Vol.2022-DPS-190 No.7
Vol.2022-CSEC-96 No.7
2022/3/10

Ethereum RPC ハニーポットの最小要件の調査と軽量化手法の提案

内田 大暉[1]　面 和成[1,2]

概要：近年，暗号資産が世界中で注目されている．代表的な暗号資産の 1 つである Ethereum においては，資産を盗むための攻撃が盛んに行われていることがハニーポットを用いた観測結果より報告されている．ハニーポットは攻撃の実態を調査するうえで重要なシステムである一方，Ethereum ノードに対する攻撃を観測するハニーポットは運用コストが一般的なハニーポットと比べ高いことが問題視されている．本研究ではそのコストを削減するために，Ethereum におけるハニーポットに必要な最小要件を複数の実験を通して明らかにした．そして軽量化された新しいハニーポットを提案し，従来手法と比較することでその有効性が従来手法と同等であることを示した．

Minimum Requirements and Lightweight Method for Ethereum RPC Honeypot

Taiki Uchida[1]　Kazumasa Omote[1,2]

Abstract: In recent years, cryptocurrencies have become popular worldwide. In Ethereum, one of the representative cryptocurrencies, honeypots' observation results revealed active attacks to steal cryptocurrencies from its nodes. While honeypots are an effective system for investigating attacks, the operational cost of honeypots for observing attacks on Ethereum nodes is higher than that of ordinary honeypots, which is a problem. This study shows the honeypot's minimum requirements to reduce the costs through several experiments. We propose a new lightweight honeypot, and its effectiveness is equivalent to a conventional method by comparing results.

1. 序論

1.1 背景と目的

近年，暗号資産がその利便性や可能性から世界中で注目され利用者数が増えている．暗号資産とは一般的に国家等に依存せず電子上で暗号技術を利用し発行，やりとりされるものである．暗号資産の代表的なものに Bitcoin[1] や Ethereum[2] などがある．それらは，ブロックチェーンを前提とした耐改ざん性をもつ非中央集権的なプロトコルをもとに Peer to Peer（P2P）ネットワークを形成することで運営されている．このネットワークには，誰でもノードを立ち上げることで参加できる．また，それらの主要なノード実装は多くの機能を持っている．たとえば個人の資産に関連する機能として，暗号資産を管理，送金できるウォレット機能や，ブロックを作成して暗号資産を得ることができるマイニング機能などが挙げられる．

多くのノード実装は管理者がノードの各種機能を利用できるように JSON-RPC 機能を提供している．一方で，適切にセキュリティ設定しないと外部のユーザが HTTP 等を経由して JSON-RPC 機能を利用しノードを操作できてしまう．実際に Cheng ら [3] は適切なセキュリティ設定がされていない Ethereum ノードを狙って出金等の不正な操作をする攻撃を，作成したハニーポットを利用して観測した．ハニーポットとはある種の攻撃を受けやすいように設

[1] 筑波大学, 〒 305–8573 茨城県つくば市天王台 1–1–1, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki 305–8573 Japan
[2] 情報通信研究機構, 〒 184-0015 東京都小金井市貫井北町 4–2–1, National Institute of Information and Communications Technology, 4-2-1 Nukui-Kitamachi, Koganei, Tokyo 184-0015, Japan

置された，調査用のシステムのことである．

そのような攻撃の実態を明らかにすることは，多くの利用者の資産を守るうえで重要であり，常時の観測や調査が求められる．従来の Ethereum RPC ハニーポットは Ethereum のフルノードを動作させることを前提としている．しかし，フルノードはブロックチェーンの検証および保持等を行うためその時間的および経済的コストが無視できない．また，ブロックチェーンの肥大化によってそのコストは日々増加している．そのため，長期間の観測はそのようなコストの観点から難しい状況である．そういった状況から Chin ら [4] はハニーポットの軽量化の方向性について提案したが，その有効性は十分示されていない．

本研究では，Ethereum ノードに対する攻撃の監視コスト削減のために，ハニーポットを運用するうえで満たすべき最小要件を複数の実験を通して明らかにした．そしてその結果から軽量化した Ethereum RPC ハニーポットを提案および性能評価し，提案手法がコスト削減した一方で従来手法と同等の有効性をもつことを示した．

1.2 貢献

本研究では事前調査および性能評価を通して，観測の観点で Ethereum RPC ハニーポットが満たす最低要件を次のように明らかにした．

- Ethereum メインネットに接続する必要がある
- RPC レスポンスで，ノードがメインネットに参加しブロックチェーンの同期が完了しているように振る舞えば，ブロックチェーン情報を実際に検証および保持する必要はない

そして新たに提案したハニーポットは従来手法と同等の観測結果が得られることを示した．さらに従来手法と比べ経済的コストを約 8 割削減できた．これらの結果は将来的な関連研究の活発化を促すことが期待できる．

2. 準備

2.1 Ethereum

Ethereum は Ether と呼ばれる暗号資産を軸とした非中央集権型の分散システムである．Ethereum はオープンソースでそれを支える P2P ネットワークには誰でも参加できる．利用者は Ether を保持でき，それを他のアカウントへ移動させたりスマートコントラクトで利用したりできる．スマートコントラクトとは Ethereum 上で動作するアプリケーションのことで，トランザクションを発行することで作成，動作できる．

2.2 ノード

Ethereum ネットワークを維持する最小単位にノードがある．ノードは他のノードとの接続のうえ P2P ネットワークを形成し，ブロックやトランザクション等の情報の伝播や検証をする．ブロックチェーンのすべての情報を保持，検証するノードをフルノードという．また，マイニングを行うノードもある．

ノードは Ethereum プロトコルを実装していればよいため，有志によって複数の実装が存在している．よく利用されているものに Geth[*1]，OpenEthereum[*2]がある．

2.2.1 プロトコル

Ethereum を支えるプロトコル群は devp2p[*3]と呼ばれる．本論文で特筆すべきプロトコルに，Node Discovery Protocol と Ethereum Wire Protocol（ETH）がある．

Node Discovery Protocol ではノードの探索および，Ethereum Node Records（ENR）を管理しルーティングテーブルに基づいて他のノードにノード情報を提供する．ENR は Ethereum ノードの情報を表すデータ構造で，ノードの ID や IP アドレス，ポート番号を保持している．またこのプロトコルでは Ethereum ネットワークの情報は保持しない．そのため Ethereum のメインネット以外にもテストネットや他のコインのシステムも本プロトコルを利用している．なお応答がないノードはルーティングテーブルから定期的に削除される．

Ethereum Wire Protocol は Ethereum ブロックチェーンの情報を交換するプロトコルである．Node Discovery Protocol を用いて発見したノードに対して，ハンドシェイクを試行し同じネットワークであれば本プロトコルで接続する．接続後はブロックやトランザクション，その実行結果であるレシート等を伝播し，ネットワーク全体でブロックチェーンを検証，維持する．このプロトコルを利用することで他のノードからブロックチェーンの情報を得ること，つまり同期ができる．

2.3 同期

フルノードを初回起動したとき，そのノードは最新のブロックチェーンのデータを所持していない．フルノードとしての役割を果たすためにはそのデータを得る必要がある．そのため，フルノードはまず初回同期を行う．ブロックやトランザクション等の情報は接続した他のノードに要求できる．最初のブロックから順にブロックを取得および検証していくことで，最終的には Ethereum の最新の状態を再現できる．

2.4 JSON-RPC

Ethereum の主要なノード実装はノードに対する操作の手段として JSON-RPC 機能を提供している．Ethereum ノードの主要実装では通信に HTTP 等が利用できる．ノードに対しての操作は公開情報であるブロックチェーンの情

[*1] https://github.com/ethereum/go-ethereum
[*2] https://github.com/openethereum/openethereum
[*3] https://github.com/ethereum/devp2p

報の取得から，ノードのウォレットより送金する操作に渡るまで広く行うことができる．多くの場合ノードの設定によってソケットをバインドする IP アドレスやポート番号，有効にする RPC メソッドを変更できる．

3. 関連研究

本章ではまず本研究に関連する研究を紹介する．その中で Chin ら [4] は Ethereum RPC ハニーポットの運用コストに着目した．一方で，それをどう解消するかの検討は不十分であり，本研究はそのコストについて着目することとした．そこで，現在における Ethereum RPC ハニーポット運用に係るコストの問題性と，Chin ら [4] が述べたハニーポットの最小条件の妥当性について取り上げる．

3.1 Ethereum ノードに対する RPC 経由での攻撃観測に関する研究

Cheng らの研究 [3] は，Ethereum ノードの RPC ポートに対するリクエストを観測した．その結果を元に脆弱なノードから Ether や Ethereum 上のトークンを盗む攻撃者の分類とその行動を分析した．約 6 ヵ月間の観測では約 1000 の IP アドレスから約 3 億リクエストを観測した．またそれらの攻撃者が 2019 年 3 月において約 58 万ドル（2022 年 1 月現在で約 1400 万ドル）に相当する Ether を奪取していることを推定した．リクエストの大半は少数が行っていること，ガス量がゼロのトランザクションを利用したフィッシングトークンの奪取やエアドロップを大量に入手する手法も発見した．

Hara らの研究 [5] は，Cheng らの研究 [3] を発展させ，テストネットに参加するノードを 9 ヵ国に設置し RPC リクエストの地域的な傾向を分析した．その結果，攻撃者は攻撃先を特定の地域に限定しないとした．また，攻撃者の詳細な行動分析を行い，その中でダークネットにパケットを送る者が存在していることを明らかにした．

Chin らの研究 [4] は，上記 2 つの研究に関連し，長期間効率的に RPC リクエストを観測できるよう攻撃者が攻撃の前に行う事前調査の手法を明らかにした．その結果からハニーポットは Node Discovery Protocol が形成するネットワークに参加する必要はあるが Ethereum のメインネットに参加する必要はないとした．ただし，ハニーポットの RPC サーバがメインネットに参加しているように振る舞う必要があるとした．また，Ethereum を奪取する攻撃をより誘発するためには実際に暗号資産を保有する必要性があることや，全体的に先行研究と比べ攻撃が活発化していることを示した．

Wang らの研究 [6] は，Chin らの研究 [4] を拡張して Etherpot と呼ばれる Ethereum RPC ハニーポットを作成し 31 日間リクエストを観測した．この研究では先行研究においては述べられていないスマートコントラクトに対する攻撃や Ethereum ネットワークを分断する恐れがあるエクリプス攻撃の存在を明らかにした．

3.2 Ethereum ネットワークの調査に関する研究

Kim らの研究 [7] は，Ethereum ネットワークを調査するために Geth を元に作成した NodeFinder と呼ばれる仕組みを提案した．これを Ethereum のメインネットで用い，約 3 ヵ月間で約 300 万のユニークノードを発見し，約 35 万ノードとハンドシェイクを行った．結果から Node Discovery Protocol が形成するネットワーク上に存在しているノードのうち，Ethereum のメインネットに参加しているノードは約半数しかいないこと等を発見した．

Wang らの研究 [8] は，Ethereum ネットワークを調査するために Ethna という仕組みを提案した．これは従来手法より正確に各 Ethereum ノードのピア数を算出でき，さらにネットワークにおけるメッセージ伝播のレイテンシを計測する機能を持っている．Ethna は NetworkObserverNode と LocalFullNode の 2 種のノードによって構築されている．前者は Geth の Fast 同期モードにおける同期中の動作と同等の振る舞いをする．Geth の Fast 同期中はノードがブロードキャストで受け取ったブロックやトランザクションは伝播せず観測に徹する．後者の LocalFullNode は実際に同期を完了したノードである．実際に Ethna を利用してメインネットで計測したところ次のような結果が得られた．Ethereum ネットワークはスモールワールドネットワークの特性を持っていること，ネットワーク上の各ノードのピア数はべき乗法則に従い，スケールフリーネットワークであること等である．

3.3 Ethereum RPC ハニーポットにおける課題

従来手法のハニーポットは基本的に，メインネットに接続する Ethereum フルノードを利用している．その運用には多くの計算資源を要するため，結果的にハニーポット運用の時間的・経済的コストが高くなる．またフルノードが管理しなければならないブロックチェーンの総データ量は増加する一方である．

一方で，果たしてハニーポットに Ethereum フルノードは必要なのであろうか．Chin ら [4] もこの点を指摘しており，ハニーポットはメインネットに参加する必要はないが Node Discovery Protocol が形成するネットワークには参加する必要があるとした．この説は 3 点から導出していた．1 つ目は Ethereum ノードを走らせなかったハニーポットでは RPC に対する十分なリクエストを観測できなかったこと．2 つ目はメインネットに参加するハニーポットの Ethereum ノードと 5 分以上直接ピアとなっていた攻撃者が観測できなかったこと．3 つ目は攻撃者が攻撃対象のノードでメインネットを利用しているかどうかを RPC リクエストで判別していることである．メインネットには

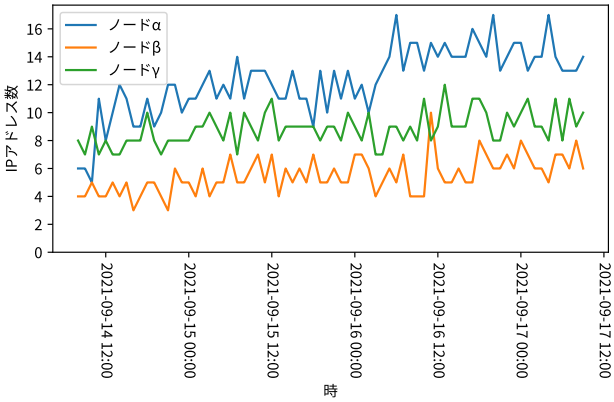


図 1 RPC リクエストした IP アドレス数 (事前調査 1)

参加せず Node Discovery Protocol の形成するネットワークに参加する手法として彼らは 2 点挙げた．1 つはハニーポットの Ethereum ノードをテストネットに参加させること．もう一方は Node Discovery Protocol のみ利用してハニーポットの情報をノードとしてネットワーク上のルーティングテーブルに登録させる方法である．ただし，これは攻撃者が攻撃対象のノードを調べる端末の IP アドレスと，ノードに RPC リクエストを送る IP アドレスが同一であるということを暗に仮定している．そのことから我々はこの説に関してより詳しく検討することにした．

4. 事前調査

4.1 異なるネットワークでの観測調査（事前調査 1）

4.1.1 概要

事前調査 1 では Chin ら [4] の考察を検証するために，ハニーポットはメインネットではなくテストネットに参加しても観測上問題がないかを調査する．そのために，メインネットとテストネットに参加するハニーポットを同時期に設置し，観測結果を比較することにした．同じような観測結果が得られればこの説は正しいといえる．

3 つのノード α，β，γ を約 3 日間動作させ，これらに対し RPC リクエストを送った IP アドレスの総数を調査した．ノード α は P2P 通信や RPC サーバはメインネットに接続する．ノード β は P2P 通信や RPC サーバはテストネットに接続する．ノード γ では P2P 通信ではテストネットに接続する一方で，RPC サーバはメインネットに接続しているように振る舞うようにした．

4.1.2 結果

結果を時間ごとに集計しグラフ化したものを**図 1** に示す．テストネットに参加したノード β はメインネットに参加したノード α と比べ，RPC リクエストを送った IP アドレス数が減少した．また，テストネットに参加するが RPC でメインネットに参加しているように振る舞ったノード γ はノード β よりも観測した IP アドレスが多かった．一方，ノード α よりは少なかった．

4.1.3 考察

ノード β が観測した IP アドレス数がノード α より減少した理由として，ハニーポットをメインネットでの通信経由で発見した (1) か，RPC での通信経由で発見した (2) 者がいると考えられる．(1) は Kim ら [7] が行ったようにメインネットに参加して自分で探すか，Ethernodes[*4]のようなメインネットのノード情報を提供する外部サービスを利用する場合が考えられる．(2) は何らかの方法でハニーポットを発見したうえで RPC リクエストを送ることでそれがメインネットに参加していると判明した場合に，複数の IP アドレス経由でリクエストを行う場合である．Chin ら [4] が指摘しているとおり，一部の攻撃者は RPC リクエストでノードが参加しているネットワークを判別している．結果としては単にテストネットに参加するだけではハニーポットとしては不十分であると分かった．

また，テストネットに参加するが RPC でメインネットに参加しているように振る舞ったノード γ は単にテストネットに参加したノード β よりも観測した IP アドレスが多かった．両者は Ethereum ネットワークでの振る舞いは同一であるため，RPC でメインネットと判明した場合に複数の IP アドレス経由でリクエストを行う者が存在するといえる．もしくは同一人物ではなく外部にその情報を提供しているサービスが存在していることも考えられる．

一方，ノード γ とメインネットに参加したノード α を比べると依然ノード γ の方が少なかった．RPC での振る舞いは両者同一であるため，ハニーポットをメインネットでの通信経由で発見する者が存在しているといえる．結果としてテストネットに参加して RPC ではメインネットに参加しているように振る舞ってもハニーポットとしては不十分であるとわかった．結論として我々は新たに，ハニーポットはメインネットに参加する必要があるとした．

4.2 同期完了前後での観測調査（事前調査 2）

4.2.1 概要

4.1 節では Ethereum RPC ハニーポットは Ethereum のメインネットに参加する必要があるとした．本節では Ethereum RPC ハニーポットがメインネットに参加するうえでブロックチェーンの同期を行う必要があるかどうかについて検証する．同期を行わない場合，必要とするストレージやセットアップ時間を削減できる．

本調査ではメインネットに参加し，初回同期を行う OpenEthereum のノードを立ち上げ RPC ポートへのリクエストを監視した．同期完了前後で RPC リクエスト数等に違いがなければ同期を完了させる必要はない．

4.2.2 結果

初回同期は 2021-09-23 の 22 時台に完了した．**図 2** に

[*4] https://www.ethernodes.org/

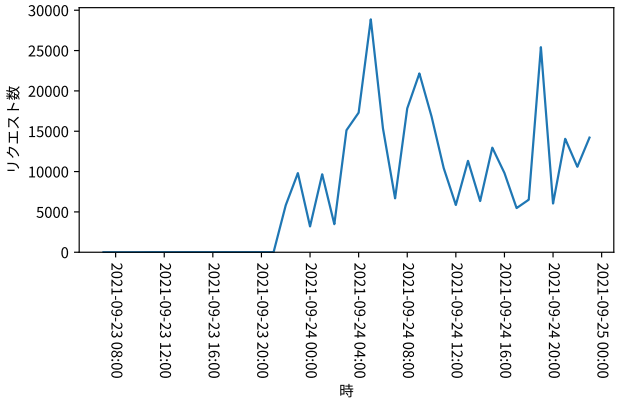


図 2 ある 1 つの IP アドレスによる RPC 利用回数 (事前調査 2)

ある 1 つの IP アドレスによる時間ごとの RPC リクエスト数を示す．ちょうど同期の完了時点からこのクライアントによるリクエスト数が増大している．

4.2.3 考察

結果で表したあるクライアントはハニーポットが同期を完了したタイミングからリクエストを多く送信するようになった．これは何らかの方法によってハニーポットの同期が完了していることを検知しているといえる．対象のノードの同期状況について確認するには ETH プロトコルで通信する，RPC リクエストで確認する，外部サービスを利用するという 3 点が考えられる．外部サービスは同期情報の取得に前者 2 つのどちらかを利用していると考えられる．

我々は RPC で同期が完了しているように振る舞えば実際に同期しなくてもよいのではないかと考えた．ETH プロトコルで直接通信している場合には対処できていないが，常にすべてのノードの最新情報を ETH プロトコル経由で取得するのは現実的に難しい．そのため，本研究では Ethereum P2P ネットワーク上で同期情報を変更する必要はないと考えた．

5. 提案手法

5.1 概要

本章では従来手法のコスト的問題を緩和した新たな手法を提案する．その問題とは，Ethereum RPC ハニーポットにおいて必要な Ethereum ノードが，多くの計算資源を必要とするというものである．提案手法におけるハニーポットでは，メインネットで動作することによって従来手法と同等の観測ができ，ブロックチェーンの同期は行わないことによってコスト的問題が緩和されることを期待している．

5.2 各サービスの説明

本ハニーポットは Ethereum ノードの RPC が利用される毎に調査で必要なデータを保存するシステムである．本ハニーポット内では複数の役割をもつソフトウェアが動作しており，それらのことをここではサービスと呼ぶ．さらに，ハニーポットの全体図を**図 3** に示した．以下より主なサービスについて説明する．

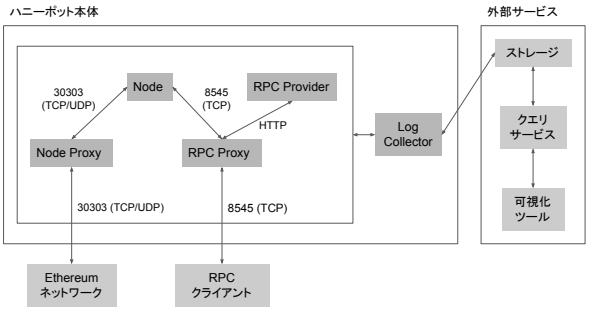


図 3 提案ハニーポットの全体図

5.2.1 Node サービス

Node サービスは Ethereum メインネットにノードとして参加する．これまで議論したように，ハニーポットが広く発見されるためにはメインネットに参加する必要がある．Ethereum ノードの実装には OpenEthereum を採用する．この実装は RPC 機能を提供しているが，外部からのリクエストは後述する RPC Proxy サービスを経由する．

軽量化のためブロックチェーンの同期をしないように Ethereum ノードに変更を加えた．同期を無効にするというのは，ブロックのダウンロード，保存，検証処理をスキップするということである．接続しているピアからブロードキャストで受け取ったブロックやトランザクションは検証できないため伝播しない．この挙動は OpenEthereum における同期中の挙動や Wang らの研究 [8] で利用された NetworkObserverNode と同等の動きである．また，このようにすることで誤ったデータをネットワークに広めることを防ぐことができる．

5.2.2 RPC Proxy サービス

RPC Proxy サービスは 8545 (TCP) ポートで Ethereum ノードに対する RPC リクエストを待ち受け，取得先を決定する．その際にリクエストのペイロードと接続先 IP アドレスをログデータとして出力する．

前章では，ハニーポットのノードが同期を行わない場合は同期が完了しているように RPC レスポンスを返す必要があるとした．そこで次に示す 2 つのメソッドが呼び出された場合は特別に扱う．その他のリクエストは変更の必要がないと考え，Node サービスに引き渡す．

特別に扱うメソッドの 1 つ目は `eth_blockNumber` である．このメソッドは呼び出し時点での最新ブロック番号を取得するメソッドである．このメソッドを利用すると同期がどこまで完了しているかが分かるため，攻撃者はこのメソッドを利用してノードの同期状態を確認しているのではないかと考えられる．そこでこのメソッドが呼び出された場合は外部サービスから最新のブロック番号を取得する．

表 1 性能評価に利用したインスタンス一覧

名称	CPU コア	RAM（GB）	SSD（GB）
従来手法	6	16	320
提案手法$_{Eth}$	1	2	55
提案手法$_{RPCProvider}$	1	2	55

2 つ目は `eth_getBlockByNumber` メソッドである．Hara ら [5] は，一部の攻撃者はこのメソッドを利用し，そのノードのチェーンの種類を確認しているとした．任意のブロックが判別に利用される可能性があるため，ハニーポットは外部サービスを利用するなどしてすべてのブロック情報を返却できることが理想的である．一方，事前調査において外部から要求されたブロックは少数であったため，それらのみ値を定数で持ち対応した．そのブロック番号は `0x0`, `earliest`, `0x1`, `0x1d4c00`, `0x1d7310` である．なお，`earliest` は `0x0` を指す．今回は実装単純化の制限として，上記のブロック以外が要求された場合は空の値を返す．

6. 性能評価

6.1 概要

本章では前章で述べた提案手法を従来手法と比較し，性能を評価する．提案手法では従来手法より必要とするコストが減る一方で，従来手法と同等の観測結果が得られることを示したい．そこで一定期間，提案手法と従来手法のハニーポットを動作させる．なお，ここではハニーポットの性能を以下より評価する．

- 単位時間あたりに RPC リクエストを行った IP アドレス数
- リクエストされたメソッドの割合

ただし，同時刻にホスト以外同条件でハニーポットを複数設置したとしても Ethereum ノードが接続するピアは非決定的であることなどから同じ結果を得ることは困難である．その点も考慮しつつ分析および考察する．

6.2 準備

今回は従来手法 2 ノード，提案手法 2 ノードで検証する．ハニーポットはホスティングサービスの Vultr に設置した．従来手法は同期を行うために比較的高性能のインスタンスが必要であり，一方提案手法は比較的性能が低いもので十分と考えられる．そこで**表 1** に示すように利用するインスタンス種を分けた．

提案手法では最新のブロック番号を外部サービスから取得する必要があり，その取得は図 3 における RPC Provider サービスが行う．今回，外部サービスからの値の取得回数を抑える必要があったため，別途専用のインスタンスを 1 台用意しそこで一元的に取得および提供した．

Hara ら [5] によってハニーポットの設置場所は任意の場所で問題ないと報告されているため，本ハニーポットはすべて東京に設置した．また，Chin ら [4] は WebSocket 経由のハニーポットでのアクセスも確認したが，HTTP 経由の方が一般的であり性能評価としては十分と判断したため今回は HTTP のみで観測する．

表 2 Ethereum ノードによるストレージ使用量

ノード	使用量（GB）
従来手法$_{Eth1}$	145
従来手法$_{Eth2}$	145
提案手法$_{Eth1}$	<1
提案手法$_{Eth2}$	<1

表 3 累計インターネット転送量と使用料金

インスタンス	転送量（GB）	料金（米ドル）
従来手法$_{Eth1}$	364	37.30
従来手法$_{Eth2}$	392	37.30
提案手法$_{Eth1}$	61	4.69
提案手法$_{Eth2}$	47	4.69
提案手法$_{RPCProvider}$	<1	4.69

従来手法はブロックチェーンの同期を行い，すべての RPC リクエストを Ethereum ノードが受ける．一方，提案手法は前章で説明したとおり同期はせず，メインネットで同期が完了しているように振る舞うために一部 RPC メソッドのレスポンス取得方法に手を加えている．

なお，従来手法には`--no-ancient-blocks` オプションを利用した．先行研究ではこのオプションが利用されているかは明らかではない．このオプションを利用すると初回同期に利用したスナップショット以前のブロックを取得しないため，一部振る舞いに違いがある可能性もあるが，コストの観点から利用することとした．

6.3 結果

実験は 2021-11-30 から 2021-12-14 にかけて行った．初日と最終日は一部の時間帯のみでの観測のためリクエスト数等が少なくなっていることに注意されたい．

6.3.1 コスト

表 2 に観測終了時点での各 Ethereum ノードによるストレージ使用量を示した．提案手法による使用量は 1GB 未満であり，従来手法と比べると 99%以上の削減となった．

表 3 にインスタンス別の累計インターネット転送量とそれを動作させるのにかかった料金を示した．転送量の算出には Vultr が報告した値を利用した．インバウンドとアウトバンド転送量のうち多いほうが採択されている．それぞれ平均値を元にすると提案手法は従来手法と比べ約 85%転送量が削減されている．また，料金には CPU，メモリ，ストレージ，インターネット転送量すべて含めたコストが含まれている．結果として従来手法は提案手法と比べて約 81%料金が削減された．[*5]

なお CPU やメモリに関しては表 1 に示したように，提

[*5] https://www.vultr.com/ja/resources/faq/

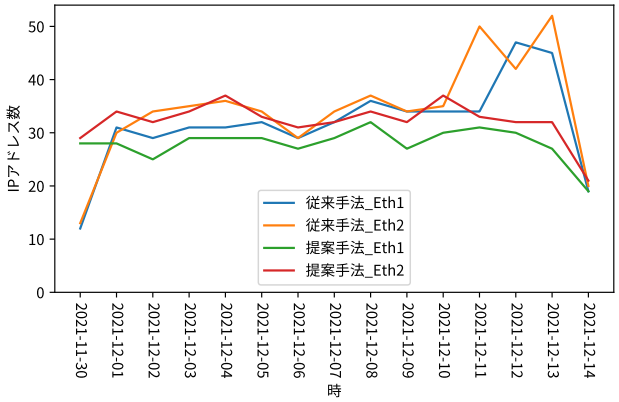
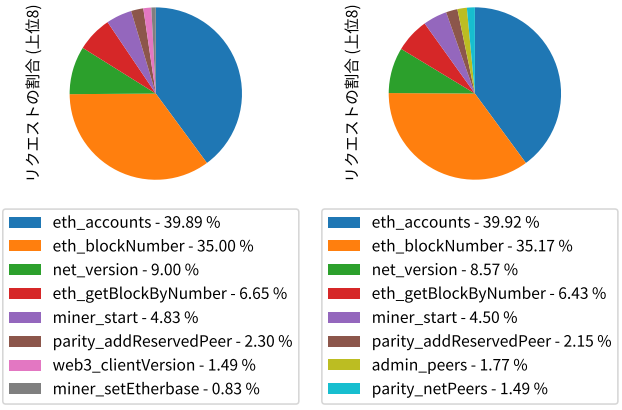


図 4 RPC リクエストした IP アドレス数



(a) 従来手法$_{Eth1}$ (b) 従来手法$_{Eth2}$

図 5 従来手法における上位 8 メソッドの RPC リクエスト数の割合

案手法は従来手法と比べ低性能なものを利用したが観測上問題なく動作した．

6.3.2 リクエストユーザ数

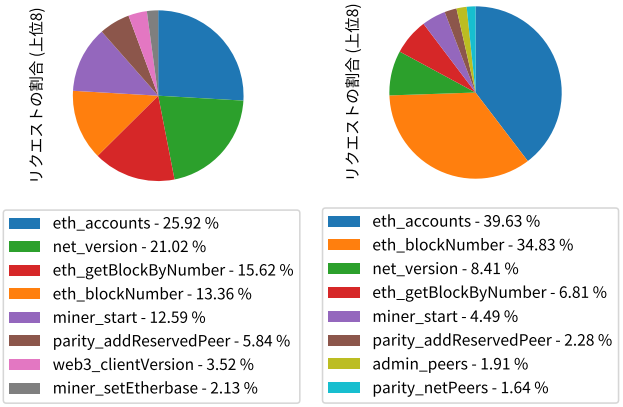
図 4 にはハニーポットの RPC を利用したユニークな IP アドレス数をノード別に日毎で表した．12/11 より前では 提案手法$_{Eth2}$ は 2 つの従来手法のノードと同程度の IP アドレスを観測できたことが分かる．一方，12/11 以降は従来手法のみ観測数が増えている．また， 提案手法$_{Eth1}$ は 提案手法$_{Eth2}$ と比べ全体的に数が少なくなっている．

6.3.3 リクエストメソッドの割合

次に，リクエストされたメソッド別のリクエスト数に関して**図 5** に従来手法 2 ノードにおけるグラフを示し，**図 6** には提案手法 2 ノードにおけるグラフを示した．なお，リクエスト数上位 8 位のメソッドのみ表示した．

6.3.4 外部サービスによるデータ

Chin ら [4] は一部の攻撃者は攻撃対象を効率的に見つけるために，Ethereum ノードの情報を収集している外部サービスを利用し Ethereum ノードの IP アドレスを取得していると考察している．そこで有名なノード情報収集サービスの 1 つである Ethernodes において観測期間中，設置し



(a) 提案手法$_{Eth1}$ (b) 提案手法$_{Eth2}$

図 6 提案手法における上位 8 メソッドの RPC リクエスト数の割合

表 4 Ethernodes による各ノードの最終発見日時および最終掲載日

ノード	最終発見時間	最終掲載日
従来手法$_{Eth1}$	2021-12-13 23:05	2021-12-14
従来手法$_{Eth2}$	2021-12-14 01:30	2021-12-14
提案手法$_{Eth1}$	N/A	N/A
提案手法$_{Eth2}$	2021-11-30 22:10	2021-12-07

た各ノードが最後に発見された日時とその最終掲載日を表したものを**表 4** に示した．提案手法$_{Eth1}$ は Ethernodes によって 1 度も発見されなかったため，データは存在しない．

6.4 考察

6.4.1 コストから見た有効性

まず提案手法のハニーポットが従来手法のものと比べ，運用コストの観点で軽量化できているかを考察する．結果として 6.3.1 項に示したように，ストレージ使用量，インターネット転送量，CPU，メモリの観点から提案手法は軽量化できたといえる．これらが軽量化された大きな理由として提案手法はブロックチェーンの同期を行わないためであるといえる．

6.4.2 リクエストユーザ数から見た有効性

ハニーポットに RPC リクエストを行う IP アドレス数はハニーポットの性能を測るうえで重要な指標である．図 4 からは 提案手法$_{Eth1}$ が 提案手法$_{Eth2}$ と比べ全体的に少なくなっていること，12/11 以降で従来手法のみ観測数が増えていることが分かるがそれについて原因を考察する．

まず表 4 を見ると， 提案手法$_{Eth1}$ は観測期間全体を通して一度も Ethernodes によって発見されず Web 上に掲載されなかった．このことが 提案手法$_{Eth1}$ に対してリクエストした IP アドレス数が 提案手法$_{Eth2}$ と比べ全体的に少なくなった原因ではないかと考察される．

また，図 4 からわかるように 提案手法$_{Eth2}$ は 2021-12-10 までは他の従来手法と同程度の IP アドレス数を観測していた．しかし，それ以降は従来手法では観測した IP アド

レスが増えた．我々はこの原因として，表 4 にあるように提案手法$_{Eth2}$ は 12/07 を最後にしてそれ以降は Ethernodes に情報が掲載されなくなったことに起因するのではないかと考えた．一方，従来手法の 2 ノードは観測最終日までノードの情報が掲載されている．これらの IP アドレスが Ethernodes を見ていると考察される根拠は次のとおりである．12/11 以降に従来手法だけが観測した IP アドレスの約 9 割は 12/08 以降に初回リクエストを行っている．また，それらは 30303 ポートでハニーポットと通信していない．つまり外部サービスを利用してハニーポットを発見した可能性が高い．掲載されていた期間のみで提案手法と従来手法とを比較すると，有意差はないと考えられる．

我々は結果として，Ethernodes に掲載されれば提案手法は従来手法と同程度の IP アドレスの観測が可能となると考えた．Ethernodes には提案手法のうち 1 つは掲載されたため，提案手法であっても掲載されることもわかった．このような外部サービスの多くは Kim ら [7] が行ったように Ethereum ネットワークにおいて実際に通信してノード情報を収集していると考えられる．ただし Ethereum ネットワークの性質上，常にネットワークに参加している全ノードの情報を取得することは困難であるため，一部取り逃しが発生しているのではないかと考えられる．これを緩和するためにはハニーポット設置の際，より多くの Ethereum ノードを設置するとよいだろう．

6.4.3 リクエストメソッドの割合から見た有効性

図 5 と図 6 によると，提案手法$_{Eth1}$ はこれまで指摘したように一部ユーザに発見されていないためにメソッドの呼び出し数が他と比べ異なっている．一方，他の 3 ノードは同様なグラフとなっており，提案手法でも従来手法と同様なメソッドが観測できた．

6.4.4 ネットワークへの影響考慮

本研究の実験では Ethereum へ悪影響を及ぼさないよう十分配慮した．提案ハニーポットはブロードキャストで受け取ったデータに関して通常ノードの同期中と同じ振る舞いを行っている．Ethereum ネットワークでは同期中や故障中など伝播しないノードがいることは想定されており，かつ各ノードは新規のブロックやトランザクション等のデータを一般的に数十のピアに対して伝播している．さらに意図しない悪影響を最大限防ぐために，実験では提案ハニーポットの数を可能な限り十分少なくした．

ただしより大規模な実験を実施する場合，ネットワークへの影響に関してさらなる検討が必要である．また，Ethereum ネットワークへの貢献をすべきだというのはもっともな指摘である．ところで，本論文の貢献は次のように言い換えられる．各ハニーポット自体は Ethereum メインネットに参加する必要があるが，ブロックチェーンの検証および保存等の処理は他のノードへ委譲できることを明らかにした，というものだ．つまり，それらの処理を行うフルノードをハニーポットとは別に用意し，処理を委譲することで先述した問題も緩和できると考えられる．

7. 結論

本研究では問題となっている Ethereum RPC ハニーポットの監視コストを軽減するために，複数の実験を通してハニーポットに必要な最低要件を次のように明らかにした．

- メインネットに参加する必要がある
- ブロックチェーンの同期を実際に行う必要はない
- RPC サーバはメインネットに参加し，最新のブロックまで同期しているように振る舞う必要がある

そのうえでブロックチェーンの同期を行わないハニーポットを提案した．従来手法と性能比較したところ，提案手法は約 8 割の経済的コスト削減を実現したうえで従来手法と同等の観測結果が得られた．結果として，提案手法の有効性を示すことができた．

今後の課題として，本研究では先行研究で発見されている RPC を経由した攻撃に関しての発展的な調査は行っていない．ただし本研究で提案した手法をもとにすることで，より大規模に長期間観測することが可能になったといえる．今後はそのような観測をしていくことが重要である．

謝辞 本研究成果の一部は，JSPS 科研費 19H04107 の助成を受けたものである．

参考文献

[1] Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System (2009).
[2] Buterin, V.: Ethereum White Paper: A Next Generation Smart Contract & Decentralized Application Platform (2013).
[3] Cheng, Z., Hou, X., Li, R., Zhou, Y., Luo, X., Li, J. and Ren, K.: Towards a First Step to Understand the Cryptocurrency Stealing Attack on Ethereum, *The 22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2019)*, pp. 47–60 (2019).
[4] Chin, K. and Omote, K.: Analysis of Attack Activities for Honeypots Installation in Ethereum Network, *2021 IEEE International Conference on Blockchain (Blockchain)*, pp. 440–447 (2021).
[5] Hara, K., Sato, T., Imamura, M. and Omote, K.: Profiling of Malicious Users Targeting Ethereum's RPC Port Using Simple Honeypots, *2020 IEEE International Conference on Blockchain (ICBC 2020)*, pp. 1–8 (2020).
[6] Wang, J., Sasaki, T., Omote, K., Yoshioka, K. and Matsumoto, T.: Etherpot: A honeypot for observing cyberattacks on Ethereum client, 電子情報通信学会技術研究報告; 信学技報, Vol. 121, No. 69, pp. 56–61 (2021).
[7] Kim, S. K., Ma, Z., Murali, S., Mason, J., Miller, A. and Bailey, M.: Measuring Ethereum Network Peers, *Proceedings of the Internet Measurement Conference 2018 (IMC 2018)*, pp. 91–104 (2018).
[8] Wang, T., Zhao, C., Yang, Q., Zhang, S. and Liew, S. C.: Ethna: Analyzing the Underlying Peer-to-Peer Network of Ethereum Blockchain, *IEEE Transactions on Network Science and Engineering*, Vol. 8, No. 3, pp. 2131–2146 (2021).