マイナンバーカードを用いたブロックチェーンによる自動確定日付システム

芳賀 慎也[1,a)]　面 和成[1]

概要：近年，公証制度のオンライン化が進んでいる．しかし，日本においてオンライン申請は存在するものの，最終的に手数料の納付や電子署名を受けたファイルを受け取りに直接公証人がいる公証役場に赴く必要がある．このような制度では，依頼者が利用しやすいとは言えない．本稿では，マイナンバーカードとスマートコントラクトを組み合わせて，自動化された確定日付のシステムを提案する．具体的には，マイナンバーカードで署名された電子書類に対して作成者と書類自体の認証を自動的に行い，その情報がブロックチェーンに格納される際に生成されるトランザクションレシートを確定日付の証明書として利用することで，より信頼性のある確定日付のシステムを構築する．従来，指定公証人が行っていた業務の一部をスマートコントラクトとしてブロックチェーン上で実行することで，書類作成から依頼者が入手するまでの時間を大幅に削減する．さらに，このシステムをスマートフォンのアプリケーションと Ethereum ブロックチェーン上で実証した．

キーワード：ブロックチェーン，スマートコントラクト，マイナンバーカード，公的個人認証サービス，確定日付

Blockchain-Based Autonomous Notarization System Using National IC Card

Shinya Haga[1,a)]　Kazumasa Omote[1]

Abstract: In recent years, the notarization system is going online. In Japan, there is a system to make the application process online. However, it is necessary to go to the notary office in order to pay the fee and receive the digitally signed file. It is not easy for the client to use the system. In this paper, we propose an automated system of fixed dates by combining a national IC card and smart contracts. Specifically, our proposed system automatically authenticates the creator and the document for electronic documents signed with a national IC card, and uses the transaction receipt generated when the information is stored in the blockchain as a certificate of the fixed date. By executing some of the tasks traditionally performed by designated notaries as smart contracts on the blockchain, the time between the creation of the document and its acquisition by the client will be significantly reduced.

Keywords: Blockchain, Smart Contract, National IC Card, Notarization System

1. はじめに

近年，公証制度のオンライン化が世界各地で進んでいる．例えば，アメリカにおいて，公証人とオンラインで接続し，デジタルな環境で公証を受けることができる Notarize [1] という事業が注目を集めている．電子政府で有名なエストニアにおいては，公証人が e-Notary [2] というシステムを使うことで，オンライン上で公証することができるような制度が存在する．

しかし，日本における公証制度のオンライン化は進んでいないと言わざるえない．電子公証制度と言われる制度は

[1] 筑波大学大学院
University of Tsukuba
[a)] s2120545@s.tsukuba.ac.jp

存在し，確定日付の申請におけるオンライン化は進んでいるが，電子書類であっても，最終的に手数料の納付や電子署名を受けたファイルを受け取りに直接公証人がいる公証役場に赴く必要がある．このような制度では，依頼者が利用しやすいとは言えない．一方で，近年の ICT 技術の発展に伴い，法務省が関連法改正を視野に入れつつ，公正証書のデジタル化に乗り出すなど，改善に積極的な姿勢が見られる [3].

本稿では，地方公共団体情報システム機構から発行されるマイナンバーカードによる本人性の確認とスマートコントラクトの第三者を介さないプログラムとしての契約を自動履行する仕組みを用いることで，公証人を必要としない自動化された確定日付のシステムを提案する．具体的には，マイナンバーカードを用いて署名された電子書類に対してスマートコントラクト内で検証を行うことで，署名者と電子書類自体の認証を自動的に行う．検証結果が正しい場合，その情報はブロックチェーンに格納されるが，その際ブロックチェーンプラットフォームが発行するトランザクションレシートを確定日付の証明書として利用することで，信頼性の高いシステムを構成する．従来，日本において，法務大臣に指定された指定公証人が行っていた業務の一部をスマートコントラクトとしてブロックチェーン上で実行することで，書類作成から依頼者が入手するまでの時間を大幅に削減する．さらに，本システムをスマートフォンのアプリケーションと Ethereum ブロックチェーン上で実証し，実現可能であることを示す．

2. 準備

2.1 ブロックチェーン

ブロックチェーン [4] とは，サトシ・ナカモトにより考案された分散台帳技術であり，このアルゴリズムは Bitcoin を含む多くの暗号資産の中核をなす技術である．ブロックチェーンは Peer to Peer（P2P）ネットワーク上で発生したトランザクションと言われる取引データを，タイムスタンプや前のブロックのハッシュ値などとともにブロックにまとめることで，鎖の連なりのようにデータを保持する．トランザクションをまとめ，ブロックを生成することができたノードは報酬を得ることができ，その一連の流れをマイニングという．マイニングできるノードは一つと定められており，そのノードを決める方法として Bitcoin, Ethereum[*1]などの暗号資産では時間のかかる計算を成功したものがマイニングできる Proof of Work（PoW）と言われるプロトコルを採用している．また，Bitcoin においては，ブロックチェーンの主鎖は最初のブロックから現在のブロックまでの最長のものと定められている．そのため，ブロックに格納されたデータを改竄するためには，それ以降のブロックを全て破棄し，それまで PoW で計算されてきた膨大な計算を行わなくてはならないため，現実的に不可能とされている．ゆえにブロックチェーンは耐改竄性を持つと言われる．また，パブリックブロックチェーンにおいては，ネットワークに誰でも参加することができ，ブロックに格納されているデータを見ることができるため，透明性が高く，誰でも追跡が可能であると言われる．

2.2 スマートコントラクトと Ethereum

スマートコントラクトとは，ブロックチェーン上に記録されているプログラムであり耐改竄性を持つ．そのプログラムが実行されるとその記録もブロックチェーンに格納されるため，一連の動作の透明性を確保することができる．また，P2P ネットワーク上で実行されるため，単一障害点となるものは基本的に存在しない．この特性ゆえに，第三者機関を介さずに，任意の契約を自動で実行できると言われている．

スマートコントラクトを最初に採用したブロックチェーンプラットフォームとして Ethereum [5] [6] があり，Ethereum においては，EVM バイトコードの形で保存される．本研究においてスマートコントラクトを記述する際は，高級言語である Solidity を統合開発環境（IDE）の Remix-IDE[*2]を用いて記述し，EVM バイトコードにコンパイルした．

Ethereum は ECDSA を採用しており，そのアカウントのアドレスは，公開鍵のハッシュ値をとるなどして作られる．このアカウントは EOA (Externally Owned Account) と言われ，ユーザーは EOA を通してスマートコントラクト内の関数を実行する．

スマートコントラクト内の関数を実行するためには，EOA がトランザクションを発行する必要があり，その際に Gas と言われる手数料がマイナーに徴収される．その値は，一般的にスマートコントラクトを実行する際の計算量に比例して大きくなる．Gas の概念により，悪意ある攻撃者が計算量が多くなるようなトランザクションを Ethereum ネットワーク上に展開する DoS 攻撃を実行しようとすると膨大な Gas がかかるため，攻撃者は Ethereum への DoS 攻撃のインセンティブを失うことになる．

また，トランザクションが実行されると，実行結果がまとめられたトランザクションレシートが発行される．トランザクションを実行した EOA や実行のログなどがトランザクションレシートの中に書き込まれ，各ノードに保存される．

2.3 公的個人認証サービスとマイナンバーカード

公的個人認証サービス [7] とは，オンラインで行政手続きを行う際の本人確認手段であり，電子証明書を用いて行

[*1] Ethereum については，2021 年 7 月現在，PoW から Proof of Stake（PoS）に移行中である

[*2] Remix, “Remix -Ethereum IDE,” 2021, https://remix.ethereum.org/

表 1　カード AP の種類

カード AP	用途
公的個人認証 AP (JPKI-AP)	電子証明書によるユーザー認証や電子署名を行う
券面 AP	カード券面の画像データなどを保持する
券面入力補助 AP	マイナンバーや基本 4 情報をテキストで保持する
住基 AP	住民票コードを保持する
その他 AP	空き領域に自治体が独自の業務 AP を搭載可能

われる．地方公共団体情報システム機構（J-LIS）が電子証明書をマイナンバーカードと言われる耐タンパー性の高い IC カードに記録し，発行することで，カードを用いたユーザー認証や申請書類に電子署名を施すことが可能になる．電子証明書は，ユーザー認証用の**利用者証明用電子証明書**と電子署名用の**署名用電子証明書**の 2 種類が存在し，基本 4 情報[*3]は署名用電子証明書のみ含む．また，公的認証法の改正により，2016 年 1 月から民間での利用が可能になり，総務大臣の許可を得た民間事業者もマイナンバーカードを用いたユーザー認証などを行うことができる．2021 年 5 月時点において，総務大臣認定事業者は 14 社であり，その事業者のシステムを利用し，サービスを提供する民間企業は 113 社に達する [8].

マイナンバーカードの技術的仕様 [9] については，接触型と非接触型の両方のインターフェースを有する IC カードであることが挙げられる．接触インターフェースは，ISO/IEC 7816 に準拠し，一般的な接触型 IC カードリーダーで利用可能である．非接触インターフェースは，ISO/IEC 14443 Type B に準拠しており，Near Field Communication (NFC) 対応の IC カードリーダーで利用可能である．

また，マイナンバーカードの IC チップには，IC カード用の OS が備わっており，主に 4 つのカードアプリケーション（カード AP）[10] を構成する．その種類と用途を表 1 に示す．利用者証明用電子証明書と署名用電子証明書は JPKI-AP 内に備わっており，取得する際にはマイナンバーカード交付時に設定したそれぞれ 4 桁の数字，6~16 桁の英数字の暗証番号 (PIN) を使用する必要がある．加えて，任意の電子書類に対して，証明書に記載されている公開鍵と対応する秘密鍵で電子署名を付すことも可能である．しかし，PIN 入力を利用者証明電子証明において 3 回，署名用電子証明書において 5 回間違えるとロックがかかり，その解除は地方自治体の窓口のみ可能であるため，パスワード攻撃に対して耐性を持つ．

2.4　確定日付

通常私人が作成する文書は作成日付を偽装することが容易である場合が多い．二者間で結ぶ契約書の類で合っても，二人が共謀することで，あたかも過去に作成した書類であることに偽装することも可能である．そのため，文章に対して法的効力を持たせる制度を確定日付と呼ぶ．この制度を利用することで，文書作成日付に関して言い争いが起こった場合に速やかにその証明が可能になる．

紙ベースの証書においては，公証役場で法務大臣により指定された指定公証人が確定日付印を押すことで，電子書類においては，電子署名を付すことで確定日付が完了する．また，希望することで確定日付を付した電子書類は 20 年間，確定日付に関するデータは 50 年間，公証役場のサーバーに保管することも可能になる [11].

以下に電子書類に確定日付を付す（電子公証制度）際の現状の手順を示す [12].

(1) 依頼人が電子証明書[*4]の取得
(2) 依頼人が私署証書を電子ファイル (PDF のみ可) で作成
(3) 依頼人が公証人に電話または FAX で連絡
(4) 依頼人が作成した電子ファイルに電子署名を付す
(5) 登記・供託オンライン申請システム[*5]を使用し，電子証明書と電子ファイルを公証人に送信
(6) 依頼人が公証役場に赴き，公証人から面前で電子署名を行なったか確認を受ける
(7) 公証人による審査の結果問題がない場合は，依頼人が手数料を払うことで，交渉人が電子ファイルに電子署名を付す
(8) 依頼人は公証人による電子署名がなされた電子ファイルを依頼人が持参した電子媒体で受け取る[*6]

また，確定日付の対象となる文書の必要要件 [13] として以下の 3 つがあげられる．確定日付を付す際の手順 (7) において，公証人は次の 3 つの要件を審査している．

(1) 私文書に限られる
(2) 私文書は、文字その他の記号により、意見、観念または思想的意味を表示しているものであることが必要
(3) 作成者の署名又は記名押印のあるものでなければならない

3.　関連研究

3.1　ブロックチェーンと権利に関する研究

Meng らの研究 [14] では，ブロックチェーンと電子透かしを用いてデジタル著作権保護システムを提案している．ブロックチェーンを電子透かし情報の安全な保存場所やタイムスタンプ認証として利用することで，単一サーバー型のシステムに比べ，情報漏洩やデータの改竄などのリスク

[*3] 氏名，性別，住所，生年月日の 4 つの情報

[*4] 利用可能な電子証明書は以下の 4 つ
・商業登記に基づく電子証明書
・公的個人認証サービス
・セコムパスポート for G-ID
・電子認証サービス（e-probatio PS2）

[*5] 法務省が提供するオンライン申請システム

[*6] インターネットを介して受け取ることは認められていない

を軽減することに成功した．

Chowdhury らの研究 [15] では，Personal Data Store (PDS) と言われるユーザーが個人情報を安全に管理できるサービスとブロックチェーンを組み合わせた公証システムを提案している．検証プロセスのハッシュなどをブロックチェーンに乗せることで，ユーザーのプライバシーに配慮しつつ，データの真正性を保証するデータ共有フレームワークを提案している．

3.2 ブロックチェーンと国の認証システムを組み合わせた研究

Daramola らの研究 [16] では，国が発行する IC カードとブロックチェーンを用いた大規模な国政選挙向けのシステムの提案，評価している．IC カードによる国民の認証をブロックチェーンの外で行い，その結果を Hyperledger Fabric で構成される選挙システムに組み込む仕組みを提案し，南アフリカにおいて，技術に明るくない選挙管理人とともに実際に評価することで，ブロックチェーンを用いた選挙システムに潜むリスクを明らかにした．

永田らの研究 [17] では，Bitcoin のトランザクションと公的個人認証サービスを用いて，本人性の確認を検討している．総務大臣に認定された署名検証業者を仮定し，その業者がブロックチェーンの外で実行者の本人性を検証し，実行者の本人性の根拠となる情報をブロックチェーン上に記録する方法で本人性を検討している．

3.3 関連研究の問題点

日本における，従来の電子書類における確定日付制度では，公証人が電子書類に対して審査をし，依頼人が最終的に公証役場に赴く必要があるため，時間と手間がとてもかかると言える．その問題を解決するブロックチェーンを用いた公証システムに関する既存研究も見受けられるが，署名されたデータの検証作業を，信頼機関を仮定した上でブロックチェーンの外で行うため，その信頼性に不安が残る [16] [17].

4. 提案手法

4.1 概要

日本の確定日付において，公証人による署名検証と電子署名付与をスマートコントラクトとトランザクションレシートに置き換えることにより，第三者を介さない自動化した確定日付の実現を目指す．具体的には，電子書類の本人性確認にマイナンバーカードとスマートコントラクトによる RSA 署名検証を行うことで，日本において，その書類のある時刻での存在したことと誰が署名したかを証明することができるシステムを提案する．また，行政機関はマイナンバーカードの公開鍵のハッシュ値をスマートコントラクト上に公開することを想定する．その結果，依頼主のトランザクションでスマートコントラクトが電子署名を検証する際に，その公開鍵が有効であるかを確かめることが可能となる．

4.2 構成要素

- 依頼者： 依頼者は地方自治体からマイナンバーカードを交付されており，NFC 対応のスマートフォンや特定のカードリーダーを使用することで，任意の電子書類に電子署名を付すことが可能である．また，アプリケーションなどを用いることで電子文書をハッシュ化することができるとする．
- マイナンバーカード：J-LIS から発行され，依頼者が保有するマイナンバーカード．依頼者が交付時に設定した PIN を用いることで，電子証明書の取得や電子署名の作成が可能になる．本稿においては，署名用電子証明書に記載されている公開鍵に対応する秘密鍵を用いて電子署名を行うこととする．
- スマートコントラクト：J-LIS などの行政機関が管理することを前提としたスマートコントラクト．公開鍵管理スマートコントラクトと署名検証スマートコントラクトの 2 つがある．公開鍵管理スマートコントラクトには，マイナンバーカードに登録されている公開鍵のハッシュが記録されており，もし記録されていない場合は，その鍵が無効であることを示す．署名検証スマートコントラクトには，RSA 署名検証機能が実装されており，送られてくる電子署名と合わせて，送信者がマイナンバーカードの所有者であることを確認できる．
- 行政機関（J-LIS など）：スマートコントラクトやマイナンバーカードを管理する組織．マイナンバーカードを交付する役割を持つ．また，マイナンバーカードの公開鍵を公開鍵管理スマートコントラクトのリストに追加・削除することで，鍵の失効を管理する．

4.3 手順

提案手法は以下の 3 つから構成される．その概要を図 1 に示す．

(1) 電子文書のハッシュ化：確定日付の依頼者は電子文書をハッシュ化する．
(2) マイナンバーカードによる電子署名：確定日付の依頼者はマイナンバーカードを用いて電子文書のハッシュ値に RSA 署名を施す．
(3) スマートコントラクトによる署名検証，トランザクションレシート発行：確定日付の依頼者はトランザクションを発行することで，電子文書のハッシュ値，RSA 署名，署名に用いた公開鍵を署名検証スマートコントラクトに送信する．署名検証スマートコントラクトは，公開鍵の有効性を公開鍵管理スマートコント

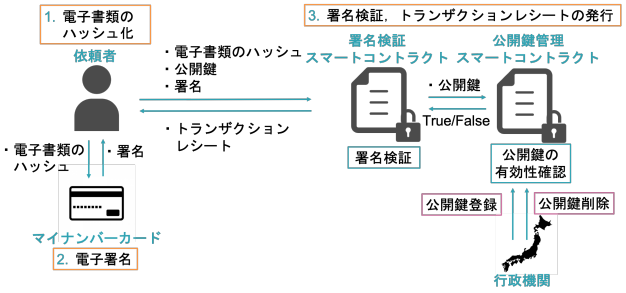


図 1　提案手法の全体図

ラクトに問い合わせ，有効だったならば，電子文書のハッシュ値，RSA 署名，公開鍵を用いて署名検証を行う．そして，検証した結果正しいならば，トランザクションレシートを発行する．トランザクションレシートが出力されてブロックチェーンに格納されることで確定日付が完了する．

確定日付が付された電子文書を法的証拠として使う際は，電子文書とともにトランザクションレシートを裁判所などに提出すればよい．なぜなら，トランザクションレシートからブロックに埋め込まれたトランザクションを特定し，その中に格納されている電子文書のハッシュ値と提出された電子文書のハッシュ値を比べることで，その電子文書がトランザクションが発行された時間に存在したことを証明することができるからである．加えて，そのトランザクションは，マイナンバーカードによって生成された電子署名が改竄不可能なプログラムであるスマートコントラクトによって実装された署名検証機能で正しいと判定された場合にのみ発行される．ゆえに，署名作成者が署名したことも示す．

これは確定日付の対象となる文書の必要要件である（3）を満たす．（1），（2）について，証拠として提出された後に，必要要件を満たしているか判定すれることができるため，問題はないと考える．

5.　実装評価

本稿ではプロトタイプとして，マイナンバーカードによる電子署名アプリケーションと電子署名を検証するスマートコントラクトの 2 つを実装する．前者は，swift を用いて IOS 上搭載のスマートフォン上で作動するアプリケーションを Xcode12.5 を用いて作成し，IOS 14.6 搭載の iPhone Xs で実証した．後者は，Solidity を用いて Ethereum のテストネットワークの 1 つである Rinkeby 上で機能するスマートコントラクトを作成した．Solidity のバージョンは 0.6.0 を使用した．

5.1　マイナンバーカードによる電子署名アプリケーション

swift を用いて IOS 上で作動するアプリケーションを作成する．主な機能として 3 つあり，以下に実装の詳細を記載する．

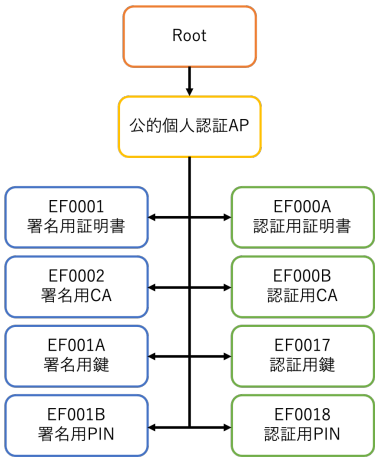


図 2　公的個人認証 AP のディレクトリ図

5.1.1　マイナンバーカードから署名用電子証明書を取得

マイナンバーカードは ISO7816 に準拠しているため，IC カードとアプリケーションとの通信には，APDU コマンド[*7]を用いることで可能になる．swift においては，**coreNFC** フレームワークを用いることで NFC で APDU コマンドを IC カードに送ることができる．

また，公的個人認証 AP の構成は図 2 のようになっているため，次に示す手順で APDU コマンドをマイナンバーカードに送信することで，署名用電子証明書を入手できる．

（1）SELECT FILE 公的個人認証 AP
（2）SELECT FILE 署名用 PIN
（3）VERIFY 署名用 PIN[*8*9]
（4）SELECT FILE 署名用電子証明書
（5）READ BINARY

なお，DER 形式で入手した署名用電子証明書はスマートフォン内部のセキュアな領域に保存する．

5.1.2　マイナンバーカードを用いて電子書類に電子署名を付す

最初に任意の電子書類のハッシュ値を取得する．次に，**6.1.1** と同様に APDU コマンドを用いて，電子書類のハッシュ値に電子署名を付し，次に示す手順で APDU コマンドをマイナンバーカードに送信することで，マイナンバーカード内の秘密鍵を用いて電子署名を作成する．

（1）SELECT FILE 公的個人認証 AP
（2）SELECT FILE 署名用 PIN
（3）VERIFY 署名用 PIN[*10]
（4）SELECT FILE 署名用鍵
（5）COMPUTE DIGITAL SIGNATURE

[*7] Application Protocol Data Unit コマンドの略．バイナリ配列のコマンドであり，カード OS にリクエストするとバイナリ配列のレスポンスが返る
[*8] マイナンバーカード交付時に設定した英数字 6~16 桁の暗証番号
[*9] 暗証番号の検証により，セキュリティステータスを更新することで署名用電子証明書にアクセス可能になる
[*10] *9 と同様の暗証番号

なお，入手した電子署名はスマートフォン内部のセキュアな領域に保存する．

5.1.3 スマートコントラクトに電子署名などを送信

6.1.1 で入手した署名用電子証明書から公開鍵を取り出し，**6.1.2** で入手した電子署名と電子書類のハッシュ値とともにスマートコントラクトへ送信する．この際，Infura と言われる Ethereuem のノードホスティングサービスを利用して，テストネットワーク Rinkeby に接続する．スマートコントラクトでの検証に成功すると，そのトランザクションレシートを受け取り，スマートフォン内部のセキュアな領域に保存する．

5.2 スマートコントラクト

Solidity を用いて，Ethereum で動作するスマートコントラクトを作成する．電子署名を検証する署名検証スマートコントラクトと公開鍵を管理する公開鍵管理スマートコントラクトの 2 種類で構成される．以下に実装の詳細を記述する．

5.2.1 署名検証スマートコントラクト

実際に署名検証を実行するスマートコントラクトであり，RSA 署名検証を行う関数 **RSA_verify** で構成される．**RSA_verify** は，まず，公開鍵の失効情報を検証するため，公開鍵管理スマートコントラクトの関数である **Find-Key** に対してインターナルトランザクションを利用して，公開鍵の有効性を問い合わせる．公開鍵が有効であった場合にのみ，RSA 署名検証を実行する．

マイナンバーカードの署名認証方式としては 2048bit の RSA 署名が採用されている [18]．ゆえに本稿では，パッディングとして PKCS#1 SHA256 を使用し，スマートコントラクト内の関数で RSA 署名検証を実装する．しかし，2048bit の RSA 署名において，署名検証の計算量がかなり多いため，Ethereum 上のスマートコントラクトで実装し，実行しようとした場合，仕様上 Gas が莫大な量になってしまう．ユーザーの資金面から考えたときに，電子署名を施すだけで大きな資産を失うことは意図に反する．

そこで本稿においては，Ethereum 上に実装されている EIP[*11]-198 [19] を用いることで，RSA 署名検証時の Gas を大幅に削減することを試みる．EIP-198 とは，大きな整数の冪乗剰余を効率よく計算し，その際の Gas の発生を小さくする実装である．ゆえに，スマートコントラクト上で冪乗剰余を計算するプログラムを記入するのではなく，EIP-198 を呼び出すプログラムを記入する．

5.2.2 公開鍵管理スマートコントラクト

行政機関が所有して公開鍵を管理するスマートコントラクトである．機能としては，ハッシュテーブルに公開鍵のハッシュ値を追加する関数 **Add-Key** と削除する関数 **Delete-Key** に加え，公開鍵のハッシュ値がハッシュテーブルに存在しているかどうかを true/false で返す関数 **Find-Key** の 3 つの関数で構成される．なお，**Add-Key** と **Delete-Key** は行政機関の EOA のみが実行できるようにアクセス制御をかける．ハッシュ関数として，Ethereum で使われる keccak256 を使用する．

以下にそのアルゴリズムを示す．

Algorithm 1 RSA_verify

Input: pub_key，電子署名，電子署名のハッシュ値
Output: void

```
if Find_key(hash(pub_key)) == true then
    if RSA 署名検証 == true then
        pub_key，電子署名，電子署名のハッシュ値の登録
    else
        エラー処理
    end if
else
    エラー処理
end if
```

Algorithm 2 Add-Key

Input: pub_key のハッシュ値
Output: void

```
if 実行者のアドレス == 行政機関の EOA then
    hashtable(pub_key のハッシュ値) = true
else
    エラー処理
end if
```

Algorithm 3 Delete-Key

Input: pub_key のハッシュ値
Output: void

```
if 実行者のアドレス == 行政機関の EOA then
    hashtable(pub_key のハッシュ値) = false
else
    エラー処理
end if
```

Algorithm 4 Find-Key

Input: pub_key のハッシュ値
Output: true/false

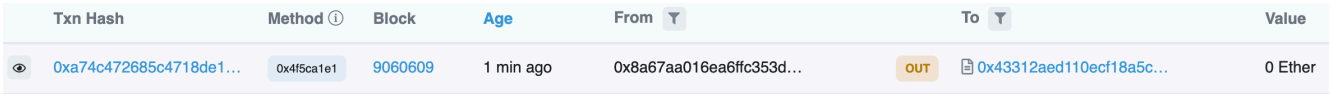
```
return hashtable(pub_key のハッシュ値)
```

5.3 実現可能性の評価

プロトタイプの実装が実際に動くことを示し，使用料としての Gas が実現可能の範囲であることを示す．

まず，行政機関の立場で，ユーザーの公開鍵を公開鍵管理スマートコントラクト上のリストに関数 **Add-Key** を用いて追加する．その際のトランザクションを図 3 に示す．

[*11] Ethereum Improvement Proposal の略．Ethereum の改善案として github 上で提案されるものであり，議論の末に実装されるものがある



Txn Hash	Method ⓘ	Block	Age	From		To	Value
0xa74c472685c4718de1…	0x4f5ca1e1	9060609	1 min ago	0x8a67aa016ea6ffc353d…	OUT	0x43312aed110ecf18a5c…	0 Ether

図 3 行政機関からのトランザクション（Add-key）

表 2 Gas コスト

トランザクション	Gas Used	手数料	日本円
Rsa-verify	39,871 Gas	1675778.13 Gwei	346 円
Add-Key	46,197 Gas	1941659.91 Gwei	401 円
Delete-Key	24,363 Gas	1023976.89 Gwei	211 円

表 3 アドレスの振り分け

役割	アドレス
ユーザーの EOA	0xAdae05c081Ad663C6f0106e6B32b0728771dFcB3
行政機関の EOA	0x8a67AA016EA6fFC353D6ea236A3141EB73cebD2B
署名検証スマートコントラクト	0xE23C106D1E9187301c72135d5C4bE7D7C4e138fA
公開鍵管理スマートコントラクト	0x43312aEd110eCf18A5C3168531067F70e4141F8d

次に，ユーザーの立場で，アプリケーションを利用してマイナンバーカードから電子署名の作成，関数 **RSA-verify** を利用して，スマートコントラクト内で署名検証を行う．その際のトランザクションを図 4 に示す．この結果は，マイナンバーカード内の秘密鍵は所有者であっても取り出すことができないため，作成したアプリケーションがマイナンバーカードから署名を作成し，その検証をスマートコントラクト内で実現できたこと示す．

また，実行した関数ごとに発生する Gas コストを表 2 に示し，スマートコントラクトのコントラクトアドレスや行政機関，ユーザーの EOA を表 3 に示す．なお，コード量は署名検証スマートコントラクトは 109 行，公開鍵管理スマートコントラクトは 24 行である．これは MacOS Big Sur 11.4 の **cloc** コマンドで計測した．また，Gas Price は 2021 年 7 月 17 日現在のメインネット平均値である 42.03 Gwei として計算した．本稿では，パブリック型のブロックチェーンを利用したため手数料が発生するが，コンソーシアム型でシステムを構成することで，手数料を発生させないことが可能になる．なお参考情報としての，日本円への変換は 2021 年 7 月 17 日 0 時において，1ETH あたりおよそ 206,614 円であるため，その値を用いた．

6. 考察

6.1 マイナンバーカードと EOA の紐付け

提案手法では，マイナンバーカードと EOA の紐付けを行う必要がない．ゆえにユーザーが普段利用している EOA の秘密鍵が盗まれたとしても，マイナンバーカードと交付時に入力した署名用電子証明書の PIN を入手しない限り，電子署名を作成することができない．つまり，ブロックチェーンに格納されるまでの確定日付の対改竄性はマイナンバーカードによる RSA 署名にのみ依存する．仮に適当な電子署名を用いてトランザクションを発行したとしても，署名検証スマートコントラクトによる検証でそのトランザクションを拒否することができる．よって，EOA の秘密鍵が盗難にあった際のリスクが提案手法においては少なく，本システムにおいては厳重に EOA の秘密鍵を管理する必要がないという利点があげられる．

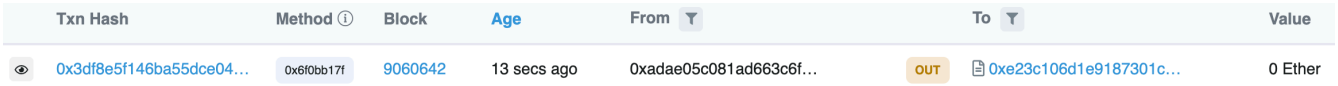
6.2 トランザクションレシート

トランザクションレシートが確定日付の証明になるということについて考察する．従来，書類が改竄されていないことと書類の発行元が確かに発行したことを示すためには電子署名が使われていた．一方，本研究では，電子署名の代わりにトランザクションレシートを用いる．そのため，このレシートに改竄がなく，正しく発行されていることを示す必要がある．以下の 2 点からレシートの改竄が難しいことを示す．

1 点目は，ブロックチェーン格納前の耐改竄性についてである．トランザクションから生成したレシートを改竄するためには，OSS で管理される Geth や Parity などの Ethereum クライアントを不正に改変する必要がある．しかし，ブロックチェーン上の全ての参加者のクライアントを改変することは現実的に不可能である．その結果，改竄されたレシートを含むブロックを他のノードに伝播させたとしても，他のノードはそのブロックを受け取ることはないため，改竄されたレシートを拡散できない．さらに，レシートを発行するスマートコントラクトを改竄できないことも重要である．2 点目は，ブロックチェーン格納後の耐改竄性についてである．ブロックチェーンに格納されているレシートが暗号学的ハッシュ関数によって守られているため，この改竄は実質的に不可能である．したがって，上記 2 点からトランザクションレシートの改竄は不可能であると言える．

6.3 電子文書の保存先

現在は，希望することで公証役場のサーバーに電子書類を 20 年保存することができる．提案手法では，電子文書のハッシュのみがブロックチェーンに保存されるが，文書自体の保存については対象外としている．提案手法において，電子文書をどこに保存するかを考えたときに，IPFS などの分散型ファイルシステムが有力な候補に上がる．P2P で繋がったファイルサーバーにメディアを保存し，耐改竄性や耐障害性を持つため，ブロックチェーンと非常に相性が良いと言われている．このシステムに暗号化を施した電子書類を保存し，その url を署名検証の際にスマートコン



Txn Hash	Method ⓘ	Block	Age	From		To	Value
0x3df8e5f146ba55dce04…	0x6f0bb17f	9060642	13 secs ago	0xadae05c081ad663c6f…	OUT	0xe23c106d1e9187301c…	0 Ether

図 4 ユーザーからのトランザクション（Rsa_Verify）

トラクトに送信するようにすることで，電子書類そのものもトランザクションレシートに紐づけることが可能になる．

6.4 スケーラビリティ

提案手法の実現性を考えた時，公開鍵の登録がマイナンバーカードの数だけ行われなければならない．その際，探索時間と探索した分だけ増加する Gas 代が懸念材料となるが，スマートコントラクト内で **Mapping** を使用することで一定の Gas と時間で探索が可能になる．なぜなら，スマートコントラクトを記述する Solidity において **Mapping** はハッシュテーブル [20] であり，データの探索や追加は実質 $\mathcal{O}(1)$ であることが知られている．

7. まとめ

本稿では，マイナンバーカードによる電子署名機能とスマートコントラクトによる契約の自動履行を組み合わせ，トランザクションレシートを確定日付の証明書とすることで，公証人などの第三者を介さない自動確定日付システムの提案，実装，評価を行なった．我々は swift を用いてスマートフォンでマイナンバーカードから電子署名を作成できるアプリを実装し，Solidity を用いて，電子署名検証を行うスマートコントラクトを実装した．また，電子署名検証の際に，EIP-198 の機能を用いることで Gas コストを小さく抑えることに成功し，提案手法が実現可能であることを示した．加えて，その際に発生するトランザクションを記録し，Gas コストを計測した．

今後の展望としては，電子文書を IPFS などの分散型ファイルシステムに保存できるようにアプリケーションを改善することを検討している．

謝辞 本研究は公益財団法人 GMO インターネット財団の助成を受けたものです．

参考文献

[1] Notarize. Legally notarize your documents online. anytime. anywhere. https://www.notarize.com/, 2021.
[2] Centre of Registers and Information Systems. e-notary — rik. https://www.rik.ee/en/other-services/e-notary, 2021.
[3] 時事通信社. 公正証書のデジタル化検討　利便性向上へ 上川法相指示. https://www.jiji.com/jc/article?k=2021052800450, 2021.
[4] S. Nakamoto. Bitcoin:a peer-to-peer electronic cash system. http://www.bitcoin.org/bitcoin.pdf, 2009.
[5] Ethereum Wiki. A next-generation smart contract and decentralized application platform. https://github.com/ethereum/wiki/wiki/White-Paper, 2021.
[6] G. Wood. Ethereum: A secure decentralised generalised transaction ledger. https://ethereum.github.io/yellowpaper/paper.pdf, 2021.
[7] 総務省. 公的個人認証サービスによる電子証明書（民間事業者向け）. https://www.soumu.go.jp/kojinbango_card/kojinninshou-02.html, 2021.
[8] 総務省. 公的個人認証サービスの民間利用. https://www.soumu.go.jp/main_content/000747198.pdf, 2021.
[9] 西村幸浩, 小野津崇之, 志賀正裕. マイナンバーカードの技術仕様と利活用方式 (特集マイナンバー)–(マイナンバー制度の利用拡大). *Fujitsu*, Vol. 68, No. 4, pp. 59–65, 2017.
[10] 総務省. 総務省｜マイナンバー制度とマイナンバーカード｜マイナンバーカード. https://www.soumu.go.jp/kojinbango_card/03.html, 2021.
[11] 日本公証人連合会. 7-5 電子公証 — 日本公証人連合会. http://www.koshonin.gr.jp/business/b07_5, 2021.
[12] 法務省. 法務省：１． ２　電磁的記録の認証（定款を含む私署証書の認証）の嘱託. http://www.moj.go.jp/MINJI/DENSHIKOSHO/denshikosho1-2.html, 2021.
[13] 日本公証人連合会. 8 確定日付 — 日本公証人連合会会. https://www.koshonin.gr.jp/business/b08, 2021.
[14] Zhaoxiong Meng, Tetsuya Morizumi, Sumiko Miyata, and Hirotsugu Kinoshita. Design scheme of copyright management system based on digital watermarking and blockchain. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, Vol. 2, pp. 359–364. IEEE, 2018.
[15] Mohammad Jabed Morshed Chowdhury, Alan Colman, Muhammad Ashad Kabir, Jun Han, and Paul Sarda. Blockchain as a notarization service for data sharing with personal data store. In *2018 17th ieee international conference on trust, security and privacy in computing and communications/12th ieee international conference on big data science and engineering (TrustCom/BigDataSE)*, pp. 1330–1335. IEEE, 2018.
[16] Olawande Daramola and Darren Thebus. Architecture-centric evaluation of blockchain-based smart contract e-voting for national elections. In *Informatics*, Vol. 7, p. 16. Multidisciplinary Digital Publishing Institute, 2020.
[17] 永田和之, 李中淳, 福田賢一, 岩丸良明, 庭野栄一, 谷内田益義, 平良奈緒子, 鈴木裕之, 小尾高史, 大山永昭ほか. ブロックチェーンにおける本人性確認の方法に関する考察. 研究報告コンピュータセキュリティ (CSEC), Vol. 2017, No. 19, pp. 1–6, 2017.
[18] 地方公共団体情報システム機構. 公的個人認証サービス プロファイル仕様書　2.0 版. https://www.j-lis.go.jp/data/open/cnt/3/2187/1/13_profile_genkou.pdf, 2019.
[19] Ethereum. Eip-198: Big integer modular exponentiation. https://eips.ethereum.org/EIPS/eip-198, 2017.
[20] Ward Douglas Maurer and Ted G Lewis. Hash table methods. *ACM Computing Surveys (CSUR)*, Vol. 7, No. 1, pp. 5–19, 1975.