Authentication of administrative documents using blockchain technology

Zilèdem Pierre Canisius HIEN
Department of Informatics (DI)
University Joseph KI-ZERBO (UJKZ)
OUAGADOUGOU, BURKINA FASO
hiencanisius@gmail.com

Ousmane BARRA
Department of Informatics (DI)
University Joseph KI-ZERBO (UJKZ)
OUAGADOUGOU, BURKINA FASO
ousmane.barra@ujkz.bf

Yaya TRAORE
Department of Informatics (DI)
University Joseph KI-ZERBO (UJKZ)
OUAGADOUGOU, BURKINA FASO
yaya.traore@tic.gov.bf

Sadouanouan MALO
Department of Informatics (DI)
University NAZI BONI (UNB)
BOBO-DIOULASSO, BURKINA FASO
sadouanouan.malo@u-naziboni.bf

***Abstract*—Administrative documents have always played a central role not only in the functioning of the Administration but also in the relationship between the Administration and its users/customers. Indeed, they are used for traceability and serve as a basis for decision-making, communication and legal purposes. This means that all administrative documents must be reliable and of indisputable origin. However, it is increasingly common to encounter falsified digital administrative documents or those of dubious origin. This phenomenon creates a crisis of confidence between users/customers and the administration. Blockchain is a revolutionary technology can help secure administrative documents. In this paper, we propose a decentralized approach to authenticating administrative documents based on the non-repudiation, transparency and decentralized storage properties of blockchain technology. We combine the use of a smart contract, the SHA-256 hashing algorithm, digital signatures and electronic time stamping. Our process begins with the registration of administrative documents on the blockchain by an administrative entity. Thus, any user (or other administrative entity) wishing to verify the authenticity of an administrative document in their possession can also upload it to the Ethereum blockchain, which will (after research and processing) confirm or deny the authenticity of said document. This solution not only guarantees greater security and integrity for administrative documents, but also offers the ability to verify their authenticity in real time in a simple manner. Simulations show that it is potentially useful for administration.**

Keywords—authentication, security, document, blockchain, cryptography, decentralized application

I. INTRODUCTION

Document authentication is a process whereby a computer system or a human being proves or certifies that a document is authentic. A document is considered authentic if it is the original, or a copy that conforms to the original after verification and validation by an authorised or competent person. However, this form of certification is not intended to prove the veracity of the document's content. Its purpose is to ensure that the signature and stamp visible on the document originate from an official authority and that the date of signature or issue, the name and position of the signatory are also legible.

The reliability of certain administrative documents is increasingly controversial, as it is now common to find falsified documents or documents of dubious origin in the public digital space, causing confusion among many citizens. This could be due in part to the development and misuse of multiple tools based, for example, on Artificial Intelligence (AI).

Indeed, the phenomenon of falsified documents contributes to tarnishing the image of their legitimate author, which could be the government, thus creating a crisis of confidence between the government and its users/customers. To address this phenomenon, authors and recipients of documents sometimes carry out authentication operations, at enormous cost in terms of time and money, which could ultimately reduce their returns.

Blockchain is a technological revolution that has been rapidly developing in recent years. Known for its non-repudiation, transparency and decentralized storage properties, it offers numerous advantages in many aspects of human life and is proving to be an effective means of improving government public services in particular. As a result, the use of this technology in the authentication process for administrative documents could bring greater ease, reliability and security.

In this paper we propose an approach to document authentication using blockchain technology. Indeed, are therefore interested in the use of blockchain technology, particularly its consensus, verification and validation protocols, to resolve the issue of authenticating administrative documents, in order to reduce the risk of fraud and falsification of these documents.

The rest of the paper is organized as follows: in Section 2 we present a state-of-the-art review of existing work. In Section 3 we present our methodology. In section 4 we present a simulation of the approach. We end with a conclusion and prospects in Section 5.

II. STATE OF THE ART

Blockchain technology, propelled by the advent in 2008 of *Satoshi Nakamoto*'s bitcoin cryptocurrency [1], has been growing rapidly in recent decades. It is a digital technology that allows information to be stored chronologically and transmitted in the form of blocks linked together securely (heavy use of cryptography) and without a central authority.

According to *Imran Bashir* [2], third-generation blockchains are applicable beyond the financial services industry in various general-purpose societal areas such as media, healthcare, government, etc. In addition to Bitcoin, there are several blockchains (Solana, Ethereum, Cardano, Avalanche, etc.), among which we have chosen Ethereum for this experiment.

A. Ethereum Blockchain

The public Ethereum blockchain, whose consensus mechanism switched from Proof of Work (PoW) to Proof of Stake (PoS) in September 2022 [3], underwent several changes after it was announced at the North American Bitcoin Conference in Miami in January 2014 and then launched on 30 July 2015 [4]. In the white paper published in 2013 [5], *Vitalik Buterin* stated that "Ethereum's goal is to create another protocol for developing decentralized applications, offering a different set of trade-offs that we believe will be very useful for a wide range of decentralized applications. It will primarily focus on situations where rapid development, security for small, rarely used applications, and the ability for different applications to interact with each other in a highly efficient manner are important.". In line with this dynamic, Ethereum 2.0 regularly introduces numerous developments, such as its May 7, 2025 update via the Pectra hard fork, which makes the network more secure, accessible, and user-friendly.

The ability to produce decentralized applications based on Ethereum – operating on a peer-to-peer network – relies heavily on the use of smart contracts. This makes Ethereum a Turing Complete blockchain [6], unlike the Bitcoin network. In certain use cases of Ethereum, such as ours, we must use two (02) accounts. The first is an **Externally Owned Account (EOA)** controlled by a cryptographic private key, with several hundred ETH and identifiable by an address (a 42-characters hexadecimal hash prefixed with 0x). The second is a **Contract Account (CA)** whose address is obtained after deployment (using ≈782016 Gas) of our smart contract.

Technically, a smart contract refers to a computer program implemented in a high-level programming language such as Solidity, Vyper, etc., whose code is compiled into bytecode and deployed on a specific version of the blockchain virtual machine – the EVM, in the case of Ethereum. It is designed to run autonomously when certain criteria are met. In addition to their advantages in terms of speed, accuracy, transparency and security, smart contracts are, according to [2], a revolutionary feature of blockchain because they allow for flexibility, empowerment and highly desirable control over the actions desired by users.

B. Document authentication and security

According to Article 4 of [7], the following are considered administrative documents: "documents produced or received, in the context of public service, by the State, local authorities and other public or private entities entrusted with such a mission;". These include, for example, memos, decisions, instructions, circulars, directives, journals, deliberations, reports, minutes, sketches, plans, diagrams, notices, forecasts, official communiqués, certificates (of commencement, resumption, or termination of service, etc.), bulletins, decrees, orders, etc.

Given that a document refers to information stored on paper or electronic media, document authentication is a process whereby a computer system or human being proves or certifies that a document is authentic. In general, for authorized individuals – civil registrars, police officers, official authorities that issued the document, etc. – the process consists of ensuring that the signature and stamp visible on the document (original, certified copy of the original) originate from an official authority and that the date of signature or issue, the name and position of the signatory are also legible.

In all cases, authentication is a component of security, which itself is a set of measures to be taken and implemented to ensure traceability related to access and the protection of sensitive information (electronic or physical).

C. Existing methods of document authentication

There are several methods of authenticating digital documents.

- **Digital signature technology:** This secure method makes it possible to prove the identity of the author of an electronic document and guarantee the integrity of said document using public key cryptography. It thus guarantees non-repudiation, i.e. it is virtually impossible to challenge or deny the document. A digital signature must be authentic, unforgeable, non-reusable, unalterable and irrevocable [8].

- **Hashing or digital fingerprinting technique:** This technique involves applying a one-way algorithm (SHA-2, SHA-3, etc.) to the document to generate a unique digital fingerprint. This technique does not prove the identity of the author, but only the integrity of the document.

- **Electronic timestamping technique:** Most often based on digital signatures and hashing, this technique is used to prove that a document existed on a given date.

- **RFID and QR code techniques:** Radio frequency identification (RFID) and quick response codes (QR codes) are used in many sectors (commerce, administration, healthcare, etc.) to track certain goods or verify the authenticity of documents, particularly physical ones.

- **Blockchain technology:** *Isyak Meirobie et al.* [9] have set up a framework for authenticating electronic documents using Blockchain technology for the Indonesian government system. Go-Chain stores the hashed (in SHA256) signature (P-256 curve DSA) of the digital transcript of the document on a public blockchain. This transcript is also distributed to the public in json format. The author opted for the Proof of Authority (PoA) consensus protocol. *Ana BAKHOUM* [10] has also proposed a decentralized system for backing up and securing electronic school reports in the Ethereum blockchain via a smart contract. This aims to ensure the reliability, authenticity, transparency and security of electronic reports. Our solution innovates with the integration of electronic timestamping and the non-direct manipulation of cryptographic keys by the general public. It offers the possibility of verifying the authenticity of documents in real time, securely, transparently, and without going through a central verification authority.

III. Methodology

Our authentication approach (Fig. 1) consists of two main phases that are ordered and complementary.

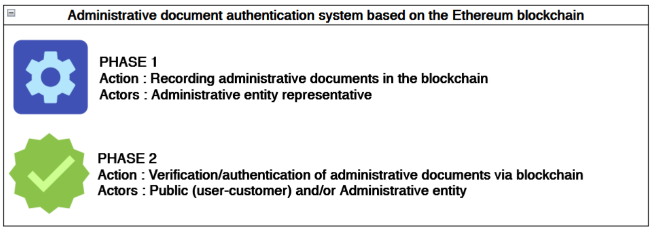

Administrative document authentication system based on the Ethereum blockchain
PHASE 1
Action : Recording administrative documents in the blockchain
Actors : Administrative entity representative
PHASE 2
Action : Verification/authentication of administrative documents via blockchain
Actors : Public (user-customer) and/or Administrative entity

Fig. 1. Constitutive phases of the approach to authenticating administrative documents using the Ethereum blockchain.

Each of these phases consists of several stages.

A. *PHASE 1 - Recording administrative documents in the Ethereum blockchain*

The execution of Phase 1 (Fig. 2) requires the payment of gas fees.

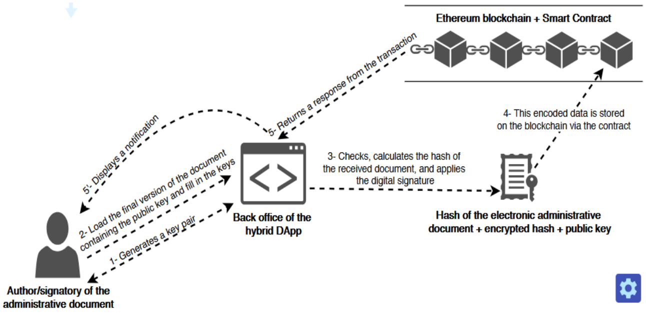

Ethereum blockchain + Smart Contract
5- Returns a response from the transaction
5'- Displays a notification
4- This encoded data is stored on the blockchain via the contract
2- Load the final version of the document containing the public key and fill in the keys
1- Generates a key pair
Back office of the hybrid DApp
3- Checks, calculates the hash of the received document, and applies the digital signature
Hash of the electronic administrative document + encrypted hash + public key
Author/signatory of the administrative document

Fig. 2. PHASE 1 - Recording administrative documents in the Ethereum blockchain.

The process for this phase 1 is described as follows:

- **1-Generates a key pair :** the author generates a pair of asymmetric keys through the Decentralized Application (DApp) back office. These keys must not be lost, and the private key must be carefully stored.
- **2-Load the final version of the document containing the public key and fill in the keys :** the author integrates their public key into the document so that it is visible. However, this remains optional. They then upload (via the DApp back office) the final PDF or Word version of the document containing their public key. During the upload, they enter the key pair. The front end transmits this data to the DApp back end.
- **3-Checks, calculates the hash of the received document, and applies the digital signature :** the backend extracts the (text) content of the uploaded file and calculates a hash (or digital fingerprint) of this content. The backend then encrypts the calculated hash with the author's private key. This produces a signed hash, which is the document's digital signature. This data (the raw hash, the signed hash, and the author's public key) is encoded and sent back to the DApp frontend.
- **4-This encoded data is stored on the blockchain via the contract :** the frontend then transfers the encoded data received from the backend (raw hash, signed hash, and public key) to the smart contract – by calling the *storeAdministrativeDocument()* method–, which stores it on the Ethereum 2.0 blockchain. During storage, the timestamp is calculated and associated with the data by the smart contract. Thus, when a user wants to verify authenticity, the corresponding public key is used to decrypt and verify the document's signature.
- **5-Returns a response from the transaction :** a notification from the blockchain transaction is returned to the frontend. (5'-Displays a notification) This notification is customized and displayed to the user.

B. *PHASE 2 - Verification or authentication of administrative documents on the Ethereum blockchain*

This phase is illustrated in Fig. 3. No fees are payable here as it is simply a matter of reading data in Ethereum. This seems to justify *Jean-Paul Delahaye*'s [11] expression "(...) freely and without charge, everyone can read (...)" in his definition of blockchain.

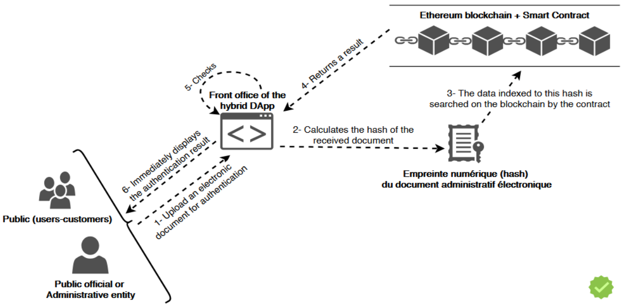

Ethereum blockchain + Smart Contract
5- Checks
Front office of the hybrid DApp
4- Returns a result
3- The data indexed to this hash is searched on the blockchain by the contract
2- Calculates the hash of the received document
6- Immediately displays the authentication result
1- Upload an electronic document for authentication
Empreinte numérique (hash) du document administratif électronique
Public (users-customers)
Public official or Administrative entity

Fig. 3. PHASE 1 - Verification or authentication of administrative documents on the Ethereum blockchain.

Phase 2 also consists of several steps that can be described as follows:

- **1-Upload an electronic document for authentication :** the user (public or public official) uploads the PDF or Word document through the DApp front office whose authenticity they wish to verify. The frontend transmits this data to the DApp backend.
- **2-Calculates the hash of the received document :** the backend extracts the (text) content of the uploaded file for authentication and calculates the hash of this content. This digital fingerprint (hash) is encoded and sent back to the frontend of the DApp.
- **3-The data indexed to this hash is searched on the blockchain by the contract :** the frontend uses this hash and queries the Ethereum 2.0 blockchain via the smart contract (deployed there) to retrieve the raw digital fingerprint, digital signature, and public key previously stored in the blockchain (through phase 1). This search is possible by using the *getAdministrativeDocument()* method of the contract.
- **4-Returns a result :** the blockchain returns a response to the DApp via the smart contract. This response is then transmitted to the DApp's backend for verification.
- **5-Checks :** the backend decodes the data received from Ethereum and performs checks: if an occurrence

is found (potentially intact document), the DApp verifies the digital signature associated with this occurrence via the previously stored public key. If this verification fails, then the signature does not come from the correct signatory, i.e., the author who validated the final version of the document. This is the verification of the document's authenticity. If the signature is valid, then the document is declared authenticated and authentic.

- **6-Immediately displays the authentication result :** a notification is sent to the frontend. This means that the authentication result is immediately personalized and displayed to the user.

C. *Implementation architecture*

Integrating smart contracts into a traditional application allows interaction with the blockchain. For successful integration, we have adopted a clear DApp architecture (Fig. 4) where the responsibilities of each component are well defined.

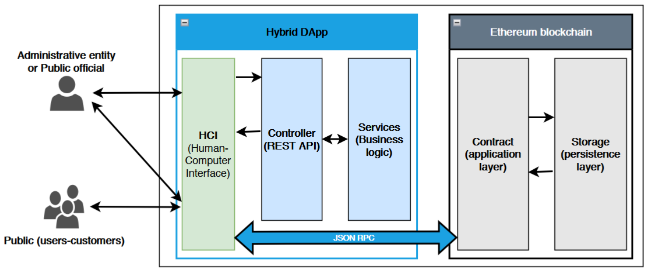


Fig. 4. Simplified layered architecture of the solution.

The Human Computer Interfaces (HCI or frontend) layer consists of two areas: a back office that is only accessible by the administrative entity, and a front office intended for the general public. The Controller layer, which is the upper web layer of the backend, is responsible for exposing REST APIs for the benefit of the HCI layer. These APIs provide resources (hash, signature, etc.) that are calculated by the Services layer (responsible for business logic). The responsibility of the application layer of the Ethereum blockchain is to host our smart contract and to work with the other layers of this blockchain to execute the contract. This application layer communicates with the HCI layer. It also communicates with the Ethereum persistence layer, where the hashed, encrypted, and encoded data from administrative documents is stored and searched. This is what enables decentralized storage of these documents through immutable transaction requests. The various interactions (transaction requests) between the EOA account and our smart contract are shown in Fig. 5.

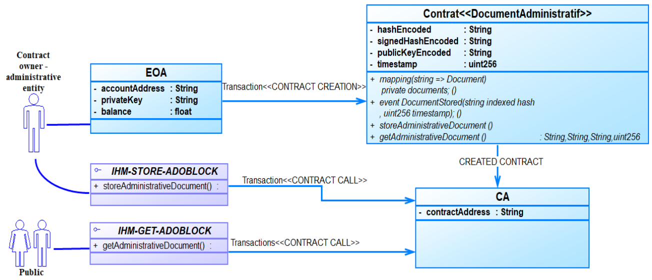

Contract owner - administrative entity
EOA
- accountAddress : String
- privateKey : String
- balance : float
Transaction<<CONTRACT CREATION>>
Contrat<<DocumentAdministratif>>
- hashEncoded : String
- signedHashEncoded : String
- publicKeyEncoded : String
- timestamp : uint256
+ mapping(string => Document) private documents; ()
+ event DocumentStored(string indexed hash , uint256 timestamp); ()
+ storeAdministrativeDocument ()
+ getAdministrativeDocument () : String,String,String,uint256
CREATED CONTRACT
IHM-STORE-ADOBLOCK
+ storeAdministrativeDocument() :
Transaction<<CONTRACT CALL>>
CA
- contractAddress : String
IHM-GET-ADOBLOCK
+ getAdministrativeDocument() :
Transactions<<CONTRACT CALL>>
Public

Fig. 5. Diagram of interactions between the smart contract and accounts.

IV. Testing the Solution

A. *Tools*

The implementation requires at least one computer with a Core i7 processor, 16 GB of RAM, and 128 GB of SSD storage, with no operating system requirements.

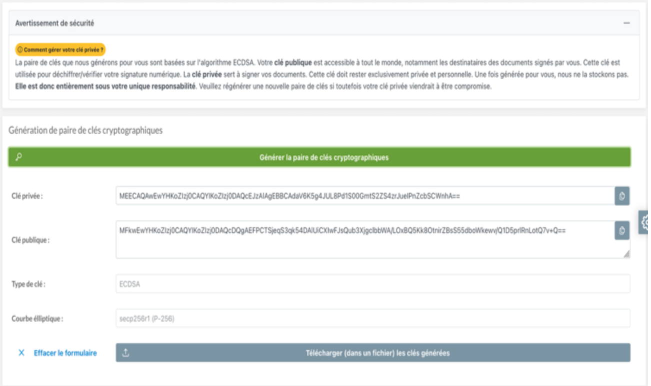
The backend and frontend parts of the DApp were implemented with Visual Studio Code 1.96.2 using the Spring-boot 3.4.1 (Maven 3.8.1 + Java 17) and Angular 15 (HTML5, TypeScript/JavaScript, SCSS/CSS3) frameworks.

The smart contract was implemented and tested in **Solidity** ^0.8.28 on Remix-Ethereum IDE. The **MetaMask** v2.132.0 cryptocurrency wallet was integrated for blockchain transaction requests and signatures, which are made via the JSON Remote Procedure Call (**JSON RPC**) protocol to Angular via the Application Binary Interface (**ABI**). In terms of core technologies, we experimented with the **Hardhat** ^2.24.0 framework and the **Ethers.js** ^6.14.0 library to configure the test blockchain network (Localhost network), compile, and deploy our contract. In order to generate ECDSA (Elliptic Curve Digital Signature Algorithm) **cryptographic key pairs**, we use the Maven **bouncycastle** dependency; this allows us to use the **secp256r1** (P-256) elliptic curve for digital signatures of digital fingerprints calculated using the **SHA-256** algorithm. P-256 is compatible with Ethereum and the standards of the National Institute of Standards and Technology (NIST) [12] [13].

B. *Simulations*

According to our experiments on official announcements, storing a file of approximately 500 KB requires an average of 322,200 Gas, or 389,212.119378 Gwei (0.000389212119378 ETH, given that 1 ETH = 10^9 Gwei = 10^{18} Wei). These operations are instantaneous on the local test network. ETH refers to “Ether”, the native cryptocurrency of the Ethereum blockchain.

The solution we propose, under the name ADOBLOCK, is accessible through a web browser. Figures 6, 7, and 8 show the main user interfaces.



Avertissement de sécurité

Comment gérer votre clé privée ?

La paire de clés que nous générons pour vous sont basées sur l'algorithme ECDSA. Votre **clé publique** est accessible à tout le monde, notamment les destinataires des documents signés par vous. Cette clé est utilisée pour déchiffrer/vérifier votre signature numérique. La **clé privée** sert à signer vos documents. Cette clé doit rester exclusivement privée et personnelle. Une fois générée pour vous, nous ne la stockons pas. **Elle est donc entièrement sous votre unique responsabilité**. Veuillez régénérer une nouvelle paire de clés si toutefois votre clé privée viendrait à être compromise.

Génération de paire de clés cryptographiques

Générer la paire de clés cryptographiques

Clé privée : MEECAQAwEwYHKoZIzj0CAQYIKoZIzj0DAQcEJzAlAgEBBCAdaV6K5g4JUL8Pd1S00GmtS2ZS4zrJuelPnZcbSCWnhA==

Clé publique : MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEFPCTSjeqS3qk54DAlUiCXlwFJsQub3XjgclbbWA/LOxBQ5Kk8OtnirZBsS55dboWkewvjQ1D5prlRnLotQ7v+Q==

Type de clé : ECDSA

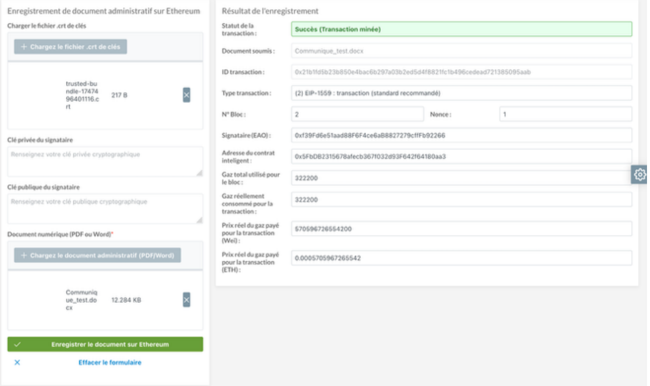
Courbe élliptique : secp256r1 (P-256)

× Effacer le formulaire

Télécharger (dans un fichier) les clés générées

Fig. 6. Get private/public keys.

This interface in Fig. 6 allows you to obtain key pairs and save them in a compressed file. These keys are reusable as long as the private key does not appear to be compromised.



Enregistrement de document administratif sur Ethereum

Charger le fichier .crt de clés

trusted-bundle-1747496401116.crt 217 B

Clé privée du signataire

Clé publique du signataire

Document numérique (PDF ou Word)*

Communique_test.docx 12.284 KB

Résultat de l'enregistrement

Statut de la transaction : Succès (Transaction minée)

Document soumis : Communique_test.docx

ID transaction : 0x21b1fd5b23b850e4bac6b297a03b2ed5d4f8821fc1b496cedead721385095aab

Type transaction : (2) EIP-1559 : transaction (standard recommandé)

N° Bloc : 2 Nonce : 1

Signataire (EAO) : 0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266

Adresse du contrat inteligent : 0x5FbDB2315678afecb367f032d93F642f64180aa3

Gaz total utilisé pour le bloc : 322200

Gaz réellement consommé pour la transaction : 322200

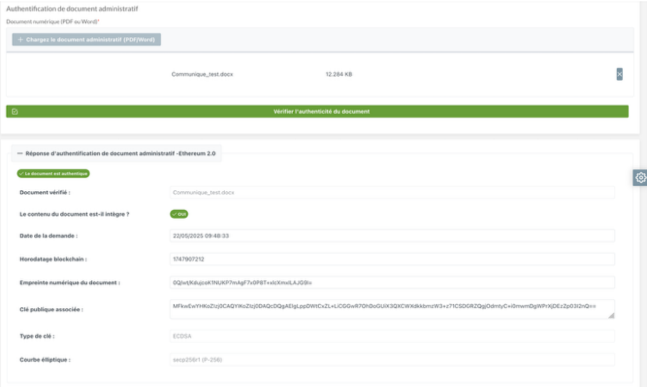
Prix réel du gaz payé pour la transaction (Wei) : 570596726554200

Prix réel du gaz payé pour la transaction (ETH) : 0.0005705967265542

+ Chargez le fichier .crt de clés
Renseignez votre clé privée cryptographique
Renseignez votre clé publique cryptographique
+ Chargez le document administratif (PDF/Word)
✓ Enregistrer le document sur Ethereum
× Effacer le formulaire

Fig. 7. Store administrative documents.

Fig. 7 shows the recording of an administrative document in the blockchain. In addition to the transaction status and hash, nonce, etc., the solution displays the amounts of gas used and fees paid.



Authentification de document administratif

Document numérique (PDF ou Word)*

Communique_test.docx 12.284 KB

Réponse d'authentification de document administratif -Ethereum 2.0

Document vérifié : Communique_test.docx

Le contenu du document est-il intègre ? ✓ OUI

Date de la demande : 22/05/2025 09:48:33

Horodatage blockchain : 1747907212

Empreinte numérique du document : 0Q/wt/KdujcoK1NUKP7mAgF7x0P8T+xlcXmxlLAJG9I=

Clé publique associée : MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEIgLppDWtCxZL+LiCGGwR7OhDoGUiX3QXCWXdkkbmzW3+z71CSDGRZQgjOdmtyC+i0mwmDgWPrXjDEz2p03I2nQ==

Type de clé : ECDSA

Courbe élliptique : secp256r1 (P-256)

+ Chargez le document administratif (PDF/Word)
Vérifier l'authenticité du document
✓ Le document est authentique

Fig. 8. Verify or authenticate document.

This interface displays the result of an authentication operation. It includes the timestamp, the integrity status of the document submitted for authentication, and the public key of the document's author found on the blockchain. The decoded digital fingerprint, key types, and signature curve are also displayed.

Conclusion

Faced with the rise of document falsification, it is crucial to leverage technological advances and innovations to rethink methods of authenticating and securing digital documents. To this end, we have proposed an approach to document authentication using blockchain technology. To test this approach, we have implemented a hybrid decentralized application that interacts with the Ethereum blockchain using a smart contract. In this solution, we have integrated the use of electronic timestamping, digital signatures based on asymmetric keys, and hashing techniques using the SHA-256 algorithm.

The results of our experiments confirm that blockchain technology shows promise for guaranteeing the authenticity and security of administrative documents. However, further consideration should be given to the sovereignty of digital data and the integration of other pipelines to support electronic documents with non-textual content.

References

[1] R. Subramanian et T. Chino, « The State of Cryptocurrencies, Their Issues and Policy Interactions », J. Int. Technol. Inf. Manag., vol. 24, no 3, p. 40, janv. 2015, doi: 10.58729/1941-6679.1045.

[2] I. Bashir, Mastering Blockchain. Packt Publishing Ltd, 2017.

[3] « What is consensus ? A beginner's guide », What is consensus ? A beginner's guide, May 13, 2022. Accessed: September 15, 2024. [Online]. Available at : https://crypto.com/fr/university/consensus-mechanisms-explained

[4] « Ethereum », Wikipédia. January 22, 2025. Accessed: February 10, 2025. [Online]. Available at: https://fr.wikipedia.org/wiki/Ethereum

[5] T. Bourbotte, « What is Ethereum? Our explanations to learn everything about this blockchain and its cryptocurrency ETH », Cryptoast, December 20, 2024. Accessed: February 10, 2025. [Online]. Available at: https://cryptoast.fr/fiche-ethereum/

[6] S. S. Kushwaha, S. Joshi, D. Singh, M. Kaur, et H.-N. Lee, « Systematic Review of Security Vulnerabilities in Ethereum Blockchain Smart Contract », IEEE Access, vol. 10, p. 6605-6621, 2022, doi: 10.1109/ACCESS.2021.3140091.

[7] BURKINA FASO, « Loi N°051-2025/CNT du 30 août 2015 portant droit d'accès à l'information publique et aux documents administratifs », p.4. Accessed: September 16, 2024. [Online]. Available at: https://www.csc.bf/index.php/textes-dereference/lois/item/76-loi-051-portant-sur-l-acces-a-l-information-publique

[8] F. Num, « La signature électronique : un outil devenu incontournable - francenum.gouv.fr », Direction générale des entreprises, December 7, 2020. Accessed: February 21, 2025. [Online]. Available at: https://www.francenum.gouv.fr/guides-et-conseils/pilotage-de-lentreprise/dematerialisation-des-documents/la-signature

[9] I. Meirobie, A. P. Irawan, H. T. Sukmana, D. P. Lazirkha, et N. P. L. Santoso, « Framework Authentication e-document using Blockchain Technology on the Government system », Int. J. Artif. Intell. Res., vol. 6, no 2, Art. no 2, Dec. 2022, doi: 10.29099/ijair.v6i2.294

[10] A. Bakhoum, « La Blockchain pour la Sécurisation des E-livrets scolaires », 2019. Accessed: September 10, 2024. [Online]. Available at: http://rivieresdusud.uasz.sn/xmlui/handle/123456789/1803

[11] T. Boudès, « La blockchain déchaîne les questions ! », Ann. Mines - Gérer Compr., vol. 131, no 1, p.83-85, mars 2018, doi: 10.3917/geco1.131.0083.

[12] É. (gapz) Gaspar, « Standardisation des courbes elliptiques : à qui faire confiance ? | Connect - Editions Diamond », April 2016. Accessed: April 18, 2025. [Online]. Available at: https://connect.ed-diamond.com/MISC/mischs-013/standardisation-des-courbes-elliptiques-a-qui-faire-confiance

[13] C. de la sécurité des télécommunications Canada, « Conseils sur la configuration sécurisée des protocoles réseau (ITSP.40.062) », Centre canadien pour la cybersécurité, October 15, 2020. Accessed: April 18, 2025. [Online]. Available at: https://www.cyber.gc.ca/fr/orientation/conseils-sur-la-configuration-securisee-des-protocoles-reseau-itsp40062