に、受信者に送金するか送信者に返金することしかできないトークンをロックするトランザクションを送信者に送信させることによって実現されます。通常、トークンは将来のある時点 T でのみ送信者に送り返すことができます。その間、トークンを (徐々に、または全額一度に) 受信者に送金できます。送信者は、より多くのトークンを使って受信者に向けたトランザクションに継続的に署名し、それらをブロックチェーンではなく、直接受信者に送信します。受信者は T 時点までいつでも、最後に受け取ったトランザクションを送信して、送信された合計金額を請求できます。

支払いチャネルは、送信者が、継続的な支払いに関する暗号化された証拠を受信者に提供する効率的な方法を提供します。中間の支払いにはトランザクション料金がかからないので、少額を任意に支払い、頻繁に任意に送金できます。実際には、トランザクションを検証するための計算のオーバーヘッドと、トランザクションを送信するための帯域幅要件がボトルネックになります。

支払いチャネルは、任意の量の中間支払いに対して一定の複雑なトランザクション料金を効果的に提供しますが、すべてのユースケースで十分に効率的とは言えません。特に、やり取りする相手が頻繁に変わる大量の送信者と受信者がいるシステムでは、新しい支払いチャネルを絶えず作成するのは費用がかかりすぎることがわかります。同様に、単一の HTTP リクエストや 10 秒のビデオ ストリーミングなど、非常に小規模または短期間で提供されるサービスの場合、必要なオンチェーン トランザクションのトランザクション料金は高すぎる可能性があります。

E.6. 確率的支払い

決済はブロックチェーンで発生し、トランザクションの手数料を伴うという仮定を変えられない場合、理論上の最小コストは単一トランザクションのコストです。ブロックチェーンでは状態遷移を実行するために少なくとも 1 つのトランザクションが必要です。したがって、なんらかの (マイクロ) ペイメントを決済するには、少なくとも 1 つのトランザクションが必要です。

支払いチャネルで必要なセットアップ トランザクションを回避し、それでも支払いが行われていることを受信者に証明できるとしたらどうでしょうか。

幸いにも、ブロックチェーン業界には同様の解決済みの問題があります。それはマイニング プールのシェアです [31]。ビットコインなどのネットワークでのプルーフオブワークの難易度が増すにつれて、マイナーは 1 人でブロック ソリューションを見つけるには何年もかかるような大きな変動を避けるために、計算能力をプールし始めました。マイニング プールは、ハッシュ パワーに比例して報酬を授与します。個々のマイナーは、基になる同一のブロック ハッシュのソリューションを継続的に送信することで、より低い難易度でハッシュ パワーを証明できます [32]。この手法により、マイニング プールは各プール メ

ンバーのハッシュ パワーを、そのプール メンバーが実際のプルーフオブワークのターゲットを満たすソリューションを見つけるかどうかに関係なく、暗号化で検証できます。

同じ考え方を支払いチャネルに適用することで、確率的な支払いスキームを構築できます。このスキームでは、送信者が、実際の支払いが行われるかどうかに関係なく、*平均すると*、受信者が支払いを受けていることを、受信者に対して継続的に証明します。これにより、セットアップ トランザクションが不要な確率的マイクロペイメントを実現できるため、受信者は「キャッシュ イン」するときにのみ、トランザクション料金を支払えば済みます。

イーサリアム スマート コントラクトを使用してこのような確率的マイクロペイメントを構築する方法を検討する前に、一歩下がって、ブロックチェーン技術に先行して 1996 年に David Wheeler [88] が最初に公開した確率的支払いの元のアイデアを見てみましょう。Wheeler は、確率的支払いの中心的なアイデアと、受信者と送信者（当該論文では「購入者」と「販売者」）のどちらも確率的イベントの結果を操作できないように乱数コミットメントを使用しながら、依然として確率と当選金額を互いに提供しあえる電子プロトコルを説明しています。

Wheeler のアイデアについていくつかの論文が続いて発表され、1997 年に Ronald Rivest [82] が電子マイクロペイメントに確率的支払いを適用する方法を説明した論文を発表しました。2015 年に、Pass と Shelat [81] は、ビットコインなどの分散通貨に確率的マイクロペイメントを適用する方法を説明し、以前のスキームはすべて信頼できる第三者に依存していると指摘しました。翌年、Chiesa、Green、Liu、Miao、Miers、および Mishra [58] は、ゼロ知識証明を使用してこの研究を拡張し、暗号通貨プロトコルに適用可能な分散型かつ匿名のマイクロペイメントを提示しました。

最近のイーサリアム ベースのシステムにおける支払いチャネルへの関心と普及を考えると、支払いチャネルの観点から確率的支払を見ることには価値があると言えます。確率的支払いでは、最初のセットアップ トランザクションの省略と引き換えに、正確な金額の送金を保証する機能が失われ、代わりに確率的な保証のみが達成されます。しかし、確率、当選額、支払い頻度の調整により確率的マイクロペイメントを非常にきめ細かくすることで、大きな欠点なしに、いくつかのクラスのブロックチェーンベースのアプリケーションの支払チャネルを置き換えることが可能だと示すことができます。

基本的に、初期セットアップ トランザクションを回避できるため、同じ送信者アカウントから、任意の数の受信者への任意の小規模なサービス セッションに支払いを行う一方で、支払い額の正確な確率をそれぞれの受信者に証明する機能が得られます。サービス プロバイダー（オーキッド ネットワークのリレーまたはプロキシ ノード）が十分な量のサービスを提供すると仮定すると、確率的な支払いの変動はすぐに均等になります。

E.7. オーキッド トークンの詳細

インセンティブ化

インセンティブ化は、ネットワークの部分的な所有権を人々に与えることにより、新しいプロトコルとネットワークをブートストラップする方法です [33]。オーキッドなどの新しい分散型ネットワークの多くは「鶏と卵」の問題に悩まされています。プロキシ ノードとリレー ノードが多いほど、ネットワークがユーザーに提供するユーティリティが増えます。一方で、ユーザーが多いほど、プロキシ ノードまたはリレー ノードを実行する価値が高くなります。新しいネットワーク トークンを展開することにより、潜在的なすべてのユーザーが早期にネットワークを使用するように動機付けられるため、ネットワーク効果を加速できます。

分離

他の分散型システム上に構築された分散型システムでは、新しいトークンによって新しいシステムの市場価値を基礎となるシステムから分離します。たとえば、2017 年 10 月現在、イーサの時価総額は 300 億ドルであり、毎日のグローバル取引量は 5 億ドル [34] です。イーサの価格は、暗号通貨に関する全般的な推測、イーサリアム マイナーのハッシュ パワー、イーサリアムで構築された数百のプロジェクトの成功と失敗など、さまざまな要因の影響を受けます。ただし、1 つのプロジェクトの失敗や成功はイーサの価格には大きな影響を及ぼしませんが、問題のプロジェクト固有のトークンには劇的な影響を与えます。新しいトークンを使用して市場価値を切り離すことにより、問題のプロジェクトおよびシステムの規模と健全性のより良い指標が作成され、そのシステムの将来に関する予測市場を効果的に作成できるようになります。

流動性の高い市場

システム固有のトークンの流動的な市場がある場合、そのシステムに大きく依存しているユーザーは、ショート ポジションを取ることでシステムの潜在的な障害に対してヘッジすることができます。これが突飛だと思えるのなら、金融デリバティブは当初、企業が将来の不運な出来事をヘッジできるように作られたものであったことを思い出してください。0x [35] や etherdelta [36] などの分散型取引所、および Augur [37] や Gnosis [38] などの予測市場の出現により、イーサリアム ベースのトークンやシステムのデリバティブの実現もそれほど遠い未来ではないでしょう。実際、こうしたデリバティブは、従来の金融デリバティブよりも効果的です [39]。前者には信頼できる当事者も許可も必要なく、匿名となる可能性さえあります。

新しいトークン

また、新しいトークンにより、ステークホルダー向けの特定のインセンティブが設計しやすくなります。トークンは新しいシステムからのみその価値を引き出すため、システムの成功に向けて働くすべての人にとって強力なインセンティブとして機能します。イーサリアム スマート コントラクトは、トークンの自律ロックを実装して、トークン ホルダーが定められたスケジュールに従ってのみ、トークンにアクセスできるようにします。これにより、時間をかけてインセンティブが調整され、トークン ホルダーは特定のチームや関連企業などの社会構造ではなく、*システム*の長期的な成功を重視するようになります。オーキッド ネットワークが単にイーサを使用し、ステークホルダーがイーサのロックアップを受け取った場合、彼らは実際には、イーサリアムを使用する特定のシステムではなく、イーサリアムの全体的な成功に向かって動機付けられます。そのような結果は、オーキッド ネットワークとプロジェクトにとって最適なインセンティブ調整ではないと主張することができます。

E.8. 検証可能なランダム関数

前のセクションで説明した支払いチケットは、受信者の乱数コミットメントを検証可能なランダム関数 (VRF) に置き換えることにより、インタラクティブ性を低くすることができます。VRF の IETF ドラフトは、1999 年に Micali、Rabin、および Vadhan によって最初に公開され [84]、最近 Goldberg と Papadopoulos によって提案されました [40]。このドラフトでは、RSA を使用するものと楕円曲線 (EC-VRF) を使用するものの 2 つの VRF 構造を特定しています。

オーキッド ペイメント チケットの送信者は、VRF を使用して、受信者からチケットごとの（当選チケットが見つかるまでチケットごとの）コミットメントを必要とせずにチケットを作成できます。むしろ、送信者は

受信者の公開鍵を知るだけで済みます。送信者は、前述のチケット スキームの乱数ハッシュをこの公開鍵に置き換えます。効率のために、これはチケットにすでに存在する資金を受け取るための受信者公開鍵である可能性がありますが、鍵分離の暗号化原則を順守するために、2 番目の鍵が必要になる場合があります。

ただし、オーキッドの支払いスマート コントラクトで EC-VRF を検証するには、楕円曲線演算の明示的な EVM 加速が必要になります。これらをソリッドまたは EVM アセンブリに直接実装すると、ガス コストの面で非常に高価になるためです。

幸いなことに、イーサリアム ビザンチウム [41] のリリースでは、イーサリアムのネットワークは、楕円曲線スカラーの加算と乗算 [42]、および alt bn128 曲線 [49] に対するペアリング チェック [43] の EVM サポートを追加しました。EC-VRF の構造は任意の楕円曲線に対して定義され、IETF ドラフトは EC-VRF-P256-SHA256 を EC-VRF 暗号スイートとして明確に定義しています (ここで P256 は NIST-P256 曲線 [53])。ただし、十分なセキュリティ レベルを達成しながら、代わりに alt bn128 曲線を使用できない理由はないようです。また、SHA256 は Keccak-256 に置き換えることができます。これにより、イーサリアム スマート コントラクトでの VRF 検証が可能になり、オーキッド ペイメントの支払いのスマート コントラクトとの統合が可能になります。

ただし、alt bn128 曲線は zcash で使用されますが、P256 と比べてはるかに新しい曲線であり、十分に研究されていません。おそらくより重要なのは、EC-VRF の構築がレビュー待ちの初期のドラフトであり、EVM ビザンチウムのアップグレードが本書の執筆時点で行われており、重要な価値を扱うライブ システムでの実証がまだなされていないことです。したがって、オーキッドの確率的マイクロペイメントでの EC-VRF 使用はすぐに実行可能ではなく、オーキッド プロジェクトは、EVM で検証できる EC-VRF-ALTBN128-KECCAK256 構造などの使用可能性に関するさらなる研究の実施を目指します。

E.9. 非対話型支払いスキーム

セクション E.8 では、オーキッドの支払いスキームの乱数コミットメントを VRF に置き換えることで、乱数コミットメントに関連付けられた通信手順がなくなり、スキームの非対話性が高められることを示しています。送信者がチケットを作成する前に受信者がコミットメントを送信者に伝える必要がある代わりに、送信者は公開されている受信者情報のみからチケットをすぐに作成できます。

各受信者は、VRF 専用の新しい鍵ペアを生成し、セクション 4.1 で詳述されている他の公開受信者情報とともに公開鍵を公開します。送信者はこの公開鍵をチケットに設定するだけで、受信者は受信したチケットに対応する秘密鍵で署名します。セクション D.4 で定義されたチケット検証ロジックは、当選確率しきい値と比較する値として受信者の VRF 署名を解釈します。

セクション E.8 で説明したように、これは支払いスキームに対する比較的単純な変更ですが、EVM での VRF 検証の実現可能性にはさらなる調査が必要です。

F. 関連作業

オーキッド プロジェクトは、ピアツーピア ネットワーク (P2P)、ブロックチェーン、暗号化やオーバーレイネットワークの分野での膨大な作業に基づいています。オーキッドは、それらの初期の研究によって提供された洞察を、ブロックチェーン技術、特にイーサリアム [11] および Zcash [12] のより最近の P2P 研究と組み合わせます。

次のセクションでは、オーキッド プロジェクトで以前の研究が果たす役割について説明します。

F.1. 仮想プライベート ネットワーク

仮想プライベート ネットワーク (VPN) では、暗号化を使用して、VPN サブスクライバーのトラフィックをより大規模でセキュアではないネットワークに安全に転送します。この暗号化により、閲覧習慣やユーザーの IP アドレスなど一意のオンライン識別子の追跡を防止し、アクセス制限を回避できる場合があります。

VPN ユーザーは、VPN 接続が本当に安全または匿名であると想定すべきではありません。一部の VPN サービス プロバイダーは、カスタマーのネットワーク アクティビティを追跡し、VPN サブスクライバーの承認や認識すらなく、サードパーティの商業組織にデータを販売します。VPN プロバイダーのネットワーク ノードの IP アドレスも識別可能な場合があります。これにより、政府機関または Netflix などの商業組織では VPN プロバイダーのサーバーとの間のトラフィックをブロックできます。[13]。

VPN のこれらの弱点が分散型オーバーレイ ネットワークの開発につながりました。分散型オーバーレイ ネットワークは、絶えず変化する一連の出口ノードを VPN サービスに提供します。サイトが VPN 出口ノードからのトラフィックを 1 つブロックした場合、1 つ以上の代替出口ノードが動的にサービスに加えられます。

F.2. ピアツーピア プロトコル

ピアツーピア プロトコルの歴史は、ナップスター ファイル共有ネットワークにまでさかのぼります。[42]。ナップスターは、インセンティブを利用して、他のピアからファイルをダウンロードできることと引き換えに、サブスクライバーが音楽ファイルをホストすることを奨励しました。

ナップスターのネットワーク

ナップスターは、ファイルとピアの場所のインデックスを作成する集中型ディレクトリ サービスを使用していました。ナップスターのアプローチから生じた知識の集中化は、MPAA (米国映画協会) による訴訟を受けることになりました。その結果、最終的にナップスターは廃業を余儀なくされました。

ナップスターの中央ディレクトリの脆弱性に刺激されたナップスターの後継であるグヌーテラの設計者は、ネットワーク内の各ピアにファイルとノード アドレスのインデックスを配布するようにしました [43]。

グヌーテラ ネットワークの分散型インデックスのレスポンス

グヌーテラ ネットワークの設計者は、分散型インデックス アプローチを実装することにより、ナップスターの集中型ディレクトリが持っていた欠点を改善しました。このアプローチにより、ナップスターよりも回復力とスケーラビリティが向上しました。これらの改善は、P2P ネットワーク全体にインデックスを配布する別のフレームワークの開発も促しました。注目すべき例の一つは、P2P ネットワーク内のノードとリソースの効率的な検出を可能にする分散型ハッシュ テーブル (DHT) の採用です。

Tor (トーア) ネットワーク

トーアは、1990 年代半ばに米国海軍によって開発されました。それ以来、オープンソース コミュニティによる開発とトーアの使用は横ばいのままです。現在、世界中に約 7,000 ノード、3,000 出口ノード、および約 200 万人のユーザーがいます。

集中型ノード選択であるトーアの使用と、出口ノード サービスを含むノード サービスを提供するボランティアへの依存は、トーアのスループットに悪影響を及ぼします。これは、トーアが BitTorrent や他の P2P ファイル共有システムを使用できないこと、そして出口ノードが出口トラフィックのコンテンツを検査する能力に起因します。さらにトーアには、出口ノードが他のユーザーに代わって違法または危険な情報に強制的にアクセスするのを防ぐメカニズムがありません。

これらの問題にもかかわらず、トーアの比較的小規模な開発コミュニティはトーア ネットワークをより速く、より信頼性が高く、より安全にする方法を調査し続けています [25]。その議論の重要な部分は、低遅延/高帯域幅のノードをネットワークに導入する方法です [20、21、22、23、24]。

これらの目標を達成するために、トーアのネットワークはユーザーのインセンティブを提供する方法を見つけなければなりません。ユーザーから金銭的貢献を受けてトーアを改善を施すことにはいくつかの障害があります。トーアが支払いをルーティングと密接に結び付けられないため、匿名のデジタルペイメントを効果的に管理することは困難です。一部のノードは無料でルーティングを続け、他のノードは「ゴールド スター」メンバーであることへの支払いを受け取ることにトーア コミュニティが固執していることで、複雑さがもう一段階加わります。

トーアの成長が制限されているもう 1 つの技術的でない理由は、技術的に洗練されたユーザー（「技術者」）が違法サービスやダーク Web サイトにアクセスできるようにすることを主な目的とするツールとして認識されることが多いことです。その種の隠されたサービスの一例が、さまざまな違法な商品やサービスを提供する Web サイト「*シルクロード*」です [18、19]。

対照的に、オーキッドのネットワークは隠されたサービスを有効にせず、インターネットへのオープンで安全な匿名アクセスのみに焦点を当てます。

オニオン ルーティング

ここで説明するオニオン ルーティングのテクニック（および セクション F.2 で説明するガーリック ルーティング）を暗号化と組み合わせれば、P2P ネットワーク全体でより高いレベルの匿名ルーティングが実現します。

オニオン ルーティングは、データ暗号化に対する「階層化された」アプローチであり、P2P ネットワークを通るパスを作成します。メッセージは発信元ノードによって繰り返し暗号化され、その後、メッセージが通過する各ノードによって連続的に復号化されます。中間ノードは、メッセージのルーティングに必要なルーティング指示のみを受け取ります。最終（出口）ノードのみが、ルーティング指示とメッセージの両方を受け取ります。

オニオン ルーティングの例として頻繁に引用されるのがトール ネットワークです（セクション F.2）。

ガーリック ルーティング

「The Invisible Internet Project」（*I2P*、不可視インターネット プロジェクト）はトール（F.2）に似た原理に基づいた分散型の匿名化ネットワークですが、自己完結型のダークネットになるように最初から設計されています。I2P の主要な設計機能は、ガーリック ルーティングの使用です [26]。

ガーリック ルーティングは、複数のメッセージを*バルブ*と呼ばれる 1 つのパケットにバンドルします。バルブ内の各メッセージは、オニオン ルーティングの階層化暗号スタイルで暗号化されます。メッセージがまとめられているということは、I2P へのアクセスが、隠されたサービスのトールよりもかなり高速であることを意味します。I2P は、より広いインターネットへのルーティングを部分的にしかサポートしていないため、パフォーマンスの改善を完全に比較して評価することは困難です。また、バンドリングにより

トラフィック分析の判断がより困難になります。

I2P ユーザーは、ピアツーピア暗号化トンネルを使用して互いに接続しますが、トールが使用する集中型ディレクトリは存在しません。I2P は、着信トラフィックと発信トラフィックを完全に分離します。その後、回線交換ではなくパケット交換を通じて、複数のピア間で透明性のあるメッセージの負荷分散を提供します。これらの設計機能が組み合わされて、セキュリティと匿名性の両方が向上します。

大幅な改善が必要な I2P の 1 つの側面は、ノードの分散型データベースの管理です。I2P は当初 2002 年に設計されたカデムリアを使用していました [27]。カデムリアの初期バージョンは継続的に大量の CPU とネットワーク帯域幅を消費したため、スケーリングすることができませんでした。次に、I2P は *FloodFill* と呼ばれるアルゴリズムへと移行しました。ただし、この FloodFill メカニズムには、I2P 分散データベース内の情報の破損および操作のために悪用できる設計上の欠陥もあります [28]。

F.3. ブロックチェーン プラットフォーム

ブロックチェーン プロトコルにより、グローバル ステートに関する許可のない分散型コンセンサスと暗号化トークンの使用が可能になり、ノードを実行するためのインセンティブが提供されます。

ビットコイン、イーサリアム、ジーキャッシュなどのブロックチェーン デザインは、状態遷移関数を使用してグローバル状態のエントリを追加または変更するブロックチェーン プロトコルの例と言えます。これらのプロトコルは、トランザクションの検証と、プルーフオブワーク [44] などのテクニックを使ったオーダーに関するコンセンサス形成についてもノードに報酬を与えます。

イーサリアム

イーサリアム [55] は、ビットコイン [78] とともに、アプリケーション固有の暗号化トークン [33] の新しい形を開拓しました。任意に選択された計算方法に基づいてスマート コントラクトをサポートすることにより、ブロックチェーン システムを使用して、投票、管理機能、料金支払いなどアプリケーション固有の機能を提供するカスタム台帳を作成できます。

イーサリアムは、*スマート コントラクト*を実行および展開する機能を備えた分散型ブロックチェーン プラットフォームです。イーサリアムのスマート コントラクト コードは不変であり、(非決定的動作が明示的に追加されない限り) その実行において完全な決定性を持ちます。これにより、どのノードでもスマート コントラクトの実行を検証し、その結果生じるアプリケーション状態への変更を監査できます。これにより、イーサリアムのスマート コントラクトをプログラムしたとおりに実行できます。

イーサリアム アプリケーションは、強力な共有グローバル インフラストラクチャ上で実行されます。アプリケーションは価値を迅速に移転して資産の所有権を表すことができ、ソフトウェア開発者は市場を形成して、債務または約束の登録を保存し、過去に作成されたルールに従って資金を移動したりできます。サードパーティ プロバイダーまたはカウンターパーティ リスクなしですべて実行可能です。

イーサリアムの機能は全般的に便利なものですが、特にサーバーのダウンタイム、破損、不正行為への対応を余儀なくされる新興市場では有用です。

イーサリアムとオーキッド マーケットの ERC20 トークン

イーサリアムのネットワークに展開されたほとんどのトークンは ERC20 標準に準拠しています [44]。この標準では、トークンとハードウェアまたはソフトウェアのユーザー ウォレットにトークンを簡単かつ迅速に統合できる、トークンとメタデータの転送用のコンパクトでシンプルな API を指定しています。

たとえば、Augur [37] および Gnosis [38] プラットフォームは、ERC20 トークンを使用してトークン データから予測市場を作成します。ERC20 を使用すると、任意のスマート コントラクトをオーキッドのプロトコルと簡単に接続できます。これは、インターネット エンドポイントに安全にアクセスしようとしている IoT デバイスにとって価値がある場合があります。

また、ERC20 準拠により、トークン交換およびアプリケーションは、新しいタイプの ERC20 トークンによって提供される拡張機能の恩恵を受けやすくなります。これにより、さまざまなアプリケーションが ERC20 準拠のトークンを使用して情報とステータスを交換できるようになります。