Computer Security Symposium 2017
23 - 25 October 2017

γ-Deniabilityに基づくプライバシを考慮したモバイル端末向けBitcoinウォレットの承認済み取引確認法

金村 晃太[1]　豊田 健太郎[1]　大槻　知明[1]

概要：Bitcoin の全取引履歴はブロックチェーンに記録される．ストレージに制約のある携帯端末向けに軽量クライアントが開発されている．軽量クライアントは，自身の Bitcoin アドレスを入力したブルームフィルタをクライアントに送信し，陽性となる取引のみを転送してもらう．しかし，軽量型クライアントは高い確率で自身に関連する Bitcoin アドレスを特定されるため，プライバシ指標に基づくブルームフィルタの設計が必要である．本稿では，γ-Deniability に基づくプライバシを考慮した軽量型クライアント向け承認済み取引確認法を提案する．特性評価により，提案法によってプライバシ保護が改善されることを示す．

キーワード：Bitcoin，プライバシ

Design of Privacy-Preserving Bloom Filter Based on γ-Deniability for Mobile Bitcoin Client

Kota KANEMURA[1]　Kentaroh Toyoda[1]　Tomoaki OHTSUKI[1]

Abstract: In Bitcoin, all transactions issued by users have been recorded in the common ledger, called blockchain, which is shared by all users. Lightweight clients are developped because of a constrained strage of portable devices like smartphone. For a lightweight client to check if there are transactions related to it, a Bloom filter where their Bitcoin addresses are involved is sent to a full client that possesses the entire blockchain. However, it is necessary to preserve the privacy of lightweight clients when designing a Bloom filter because lightweight clients' Bitcoin addresses will be identified by a full client with high probability. In this paper, we propose a privacy-preserving Bloom filter scheme for lightweight clients based on γ-Deniability. From a simulation, we show that the privacy level of lightweight clients can be improved by applying our proposed scheme.

Keywords: Bitcoin，Privacy

1. はじめに

暗号通貨の革命は世界で非常に注目を集めている [1]．特に，Bitcoin は最も成功した例であり，法定通貨と交換できる [2], [3]．Bitcoin は中央サーバや管理者が存在せず，P2P (Peer-to-Peer) ネットワーク上の参加ユーザによって管理される仮想通貨システムである [4]．ユーザは Bitcoin を送受するためにトランザクションと呼ばれるメッセージを P2P ネットワーク上にブロードキャストし，参加ユーザがそのトランザクションを承認する．承認されたトランザクションはブロックチェーンと呼ばれる台帳に記録される．

近年，Bitcoin による支払を受け付けているサービスは増えている [5]．例えば，Bitcoin による支払を受け付けている自動販売機などが挙げられる [6]．携帯端末による支払は非常に普及しているため，近い将来，より多くの携帯端末向け Bitcoin クライアントが利用可能になると考えられる．しかし，2017 年 8 月 25 日現在，Bitcoin のブロックチェーンのサイズは 100GB を超えており，スマートフォンなどのストレージに制限のあるデバイスは，ブロックチェーンの保持に適していない．この問題を解決するため

[1] 慶應義塾大学大学院　理工学研究科
Science and Technology, Keio University

に，SPV (Simplified Payment Verification) クライアントと呼ばれる軽量型クライアントが開発された．軽量型クライアントはブロックチェーンの全データをダウンロードせずに使用できる Bitcoin クライアントである．軽量型クライアントは自身の管理する Bitcoin アドレスを含むトランザクションを確認するために，完全型クライアントに Bitcoin アドレスを伝える必要がある．しかし，これは次の 2 つの問題がある．1 つ目は，軽量型クライアントの所有する Bitcoin アドレスが増加するとリクエストメッセージのサイズが大きくなることが挙げられる．二つ目は，完全型クライアントは軽量型クライアントが所有する Bitcoin アドレスを知ることができるため，購買習慣漏洩などのプライバシ問題につながる可能性がある．これらの問題を解決するため，ブルームフィルタに軽量型クライアントの管理する Bitcoin アドレスを入力し，それを完全型クライアントに送信する手法が提案されている [7]．ブルームフィルタは確率的なデータ構造であり，ある要素がある集合に含まれるかどうかを高速に検査することができる [8]．その際に，軽量型クライアントが実際には入力していないいくつかの偽陽性の要素が現れる．完全型クライアントはチェックポイントから最新ブロックまでの間のブロックに含まれる全ての Bitcoin アドレスを検査し，ブルームフィルタに陽性の Bitcoin アドレスを含むトランザクションのみを軽量型クライアントへ転送する．ブルームフィルタは偽陽性の要素を発生させるため，どの Bitcoin アドレスが軽量型クライアントの所有する Bitcoin アドレスか区別することが難しくなる [7]．しかしながら，従来，ブルームフィルタ設計時に一定の目標偽陽性率を設定しているため，時間経過に対してプライバシ保護レベルが一定でない．これは，軽量型クライアントがプライバシメトリックに基づいてブルームフィルタを設計していないことが原因である．

そこで本稿では，γ-Deniability に基づくプライバシを考慮した軽量型クライアントの承認済み取引確認法を提案する．γ-Deniability は，ブルームフィルタに入力した要素が偽陽性の要素によってどれほど隠されるかを定量化するプライバシメトリックである [9]．軽量型クライアントのブルームフィルタ設計時に γ-Deniability をプライバシメトリックとして用いることにより，パラメータ γ によってプライバシ保護レベルを制御可能である．特定の γ を満たすブルームフィルタを設計するために，軽量型クライアントが Bitcoin アドレスを生成した時刻であるチェックポイントから最新のブロックまでの間に存在するユニークな Bitcoin アドレス総数 N_u を知る必要がある．しかし，軽量型クライアントはブロックチェーンを保持しないため N_u を知ることができない．そこで，線形回帰モデルを用い，軽量クライアントが N_u を推定する手法を提案する．ブロックチェーンの実データの調査に基づき，線形単回帰を用いることで，チェックポイント以降に出現したユニークな Bitcoin アドレス総数を高い精度で予測できることが分かった．N_u の増加率は，時間経過に伴い大きくなっていくことが分かった．そのため，線形回帰モデルの係数は定期的に更新される必要があるが，本提案が適用された軽量型クライアントのウォレット開発者によって定期的に更新される．提案による $\hat{N}_u$ の推定精度を評価するため，N_u と $\hat{N}_u$ の平均誤差率を求める．さらに提案による一定のプライバシレベルを達成できるかを示すために，γ を評価する．特製評価により，N_u を学習期間 4 週間の線形単回帰によって推定した場合，直前のチェックポイントから 1 週間経過後，平均推定誤差率は 0.062 以下になる．また，提案によって推定された N_u を用いることで，直前のチェックポイントから 1 日経過したとき，指定された γ を達成可能であることを示す．

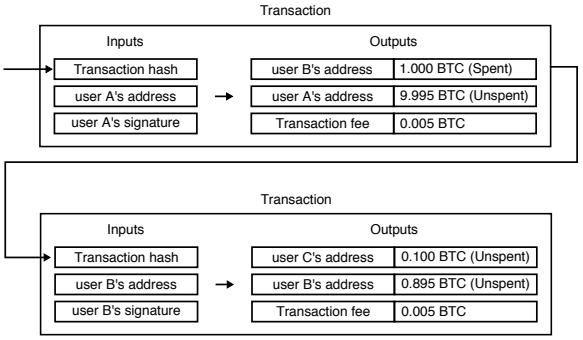


図 1 Bitcoin におけるトランザクションの例.

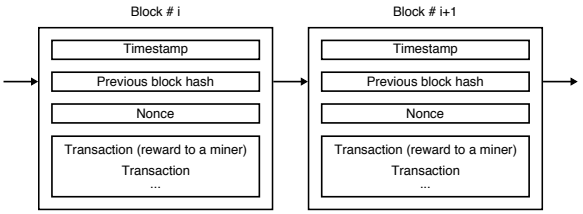


図 2 Bitcoin における連続するブロックの例.

以下，2 章では準備として Bitcoin の基礎，軽量型クライアントとブルームフィルタ，軽量型クライアントのプライバシ問題について説明する．3 章では，提案について説明する．4 章では，提案の特性評価の結果および考察を述べる．最後に 6 章では本稿の結論を述べる．

2. 準備

2.1 Bitcoin の基礎

Bitcoin ではトランザクションと呼ばれるメッセージにより送金を行う．図 1 に Bitcoin におけるトランザクションの例を示す．ユーザ A がユーザ B に 1.0 BTC を送金し，その後にユーザ B がユーザ C に 0.1 BTC を送金している．Bitcoin を送金するには，ユーザはトランザクションの unspent であるアウトプットアドレスを参照する形でインプットアドレスとして指定し，新たなトランザクションを生成する．ユーザは，unspent なアウトプットアドレ

スで指定されている公開鍵に対応する秘密鍵を用いて，トランザクションに署名する．これにより，そのユーザが実際に unspent な Bitcoin を所有していることが証明される．図 1 の場合，ユーザ A は，ユーザ A の公開鍵から計算された Bitcoin アドレスと A の秘密鍵により計算される署名を input 側に指定する．アウトプット側には，送金したい金額と送金先であるユーザ B の Bitcoin アドレスを指定する．また，ユーザ A はおつりを受け取るための Bitcoin アドレスを指定する．これにはユーザ A が既に所有している Bitcoin アドレスか，ユーザ A によって新たに生成された Bitcoin アドレスが使われる．ユーザ B がユーザ C に Bitcoin を送金する場合は，ユーザ B は，ユーザ A から受け取った Bitcoin をインプットとして指定する．ブロックチェーンを参照することで全トランザクションを確認できるため，ユーザは同じ Bitcoin アドレスを繰り返し使うことは推奨されない．プライバシ問題の影響を軽減するため，新たなトランザクションを発行する毎に新たな Bitcoin アドレスを生成することが推奨されている．

Bitcoin のトランザクションは P2P ネットワーク上でブロードキャストされる．トランザクションは，P2P 上のノードによって，そのインプットが過去に使われていないかどうか，また署名が正当であるかどうか検証される．Bitcoin は分散台帳技術を用いているため，コンセンサスアルゴリズムを用いて全ユーザが，承認されたトランザクションを含む台帳を共有する必要がある．Bitcoin では，Proof-of-Work によってコンセンサスを得る．図 2 に，ブロックチェーンの中のブロックの例を示す．マイナーと呼ばれる P2P ネットワーク上のノードがブロックを生成する．ブロックを生成するためには難易度の高い計算パズルを解く必要がある一方で，そのパズルの解は容易に検証できる．より具体的には，マイナーは目標値を下回るようなハッシュ値を生成するためのナンスを発見する必要がある．そのハッシュ値は，前のブロックヘッダ，未承認のトランザクション集合を含む．新たなブロックを生成するためには前のブロックが必要となるため，ブロックはチェーンとなる．計算パズルを最初に解いたマイナーにはブロックに含まれるトランザクション手数料と報酬が与えられるため，マイナー達はこのプロトコルに従う．

2.2 軽量型クライアントとブルームフィルタ

しかしながら，2017 年 8 月現在，Bitcoin のブロックチェーンのサイズは 100GB を超える [*1]．スマートフォンのようなストレージに制限のあるデバイス上にブロックチェーンを保持させるのは現実的とは言えない．この問題を解決するために，SPV (Simplified Payment Verification) クライアントと呼ばれる軽量型クライアントが提案されている [4]．軽量型クライアントはブロックチェーンのトランザクションを保持せず，ブロックヘッダのみを保持する．ブロックヘッダとはブロックの識別子であり，前のブロックハッシュ，タイムスタンプ，ナンス，マークルルートにより構成される．軽量型クライアントは完全型クライアントに，チェックポイント以降に存在し，軽量型クライアントが管理する Bitcoin アドレスを含むトランザクションを送信するように要求する．チェックポイントは，軽量型クライアントが最初に Bitcoin アドレスを生成した時刻が設定され，それ以降，新しく Bitcoin アドレスが生成される度に更新される．軽量型クライアントは自身の Bitcoin アドレスを含むトランザクションを受け取るため，自身の Bitcoin アドレスとチェックポイントを完全型クライアントに伝える必要がある．しかし，これには 2 つの問題がある．1 つ目は，軽量型クライアントが所有する Bitcoin アドレスの数が増加すると，リクエストメッセージのサイズも大きくなることである．2 つ目は，完全型クライアントに Bitcoin アドレスを教えることによる購買習慣の漏えいといったプライバシの問題がある．これらの問題を解決するため，Bitcoin アドレスを直接伝える代わりに，所有する Bitcoin アドレスを入力したブルームフィルタを送信する手法が提案された [7]．この手法では，軽量型クライアントが完全型クライアントにトランザクションの転送を要求する際に，軽量型クライアントは自身の Bitcoin アドレスを含むブルームフィルタを生成し，それを，接続している完全型クライアントへ送信する．ブルームフィルタは確率的なデータ構造であり，ある要素がある集合に含まれるかどうか高速に検証する際に使われる [8]．ブルームフィルタは m ビットの配列で構成され，m は以下のように計算される．

$$m = -\frac{n \ln(P_t)}{(\ln(2))^2} \tag{1}$$

ここで n はブルームフィルタに入力される Bitcoin アドレスの最大数であり，P_t は目標偽陽性率である．n は，新たに生成される Bitcoin アドレス 50 個分を考慮して $n = n_a + 50$ とするのが一般的である [10]．同様に，多くの軽量型クライアントでは，P_t の値は固定されている ($P_t = 0.1\%$ [10])．軽量型クライアントは自身の Bitcoin アドレスを k 個のハッシュ関数 $H_i(\cdot) : \{0,1\}^* \to [0, m-1]$ $1 \leq i \leq k$ に入力し，出力値に対応する m ビット配列のインデックスに 1 を立てる．ハッシュ関数の数 k は以下のように計算される．

$$k = \ln(2)\frac{m}{n} \tag{2}$$

完全型クライアントがブルームフィルタを受け取ると，直近のチェックポイント以降に生成されたブロックに軽量型クライアントの Bitcoin アドレスを含むトランザクションが存在するかどうかを検証する．それが存在する場合，そ

[*1] https://blockchain.info/ja/charts/blocks-size (2017/08/25 アクセス．)

の陽性となったトランザクションとマークル木を軽量型クライアントに送信する．マークル木 [11] はある特定のトランザクションがあるブロックに本当に含まれているかを確認するために使用される．マークル木は二分木構造であり，ノードはハッシュ値を持つ．それぞれのノードのハッシュ値は連結された子ノードの値をハッシュ化した値である．軽量型クライアントが，あるトランザクションがブロックに含まれているか確認するには，そのトランザクション，ブロックヘッダ，マークル木を用いてハッシュ値がブロックヘッダに含まれるマークルルートと一致することを確認する．

2.3 軽量型クライアントのプライバシ問題

ブルームフィルタはいくつかの偽陽性のアドレスを生むため，完全型クライアントは陽性となった Bitcoin アドレスのいずれかが軽量型クライアントによって所有されているかを判別することは困難である [7]．しかしながら，軽量型クライアントがブルームフィルタに入力した Bitcoin アドレスは高い確率で特定される可能性があると指摘されている [12]．これは目標偽陽性率 P_t を固定した場合，偽陽性の Bitcoin アドレスによって達成されるプライバシレベルはブルームフィルタで検証される Bitcoin アドレス数に依存するためである．つまり，偽陽性 Bitcoin アドレスによって達成されるプライバシレベルは，目標偽陽性率だけでなく，チェックポイントから最新のブロックまでに含まれるユニークな Bitcoin アドレス総数 N_u に依存する．例として，目標偽陽性率が $P_t = 0.1\%$ のブルームフィルタを考える．$N_u = 10^5$ のとき，そのブルームフィルタは平均して 100 個の偽陽性 Bitcoin アドレスを生成すると考えられる．一方，$N_u = 100$ の場合，偽陽性の Bitcoin アドレスは平均で 1 つとなる．したがって，N_u を考慮せずに P_t を特定の値に固定した現在のブルームフィルタ設計は，一定のプライバシレベルを提供できない．Bitcoin では，1 ブロック当たりのユニークな Bitcoin アドレス総数は徐々に増加している．また，それによって時間経過に伴い N_u も増加する．したがって，軽量型クライアントに適したブルームフィルタの設計基準が必要である．

3. 提案

本論文では，γ-Deniability に基づく軽量型クライアントのプライバシ保護を考慮したブルームフィルタ設計手法を提案する．γ-Deniability は，軽量型クライアントが入力した Bitcoin アドレスが偽陽性 Bitcoin アドレスによってどれだけ保護されるかを示す指標である．γ-Deniability を適用するためには，軽量型クライアントはチェックポイントから現れたユニークな Bitcoin アドレス総数 N_u を知る必要がある．しかし，軽量型クライアントはブロックチェーンを保持しないため，N_u を知ることができない．この問題

	B[1]	B[2]	B[3]	B[4]	B[5]	B[6]	B[7]	B[8]	B[9]
x_1	1	0	1	0	0	0	0	1	0
x_2	1	0	0	0	0	0	0	1	1
x_3	0	0	0	1	0	1	0	0	1
$BF\ (x_1, x_2, x_3)$	**1**	**0**	**1**	**1**	**0**	**1**	**0**	**1**	**1**
v_1	1	0	1	0	0	0	0	0	1
v_2	0	0	1	0	0	1	0	1	0
v_3	1	0	0	0	0	1	0	0	1

図 3 3 つの真陽性要素と 3 つの偽陽性要素を含むブルームフィルタの例．

を解決するため，線形回帰を用いた N_u の推定手法を提案する．線形回帰の係数は，1 週間など定期的に更新される必要があり，この処理は軽量型クライアントのアプリケーション開発者によって行われる．

以下では，(i) どのように γ-Deniability を軽量型クライアントのブルームフィルタ設計に適用させるか，また (ii) 線形回帰を用いて N_u を推定する方法について述べる．

3.1 γ-Deniability

γ-Deniability はブルームフィルタに入力された真陽性の要素が，偽陽性の要素によってどれほど隠されるかを示すプライバシメトリックである [9]．ブルームフィルタに入力された Bitcoin アドレス x は，$\exists j \in \{1...k\}$ s.t. $H_i(x) = H_j(v)$ のような偽陽性の Bitcoin アドレスが少なくとも 1 つあれば，deniable と言える．Bitcoin アドレス x が γ の確率で deniable であるとき，そのブルームフィルタは γ-deniable であると言う．m-bit のブルームフィルタの γ は，以下のように計算される．

$$\gamma \approx \left(1 - \exp\left(-\frac{lk}{m(1 - e^{\frac{-kn}{m}})}\right)\right)^k \tag{3}$$

ここで，$l = (N_u - n)P_t$ であり，平均的に出現する偽陽性アドレス数を表す．式 (3) は以下のように近似できる．

$$\gamma \approx \left(1 - \exp\left(\frac{2(N_u - n)P_t \ln 2}{n}\right)\right)^{-\ln P_t / \ln 2} \tag{4}$$

γ-Deniability の理解を容易にするために，以下の例を考える．図 3 に，3 つの真陽性要素と 3 つの偽陽性要素を含むブルームフィルタの例を示す．$< x_1, x_2, x_3 >$ は実際にブルームフィルタに入力された Bitcoin アドレスであり，$< v_1, v_2, v_3 >$ は偽陽性の Bitcoin アドレスである．Bitcoin アドレス x_1 は，ビット配列 $B[1], B[2], B[3]$ が v_1 と v_2 によって隠されているので，完全型クライアントは，Bitcoin アドレス x_1 が軽量型クライアントによって所有されていると分からない．Bitcoin アドレス x_2 についても同様のことが言える．一方，Bitcoin アドレス x_3 は，ビット配列 $B[4]$ がどの偽陽性アドレスからも隠されていないため，deniable ではない．この例では，3 つのうち 2 つの Bitcoin

アドレスが deniable であるため $\gamma = 0.66$ となる．γ が 1 に近づく程，高いプライバシ保護を達成していると言える．

3.2 N_u の推定手法

本提案では，N_u を線形回帰を用いて推定する．すなわち，$N_u \approx f(x) = ax^n + bx^{n-1} + \cdots$ といった線形モデルによって回帰する．$f(x)$ の係数は，N_u の真値との誤差が最小になるように最小二乗法により決定される．軽量型クライアントはブロックチェーンを保持しないため N_u の真値を得られず，$f(x)$ の係数を計算できない．そのため，この処理はアプリケーション開発者によって行われる．係数の更新は定期的に行われる必要があり，1 週間に 1 回程度で十分であることを後述する．

3.3 軽量型クライアントのブルームフィルタ生成の流れ

軽量型クライアントは以下の流れでブルームフィルタを生成する．

(1) 軽量型クライアントは，アプリケーション開発者より定期的に近似モデルを与えられる．
(2) 軽量型クライアントは前回のチェックポイントからの経過時間 T を計算し，与えられた線形回帰モデル $N_u \approx f(T)$ を用いて $\hat{N_u}$ を推定する．
(3) N_u と目標とする γ を式 (4) に代入し，ブルームフィルタのサイズ m を計算する．

4. 特性評価

提案の有効性を示すため，(i) チェックポイントからある時刻までのユニークな Bitcoin アドレスの推定総数の平均誤差率，(ii) 達成される γ，の 2 つのメトリックを，ブロックチェーンの実データを用いたシミュレーションにより評価する．本シミュレーションは Ubuntu 14.04 上で実行する．また，ブロックチェーンをダウンロードするために *bitcoind* を，Bitcoin トランザクションや Bitcoin アドレスを抽出するために *blockparser* [13] を用いた．評価のために，2014 年 1 月 1 日から 2015 年 12 月 31 日までのブロックを使用した．以下のシナリオに沿って各メトリックを評価する．最初に，評価に用いたブロック内からチェックポイントをランダムに 1 つ選択する．次に，選択されたチェックポイント以降に存在する Bitcoin アドレスを 100 個取り出し，それらがある軽量型クライアントのユーザが所有していると仮定し，それらの Bitcoin アドレスを含むブルームフィルタを生成する．提案の場合，ブルームフィルタを生成する際の N_u は線形回帰モデルによって推定される．そして，推定された N_u と指定された γ を使って P_t を計算する．本シミュレーションでは γ0.995 とする．従来では，P_t は 0.1%に固定される．チェックポイント以降のブロックに含まれる全ての Bitcoin アドレスはブルームフィルタにより検証され，γ が計算される．また，推定 N_u と実測 N_u の誤差率 r_{error} は以下のように計算される．

$$r_{\text{error}} = \frac{|N_u - \hat{N_u}|}{N_u} \tag{5}$$

特に明記しない限り，上記の手順を 100 回繰り返し，両方の特性メトリックをそれぞれ平均化する．

4.1 N_u の推定精度

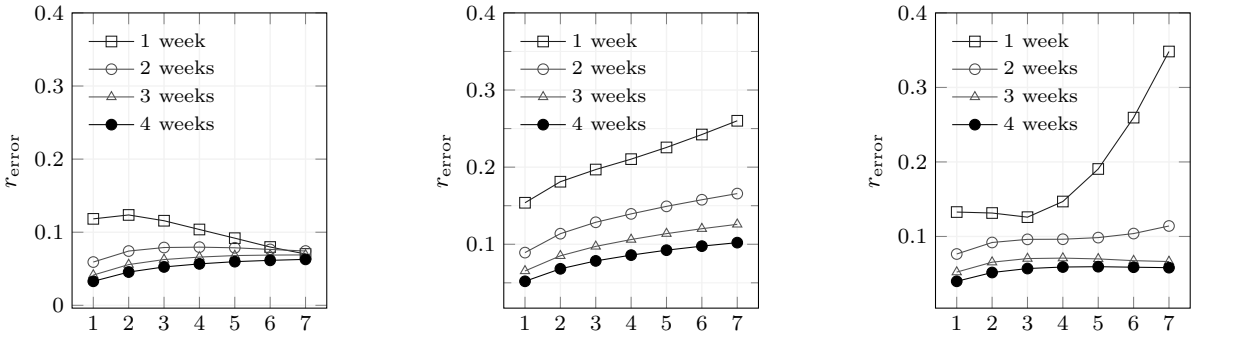
線形回帰モデルと N_u の推定精度について議論する．(i) $N_u = ax + b$, (ii) $N_u = ax^2 + bx + c$, (iii) $N_u = ax^3 + bx^2 + cx + d$ の 3 つの線形回帰モデルを評価する．N_u をカウントする学習期間を 1 週間から 4 週間の間で変化させる．線形回帰モデルの係数を決定するために最小二乗法を用いる．図 4 に，3 つの線形回帰モデルの平均 r_{error} を示す．この図より，2 つのことが分かる．1 つ目に，いずれの線形回帰モデルにおいても，学習期間が長くなると平均推定誤差は小さくなることが分かる．例えば，学習期間が 4 週間の線形単回帰モデルの係数の場合，チェックポイントから 7 日経過後は平均 r_{error} が 0.062 である．一方，学習期間が 1 週間の $N_u = ax^3 + bx^2 + cx + d$ のモデルの場合は，r_{error} が 0.348 である．2 つ目に，線形単回帰モデル $N_u = ax + b$ を用いた場合，r_{error} が最小となることがわかる．この結果から，ブロック内に存在するユニークな Bitcoin アドレス総数は 1 週間程度の期間ではほぼ線形に増加することが分かる．このことから，2014 年から 2015 年の期間では，係数 a と b は 1 週間に一度更新されれば十分であると言える．N_u の推定手法が 2016 年，2017 年など他の期間でも有効かどうか，また，1 週間以上の長期間の場合でも有効かどうかの検証は今後の課題である．

4.2 γ の評価

次に γ について議論する．この評価では，4 週間分のトランザクションを基に推定した N_u を用いての線形単回帰モデルにより $\hat{N_u}$ を推定する．図 5 にチェックポイントからの経過時間に対する γ を示す．この図から分かるように，提案の γ は従来のそれよりも常に高い．これは，提案ではブルームフィルタを生成する際に N_u を考慮しているからである．さらに，従来によって得られた γ はチェックポイントから半日経過した時点で 0.6 を下回っている．これより，偽陽性率 $P_t = 0.1\%$ と設定することは評価に用いたブロックの期間では適切ではないと言える．この結果より，γ-Deniability をプライバシメトリックとして導入することにより従来より高いプライバシレベルを提供できることが分かった．

5. 結論

本稿では，Bitcoin の軽量クライアントのプライバシ保護を図るため，γ-Deniability に基づく承認済み取引確認手法を提案した．従来法では，ブルームフィルタを用いるこ



(a) $N_u \approx ax + b$　(b) $N_u \approx ax^2 + bx + c$　(c) $N_u \approx ax^3 + bx^2 + cx + d$

図 4　3 つの線形回帰モデルの平均推定誤差率 r_{error} の比較.

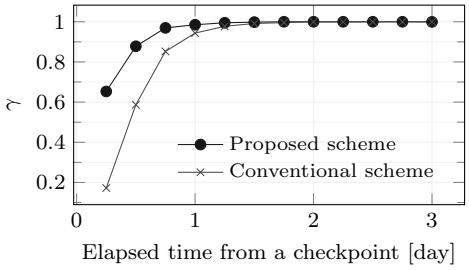


図 5　直前のチェックポイントからの経過時間に対する得られた γ.

とで軽量型クライアントのプライバシ保護を図るが，プライバシメトリックに関して議論されていないという問題があった．そこで，γ-Deniability をプライバシメトリックとして導入し，プライバシ保護を図るブルームフィルタを設計する方法について議論した．最後のチェックポイントからのユニークな Bitcoin アドレス総数 N_u が必要であるため，線形回帰を用いる N_u の推定法を提案した．ブロックチェーンの実データを用いたシミュレーションにより，4 週間分のトランザクションを基に学習した線形回帰モデルにより推定された N_u を用いることで，平均推定誤差率が 0.062 以下になることが分かった．また，提案法により推定された N_u を用いることで，提案の γ は従来よりも常に高くなることが分かった．

参考文献

[1] L. Kristoufek, "BitCoin meets Google Trends and Wikipedia: Quantifying the relationship between phenomena of the Internet era," *Scientific reports*, vol. 3, 2013.

[2] M. Van Alstyne, "Why Bitcoin has value," *Communications of the ACM*, vol. 57, no. 5, pp. 30–32, 2014.

[3] P. Ciaian, M. Rajcaniova, and d. Kancs, "The economics of BitCoin price formation," *Applied Economics*, vol. 48, no. 19, pp. 1799–1815, 2016.

[4] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[5] T. Bamert, C. Decker, L. Elsen, R. Wattenhofer, and S. Welten, "Have a snack, pay with Bitcoins," in *Proc. of IEEE International Conference on Peer-to-Peer Computing (P2P)*. IEEE, 2013, pp. 1–5.

[6] R. Jamie, "The evolution of the bitcoin vending machine," https://news.bitcoin.com/evolution-bitcoin-vending-machine/, July 2016, (Accessed on May 29 2016).

[7] M. Hearn and M. Corallo, "BIP37: Connection Bloom filtering," 2012, https://github.com/bitcoin/bips/blob/master/bip-0037.mediawiki.

[8] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," 1970.

[9] G. Bianchi, L. Bracciale, and P. Loreti, ""Better Than Nothing" Privacy with Bloom Filters: To What Extent?" in *Proc. of International Conference on Privacy in Statistical Databases*. Springer, 2012, pp. 348–363.

[10] "Bitcoinj," http://bitcoinj.github.io/.

[11] R. C. Merkle, "Protocols for public key cryptosystems," in *Proc. of IEEE Symposium on Security and Privacy (SP)*. IEEE, 1980, pp. 122–122.

[12] A. Gervais, S. Capkun, G. O. Karame, and D. Gruber, "On the privacy provisions of Bloom filters in lightweight Bitcoin clients," in *Proc. of ACM Annual Computer Security Applications Conference (ACSAC)*. ACM, 2014, pp. 326–335.

[13] M. Spagnuolo, F. Maggi, and S. Zanero, "BitIodine: Extracting intelligence from the Bitcoin network," in *Proc. of Financial Cryptography and Data Security (FC)*. Springer, 2014, pp. 457–468.