frontiers | Frontiers in Blockchain

TYPE Methods
PUBLISHED 10 December 2025
DOI 10.3389/fbloc.2025.1699773

OPEN ACCESS

EDITED BY
Alex Zarifis,
University of Southampton, United Kingdom

REVIEWED BY
Eugenia Politou,
University of Piraeus, Greece
Khamis Zidan,
Aliraqia university, Iraq

*CORRESPONDENCE
R. Ganesan,
✉ rganrao@gmail.com

RECEIVED 05 September 2025
REVISED 15 October 2025
ACCEPTED 23 October 2025
PUBLISHED 10 December 2025

CITATION
Mishra P and Ganesan R (2025) Securing E-governance: a blockchain-based framework for tamper-proof PDF document exchange.
Front. Blockchain 8:1699773.
doi: 10.3389/fbloc.2025.1699773

Securing E-governance: a blockchain-based framework for tamper-proof PDF document exchange

Priyanka Mishra and R. Ganesan*

School of Computer Science and Engineering (SCOPE), VIT University, Chennai, India

E-governance depends on the secure exchange of digital documents, with PDFs widely used for official communication. However, threats such as shadow attacks and policy tampering undermine integrity and trust. This paper proposes a blockchain-based framework that secures PDF workflows by recording complete document hashes and metadata on an immutable ledger. By integrating Policy-based Governance infrastructure, the system strengthens digital signature verification and resists tampering, even under Man-in-the-Middle (MITM) conditions. Evaluation on 1,000 official PDFs shows that the proposed framework achieves 93.1% detection of shadow attacks with 98.4% precision and only 1.6% false positives, compared to traditional PKI methods that failed to detect any tampering. The approach maintains efficiency, scalability, and cost-effectiveness, providing a resilient foundation for secure, transparent, and trustworthy e-governance services.

KEYWORDS

e-governance, blockchain, PDF security, shadow attacks, policy-based governance

1 Introduction

E-Governance has become a pivotal element in modern public administration, facilitating efficient service delivery, streamlined operational workflows, and enhanced transparency via digital platforms. The secure exchange of digital documents, especially PDFs, is fundamental to official communications and interdepartmental collaboration. However, the increasing dependence on digital documentation has escalated vulnerabilities to cyber threats, notably policy tampering attacks, where malicious actors alter digital documents covertly, compromising their integrity and trustworthiness (Alok Mishra et al., 2022).

These documents often contain sensitive information, requiring strict, policy-based access control to prevent unauthorized access or tampering. However, this reliance on access policies introduces a new vulnerability: policy tampering attacks, where malicious actors modify or bypass access policies to gain unauthorized access to classified documents. Policy tampering attacks pose severe risks to the integrity, confidentiality, and authenticity of government operations. They can undermine trust in public institutions, disrupt service delivery, and compromise sensitive information. For instance, the alteration of policy documents in transit could lead to unauthorized modifications that misguide decision-making, impact public programs, or even have legal consequences. Given that PDF files are often the standard for sharing official government documents, ensuring their security against tampering is paramount. The inherent vulnerabilities in some PDF implementations, combined with sophisticated attack techniques, make this a pressing issue for e-governance frameworks.

In an e-Governance system, access to a classified PDF document is restricted to users with the role of Authorized Personnel. If a malicious user is able to tamper with the access policy, they could alter it to grant themselves access, leading to exposure of sensitive information or malicious activities such as document modification.

Consequences of policy tampering:

- Unauthorized Access: Sensitive information might be exposed to individuals without clearance (Safa et al., 2019).
- Document Integrity Risks: Tampered policies could allow attackers to modify document contents, potentially spreading misinformation (Ne´meth et al., 2018).
- Breach of Privacy: Confidential citizen data could be accessed, violating privacy regulations (Ugwu et al., 2022).
- Security Vulnerabilities: Malware could be introduced into PDF objects, compromising the security of the document exchange process (SonicWall Capture Labs, 2018; Stevens, 2023).

To combat these threats, securing policy data with immutable storage is essential. Blockchain technology provides a robust solution for maintaining secure, verifiable, and tamper-resistant policies in e-Governance systems (Shamsan Saleh, 2024).

Traditional PDF signatures face two primary vulnerabilities: (1) attackers can either replace pages with arbitrary content before the signed page or (2) overwrite existing content by adding new elements in unfilled areas of signed pages. While digital signatures are designed to prevent such modifications, a new class of threats, known as shadow attacks, exploits hidden content within signed PDFs (Mainka et al., 2021). In a shadow attack, the attacker creates a shadow document, containing both legitimate content and concealed malicious material, for signing. After the document is signed, the attacker alters it and sends it to the victim. Although the digital signature remains valid, the modified content is displayed in the victim's PDF viewer. Despite this, many PDF viewers reduce the effectiveness of such attacks by alerting users to the presence of hidden content, thereby mitigating the risks.

To further strengthen security, policy-based protection can complement traditional digital signature defenses (Hu, 2014). Access control policies define specific rules for handling PDF content, including validation and access restrictions. For instance, a policy may stipulate that any changes to a signed document must adhere to predefined structural criteria or require additional authentication for sensitive documents. These policies can be stored on a blockchain, ensuring tamper-proof protection. When a document is accessed, the system cross-validates the stored policy with the document's current state, verifying that the document complies with its intended rules and maintaining its integrity.

Both incremental saving attacks and shadow attacks exploit the PDF Incremental Update feature (i.e., providing a method for updating a PDF file without completely rewriting it, where changes are appended to the end of the file or to PDF objects, leaving its original contents intact) to manipulate digitally signed PDFs (Software, 2025; Karsten Meyer zu Selhausen and Dankelmann, 2019). While incremental saving attacks rely on corrupted updates with improperly closed objects, shadow attacks use well-formed, compliant updates, making them harder to detect. Policy-based protection mitigates these risks by enforcing rules that require every update to a signed document to match predefined criteria. For example, policies stored on the blockchain can mandate that all incremental updates undergo validation against the original signed version, preventing the addition of hidden content. This layered approach strengthens defenses against both attack types by detecting and rejecting unauthorized changes early, enhancing document security.

In this study, blockchain is leveraged to store access control policies, making policy tampering in man-in-the-middle (MITM) attacks on exchanged PDF files (Mainka et al., 2021) virtually impossible. As policies are stored as immutable transactions on the blockchain, attackers cannot alter the policy even if they intercept the transmission. Access requests are verified against these tamper-proof policies, ensuring that unauthorized modifications are prevented. This approach provides a secure, unalterable reference for policies, effectively countering the risks of policy tampering in MITM scenarios. Our research aims to address key questions on the effectiveness of blockchain in enhancing document security and preventing unauthorized policy alterations. noitemsep

RQ1 How effective is the blockchain-based RBAC system in ensuring that only authorized users can access sensitive documents?

RQ2 How effective is the proposed approach prevent shadow attacks?

RQ3 How easy is it for an e-government system to transition from manual verification to the proposed model?

The subsequent sections of the paper are structured as follows: Section 2 provides the Background. Section 3 presents An Exploratory Empirical Analysis. Section 4 discusses Known Vulnerabilities and Associated Attacks. Section 5 details Our Approach, including the proposed blockchain-based framework and its operational workflow. Section 6 presents the Evaluation. Section 7 surveys Related Works. Finally, Section 8 concludes the paper and outlines directions for Future Work.

2 Background

E-government leverages information and communication technologies (ICTs) to deliver services ef- ficiently, transparently, and securely. While digitization improves service delivery, it also exposes sys- tems to new cyber threats. Centralized e-government systems remain vulnerable to single points of failure, malware, and denial-of-service (DoS/DDoS) attacks (Cosmin-Iulian and Adrian, 2024; Golightly et al., 2023). In particular, the heavy reliance on Portable Document Format (PDF) files for legal, financial, and regulatory workflows makes their authenticity and integrity central to the trustworthiness of digital governance.

PDFs are structured using indirect objects and unique identifiers, with digital signatures providing authenticity, integrity, and non-repudiation. Legal and regulatory frameworks worldwide recognize such digital signatures, including the U.S. Electronic Signatures in Global and National Commerce Act

(ESIGN Act) (Elisa et al., 2018), India's Information Technology Act amendments for e-documents (Batubara and Janssen, 2018), and the European Union's eIDAS Regulation (Zhao and Zhao, 2010). These frameworks establish that properly signed electronic records carry the same legal standing as handwritten ones. For example, Adobe reported processing over 8 billion digital signature transactions in 2019, covering 15 million documents daily (Layton, 2016), illustrating the scale and criticality of secure PDF use.

2.1 Shadow attack

Digital signatures are designed to prevent tampering, yet research shows that attackers can bypass them through *shadow attacks* (Karsten Meyer zu Selhausen et al., 2019). In such attacks, hidden or alternative content is embedded into a signed PDF and later revealed without invalidating the signature. Unlike dynamic code injection or JavaScript-based attacks, shadow attacks exploit the PDF incremental update feature, allowing modifications to be appended without rewriting the entire file (Software, 2025). This presents a severe risk in e-governance contexts, where signed PDFs such as tax records, land deeds, and policy documents must remain trustworthy.

2.2 Policy-based protection using blockchain in E-governance systems

Beyond document-level attacks, adversaries may manipulate access control policies that govern who can view, edit, or approve sensitive documents. In centralized architectures, such policies are stored on duplicated servers or transmitted across insecure channels, exposing them to manipulation or replay (Sharif, 2024; Wang et al., 2023; Ding and Sato, 2023). For instance, an attacker could temporarily elevate privileges to access restricted tax or land records and then restore the original policy to avoid detection. These vulnerabilities demonstrate the limitations of centralized enforcement mechanisms in safeguarding critical governance data.

2.3 Mitigating MITM attacks in E-governance with blockchain

Blockchain offers a robust defense against document and policy manipulation by ensuring immutability, decentralization, and cryptographic integrity (Roy and Karforma, 2011; Malhotra et al., 2017). Prior studies have acknowledged its potential in strengthening e-governance security; for example Roy and Karforma (2011), emphasize blockchain's role in achieving transparency and reliability in electronic transactions, while Malhotra et al. (2017) highlight decentralized data protection mechanisms for government infrastructures. More recently Ankur and Patel (2022), examine blockchain's contribution to cryptographic assurance and tamper-resistance in policy enforcement. However, these works largely consider blockchain at a system-wide level and do not directly address document-specific vulnerabilities.

In contrast, our framework integrates blockchain with PDF-centric protections by hashing individual PDF components (catalog, pages, content, fonts, metadata) and binding them to access control policies. This enables detection of shadow attacks and incremental saving exploits that bypass conventional digital signatures, while also preventing policy tampering through consensus-based verification. As a result, our approach extends the state of the art by bridging document-level attack prevention with policy-level enforcement in a unified blockchain-based model.

Furthermore, policies in our framework are codified as smart contracts, ensuring that any modification is logged and verified through network consensus. This shifts enforcement from a centralized trust model to a distributed and auditable mechanism, thereby preventing unauthorized privilege escalation and guaranteeing transparency in policy changes. When combined with digital signatures, blockchain provides end-to-end integrity: documents remain tamper-proof, and policies remain consistently verifiable and enforceable.

2.4 Challenges

Despite its potential, adopting blockchain in e-governance introduces challenges. Latency and scalability can affect real-time policy enforcement, while privacy concerns arise from storing sensitive metadata on-chain (Ankur and Patel, 2022). Moreover, integrating legacy e-government systems with blockchain- based architectures requires standardization, interoperability, and careful policy design. These challenges underscore the need for frameworks that balance decentralization, scalability, and compliance with legal regulations while resisting document and policy-level attacks.

3 An exploratory empirical analysis

To comprehend the e-government ecosystem and the progressive growth of digitization in the e- government, we conducted an empirical study using India's open digital datasets called, Digilocker.

RQ1 How effectively does e-government ecosystem is adapting the digitization of agreements or docu- ments?

In response to RQ1, this study examines digitization practices in e-government and identifies security vulnerabilities that may compromise system integrity.

Several initiatives and platforms illustrate the growing role of digital systems in enabling secure and efficient document exchange in governance. The most relevant examples include:

Digital Document Exchange (DDE): Efficiency and Cost Savings–DDE solutions streamline processes across sectors such as finance, healthcare, and administration, reducing physical interactions and operational costs while enabling remote work, particularly valuable in the post-COVID era.

India's Digital Public Infrastructure (DPI) – India Stack exemplifies DPI, integrating digital identity, payment systems, and data exchange layers. Beyond India Stack, initiatives like Electronic Toll Collection highlight the government's broader digital strategy.

DigiLocker: A Key Driver of Transformation–As part of DPI, DigiLocker offers a secure, federated digital locker for citizens to access government-issued documents via Aadhaar authentication and APIs. It replaces the earlier need for in-person verification, reducing costs and improving access, especially in rural areas.

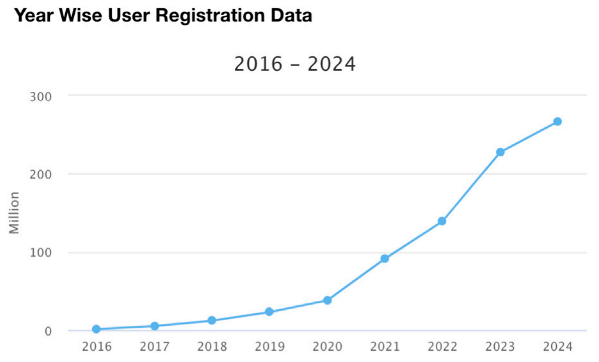


FIGURE 1
User registration statistics (2016–2024) (Corporation, 2024).

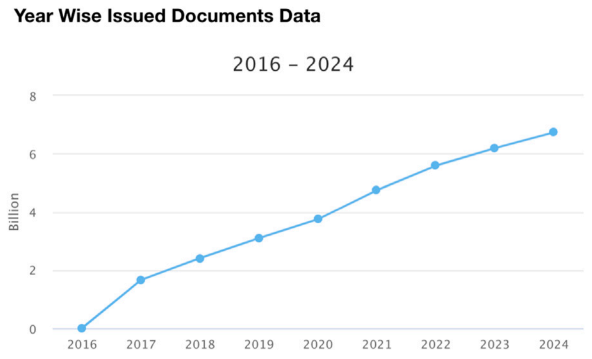


FIGURE 2
Documents issued statistics (2016–2024) (Corporation, 2024).

Impact on Document Exchange–Figure 1 shows steady growth in issued documents and DigiLocker users, while Figure 2 highlights rising paperless transactions. In 2024 alone, about 6.73 billion documents—mainly PDFs—were exchanged electronically, reflecting the scale of India's digital governance.

The choice between centralized and federated approaches shapes the design of e-government systems. Centralized models offer control, streamlined processes, and economies of scale but risk single points of failure and limited adaptability. Federated models distribute services across autonomous entities, supporting flexibility and local customization but introducing challenges in data integration, interoperability, and security.

India adopts a hybrid approach, with the central government providing standards and infrastructure while states retain autonomy in implementation. This balances centralized coordination with decentralized decision-making.

Despite progress toward digitization, e-governance still relies on centralized or federated data gover- nance, exposing systems to risks such as man-in-the-middle (MiTM) and shadow attacks. MiTM attacks threaten sensitive data and critical services, as seen in vulnerabilities affecting e-voting, transmission channels, and infrastructure (Teague and Halderman, 2025; Chigada and Mazhawidza, 2024). Mitigation requires strong encryption, digital certificates, intrusion detection, audits, and user training to ensure system integrity and protect citizens' data.

4 Known vulnerabilities and associated attacks

In consortium-based workflows, records such as agreements and contracts are typically prepared by one participant, then circulated for final review and signature by all authorized members. This collaborative process, while efficient, can be exploited if malicious content is inserted during document preparation. Figure 3 illustrates such a scenario, where hidden "shadow content" embedded in a PDF may remain undetected during signing and later be revealed, compromising the integrity of the agreement.

Shadow attacks exploit features in document formats, such as PDFs, to introduce hidden elements while maintaining the document's apparent integrity. For example, attackers can embed invisible layers or annotations that bypass casual inspection but remain part of the signed document. This risk extends to widely used cloud signing services like Adobe Cloud, Document Sign, and Digital Signature Service, which may inadvertently sign documents containing such shadow content (Haber and Stornetta, 1991).

Digital signatures are designed to ensure document integrity by enabling verification through hash comparisons. A recipient recalculates the cryptographic hash of the document, decrypts the signature to retrieve the original hash, and compares the two. Any mismatch indicates alterations, rendering the signature invalid (Nissim et al., 2015). However, digital signatures cannot detect the presence of shadow content unless such content alters the visible document or its hash representation. This issue is particularly relevant in digital governance, where secure and transparent inter- agency document exchange is critical. Collaborative platforms reduce paperwork and streamline approval processes but are vulnerable to shadow attacks if proper safeguards are not implemented.

In summary, while digital signatures are essential for verifying document integrity, they are not foolproof against sophisticated shadow attacks. Strengthening digital workflows, particularly in critical domains like e-government, requires ongoing research and the adoption of advanced security techniques to counter such threats.

In the process of inter-agency document exchange, Agency A prepares a procurement proposal and submits it through a secure e-government portal to Agency B. The proposal is reviewed to ensure it aligns with regulatory requirements and project goals. If approved, Agency B may request modifications before Agency A resubmits the revised proposal. Throughout this process, digital signatures are used to verify the integrity of documents and facilitate communication. However, shadow attacks pose a significant threat to the security of these transactions. In a shadow attack, attackers create a PDF with two layers of content: one visible to the signing agency and another hidden layer that remains inaccessible during review.

Once the document is signed by the authorized agency, the attackers can alter the document to reveal the hidden content, while preserving the original signature's validity. This

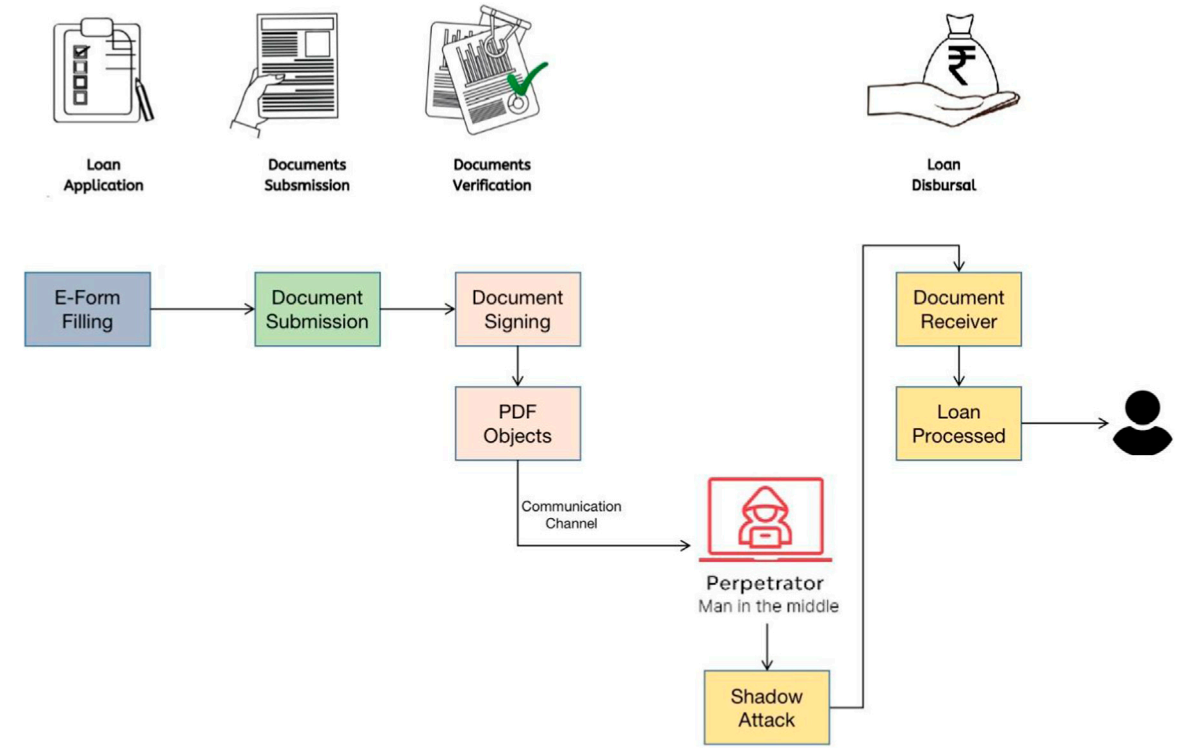


FIGURE 3
Demonstrating Shadow Attack Scenario. Since the E-Government scenario lacks the validation of PDF file based on its content, a shadow attack induced by MITM can modify the file. The shadow attack is performed on the PDF files even after file is protected by digital signature.

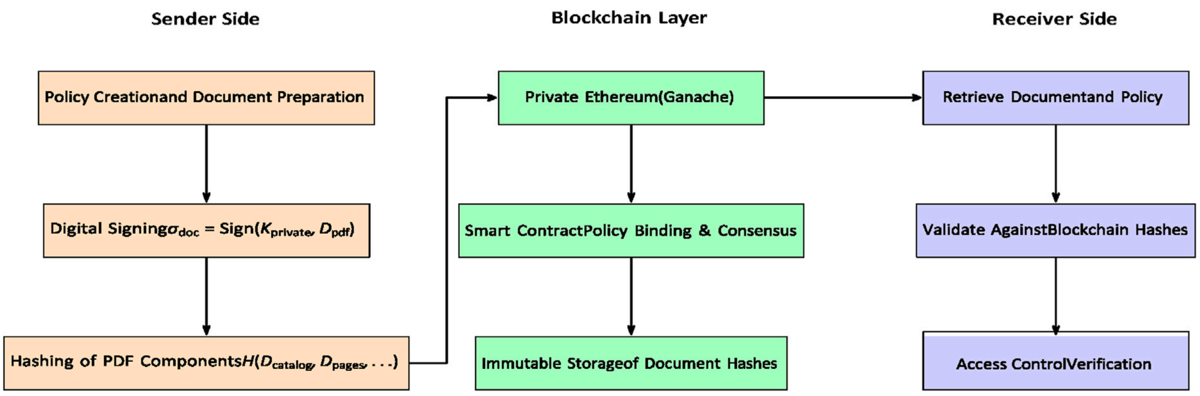

Sender Side
Blockchain Layer
Receiver Side
Policy Creationand Document Preparation
Digital Signingσ_{doc} = Sign($K_{private}$, D_{pdf})
Hashing of PDF Components$H(D_{catalog}, D_{pages}, . . .)$
Private Ethereum(Ganache)
Smart ContractPolicy Binding & Consensus
Immutable Storageof Document Hashes
Retrieve Documentand Policy
Validate AgainstBlockchain Hashes
Access ControlVerification

FIGURE 4
Proposed blockchain-based methodology showing sender-side operations, blockchain-based storage and verification, and receiver-side validation and access control. Inter-layer arrows connect from the bottom of one layer to the top of the next, illustrating hierarchical data flow.

manipulation goes undetected by the victim's PDF viewer, which still considers the digital signature valid, even though the content has been altered. The victim sees the malicious or altered content, unaware that the document has been tampered with. This vulnerability highlights the need for more robust validation techniques to detect such shadow content and prevent misuse of digital signatures, which are essential for ensuring document integrity in e-government and other secure digital exchanges (Mainka et al., 2021).

5 Our approach

Our approach outlines a secure and efficient framework for the exchange of documents and associated access policies, designed to mitigate the risks of tampering and unauthorized access (shown in Figure 4). At the heart of this model is the integration of a permissioned Ethereum-based blockchain, selected to align with e-governance requirements for controlled participation, auditability, and inter-agency trust. The blockchain ensures immutability and traceability of both documents and policies throughout their lifecycle, with each validator node maintaining a synchronized and tamper-resistant copy of the ledger.

The overall architecture is divided into two operational phases: the sender-side and the receiver-side. On the sender's side, documents and corresponding access policies are created, digitally signed, and securely uploaded to the blockchain, where each individual document component is hashed and recorded to preserve integrity. On the receiver's side, the retrieved document and policy are validated against their blockchain entries, with strict access control enforcement ensuring that only authorized users can view or modify the data. This design ensures that any modification or unauthorized access attempt during transmission or storage is immediately detectable, thus protecting both the confidentiality and authenticity of sensitive information.

Figure 4 illustrates the complete workflow of the proposed blockchain-based framework. The left section represents the sender-side operations, where policy creation, digital signing, and document hashing occur. The central layer depicts the blockchain infrastructure, showing the permissioned Ethereum network where policies and hashed document components are immutably stored and verified through smart contracts. The right section highlights the receiver-side operations, involving document retrieval, validation against blockchain records, and access control verification. The figure also shows inter-layer arrows connecting the bottom of one layer to the top of the next, indicating a sequential and hierarchical flow of secure communication and verification between entities.

By leveraging blockchain technology for immutable storage, cryptographic hashing, and digital signa- tures, this framework ensures that document exchange remains secure, transparent, and auditable. Both sender and receiver can therefore exchange sensitive e-governance documents with full assurance of integrity and trustworthiness.

6 Blockchain configuration and implementation environment

The proposed framework relies on a permissioned Ethereum-based blockchain to ensure secure, au- ditable, and tamper-proof document and policy exchange. This infrastructure was selected because it combines the flexibility of Ethereum smart contracts with the controlled participation required for e- governance systems.

Network Setup: A private Ethereum network was created using Ganache (v7.9.1), configured with 10 validator nodes representing different government departments. Each node maintained a synchronized copy of the blockchain, ensuring redundancy and consensus-based validation of all policy and document entries.

Consensus Mechanism: The network employed a Proof-of-Authority (PoA) consensus model, which provides faster block validation and reduced computational overhead compared to Proof-of-Work. This design is well suited for government-led blockchain systems, where participants are pre-verified and trusted.

Smart Contract Layer: Policies and document metadata were managed through smart contracts written in Solidity (v0.8.21) and deployed via Remix IDE. These contracts enforced role-based access control (RBAC), verified document hashes, and logged access attempts immutably. The blockchain was interfaced with the application layer using Web3.py, enabling seamless integration with document exchange workflows.

Integration with Document Exchange: The blockchain infrastructure worked alongside the document- processing module. Each PDF document was signed, hashed using SHA-256, and its metadata (hash, timestamp, policy reference) recorded on the blockchain. On the receiver side, document integrity and policy compliance were cross-verified against the blockchain-stored values.

By placing the blockchain at the core of the framework, the infrastructure provides a secure backbone for policy enforcement, data immutability, and tamper detection across all e-governance transactions. This configuration provides a permissioned blockchain setup—ensuring that only authenticated government entities can participate in policy registration and document validation—while maintaining scalability and auditability similar to public Ethereum environments.

7 Sender-side operations

The sender-side operations define the sequence of activities performed by the originating department to prepare and transmit secure documents within the proposed blockchain-based framework. This phase ensures that each PDF and its corresponding access policy are properly created, authenticated, and immutably recorded before transmission. The following steps outline the detailed workflow of the sender's process:

7.1 Policy creation and document preparation

- The sender, operating within Department 1, creates a static access policy, denoted as P_{static}. This policy specifies the following parameters:
- Authorized$_i$ is the list of individuals or entities authorized to access the document.
- Access Level$_i$ defines the permissible access type for each authorized user, such as view-only or edit permissions.
- Approval Requirement$_i$ defines any necessary conditions for accessing or modifying the document (e.g., manager approval).

We use the following notation to describe the events:

- D_{pdf}: Represents the entire document content in PDF format.
- σ_{doc}: The digital signature of the document, used to ensure its authenticity and non-repudiation.
- K_{private}: The sender's private key, used to generate the digital signature.

The document, typically in PDF format, is created and digitally signed. The digital signature ensures the authenticity and non-repudiation of the document, denoted as σ_{doc} (given in Equation 1). This signature is generated using a private key K_{private}, as follows:

$$\sigma_{\text{doc}} = \text{Sign}\left(K_{\text{private}}, D_{\text{pdf}}\right) \tag{1}$$

where D_{pdf} represents the PDF content.

7.2 Document validation and blockchain uploading

Before uploading to the blockchain, the document is parsed by extracting the following components (given in Equation 2):

$$D_{\text{catalog}}, D_{\text{pages}}, D_{\text{content}}, D_{\text{fonts}}, D_{\text{metadata}} \tag{2}$$

where.

- D_{catalog}: The catalog structure of the PDF, which organizes document contents and metadata hierar- chically.
- D_{pages}: Represents the individual pages of the PDF document.
- D_{content}: Refers to the textual and graphical content within the PDF.
- D_{fonts}: Specifies the font information and typefaces used in the PDF.
- D_{metadata}: Contains metadata about the document, such as author, creation date, and title.
- $H(D_{\text{component}})$: The hash of a specific PDF component (e.g., catalog, pages), used for integrity verification (given in Equation 3).

Each component is validated to ensure integrity. The hashed values of these components are given by:

$$H\left(D_{\text{content}}\right) = \text{Hash}\left(D_{\text{content}}\right) \tag{3}$$

These hashed values, along with the static policy P_{static}, are stored in the blockchain. This ensures that the individual components and the associated policy are immutable and traceable.

The data is uploaded to the blockchain in the form of a transaction T_{upload} (given in Equation 4), which is defined as:

$$T_{\text{upload}} = \Big(H\left(P_{\text{static}}\right), H\left(D_{\text{catalog}}\right), H\left(D_{\text{pages}}\right), H\left(D_{\text{content}}\right), H\left(D_{\text{fonts}}\right), H\left(D_{\text{metadata}}\right)\Big) \tag{4}$$

7.3 Blockchain storage

The blockchain records the document and policy components as a block, denoted as B. Each block is cryptographically hashed to maintain data integrity:

$$\text{B} = \begin{pmatrix} H\left(P_{\text{static}}\right), H\Big(D_{\text{catalog}}\Big), H\Big(D_{\text{pages}}\Big) \\ H\left(D_{\text{content}}\right), H\left(D_{\text{fonts}}\right), H\left(D_{\text{metadata}}\right) \\ T_{timestamp}, H_{block} \end{pmatrix}$$

Where:

- $T_{\text{timestamp}}$ is the timestamp of the block.
- H_{block} is the hash of the entire block.
- P_{static}: The static policy file, which defines access rules and conditions for the document.
- $T_{\text{timestamp}}$: The timestamp of when a block is created or recorded in the blockchain.
- H_{block}: The cryptographic hash value representing the integrity of the entire block.

The block is linked to the previous block via a hash pointer, maintaining a secure chain of transactions.

8 Receiver-side operations

We thank the reviewer for this valuable observation. Several lists in the Our Approach section began without an introductory statement, which may interrupt the logical flow for readers. We have added concise introductory sentences before each list—specifically under SENDER-SIDE OPERATIONS, RECEIVER- SIDE OPERATIONS, and GRANULAR ACCESS CONTROL ALGORITHM, to clearly indicate what the subsequent points represent. These additions ensure that each list is properly contextualized and that transitions between narrative text and structured steps are smooth and coherent.

8.1 Dynamic policy and document retrieval

The receiver downloads the document and its associated dynamic policy, denoted P_{dynamic}, via a poten- tially insecure communication channel. The receiver's system needs to validate this data to ensure integrity and authenticity. Where:

- P_{dynamic}: The dynamic policy file received via the communication channel, which is cross-checked with P_{static}.

8.2 Policy validation and access control

Upon receiving P_{dynamic}, the Document Validator Module cross-checks it with the blockchain-stored static policy P_{static} (given in Equation 5). If the two policies match, it confirms the integrity of the policy:

$$\text{Valid}\Big(P_{\text{dynamic}}\Big) = \text{True if } H\left(P_{\text{static}}\right) = H\Big(P_{\text{dynamic}}\Big) \tag{5}$$

Where:

- Valid(P_{dynamic}): A Boolean value indicating whether the received dynamic policy matches the blockchain- stored static policy.

If the policies mismatch, access is denied, and the channel is flagged as untrusted.

8.3 PDF object validation

The receiver also validates the received PDF components (given in Equation 6). The extracted components from the received document are:

$$D^{\text{received}}_{\text{catalog}}, D^{\text{received}}_{\text{pages}}, D^{\text{received}}_{\text{content}}, D^{\text{received}}_{\text{fonts}}, D^{received}_{metadata} \tag{6}$$

Where:

- $D^{received}_{catalog}$: The catalog structure of the received PDF.
- $D^{received}_{pages}$: The individual pages of the received PDF.

- $D^{received}_{content}$: The textual and graphical content of the received PDF.
- $D^{received}_{fonts}$: The font information in the received PDF.
- $D^{received}_{metadata}$: The metadata of the received PDF.
- $\text{Valid}\,(D^{received}_{component})$: A boolean value indicating whether a received component matches its blockchain counterpart.

Each component is hashed and compared to its blockchain counterpart (given in Equation 7):

$$\text{Valid}\left(D^{received}_{component}\right) = \text{True if H}\left(D^{received}_{component}\right) = \text{H}\left(D_{component}\right) \qquad (7)$$

If any inconsistency is detected, the document is rejected. component

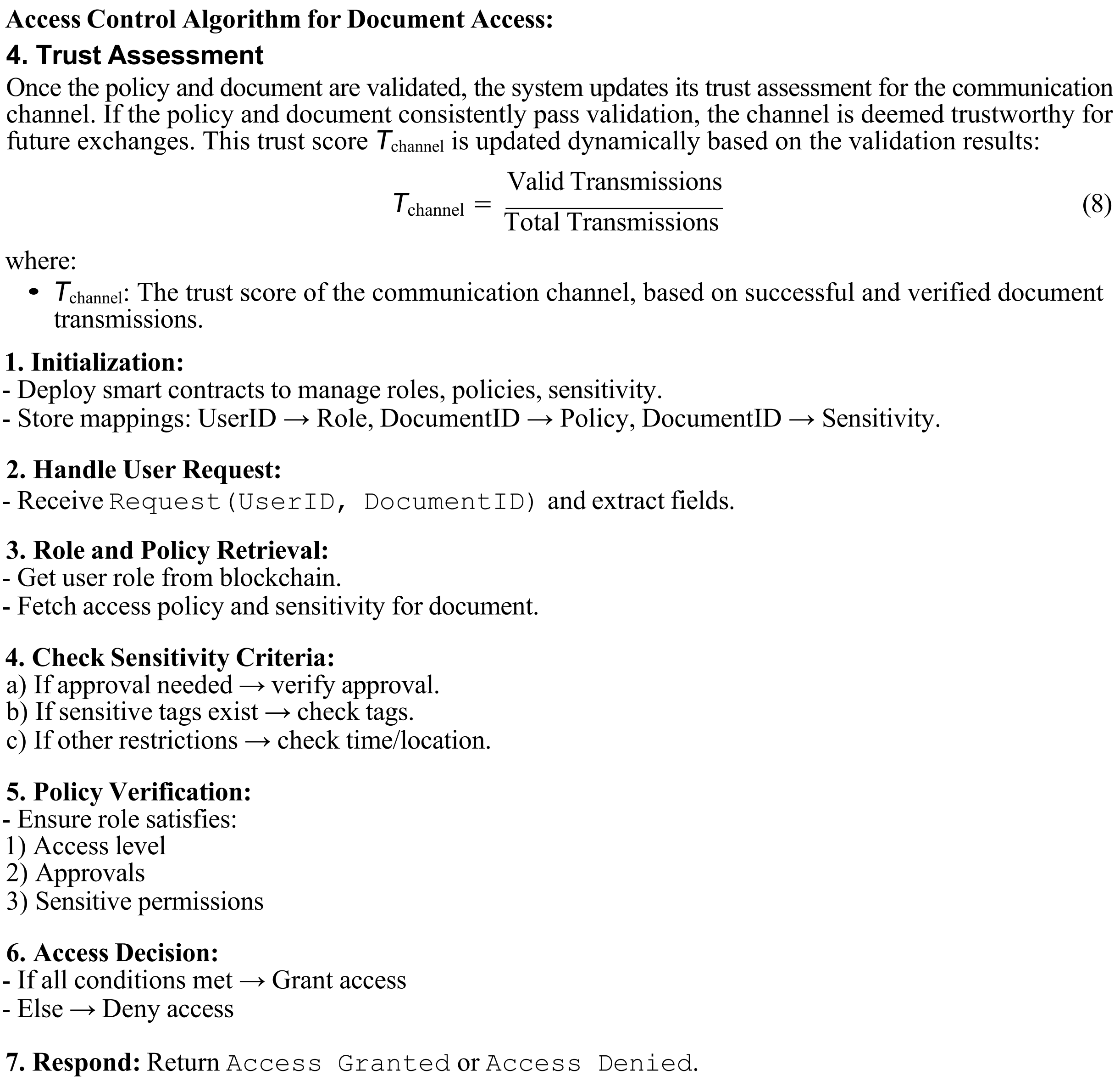
A higher score indicates a more reliable channel for future document exchanges.

This architectural flow demonstrates how blockchain integration enhances document and policy man- agement by ensuring tamper-proof storage and validation. The sender creates robust static definitions, the blockchain preserves integrity, and the receiver performs stringent checks on incoming data. This approach provides a reliable, secure framework suitable for sensitive e-governance applications, protecting against tampering, unauthorized access, and compromised communication channels.

Figure 5 demonstrates the methodology of extracting the PDF file receiver's information controlled by access control policies, as explained below.

9 Granular access control algorithm

1. Initialization: The system begins by setting up smart contracts on the blockchain. These contracts store key information such as user roles, access control policies for documents, and sensitivity



Access Control Algorithm for Document Access:

4. Trust Assessment

Once the policy and document are validated, the system updates its trust assessment for the communication channel. If the policy and document consistently pass validation, the channel is deemed trustworthy for future exchanges. This trust score T_{channel} is updated dynamically based on the validation results:

$$T_{\text{channel}} = \frac{\text{Valid Transmissions}}{\text{Total Transmissions}} \qquad (8)$$

where:

- T_{channel}: The trust score of the communication channel, based on successful and verified document transmissions.

1. Initialization:
- Deploy smart contracts to manage roles, policies, sensitivity.
- Store mappings: UserID → Role, DocumentID → Policy, DocumentID → Sensitivity.

2. Handle User Request:
- Receive `Request(UserID, DocumentID)` and extract fields.

3. Role and Policy Retrieval:
- Get user role from blockchain.
- Fetch access policy and sensitivity for document.

4. Check Sensitivity Criteria:
a) If approval needed → verify approval.
b) If sensitive tags exist → check tags.
c) If other restrictions → check time/location.

5. Policy Verification:
- Ensure role satisfies:
1) Access level
2) Approvals
3) Sensitive permissions

6. Access Decision:
- If all conditions met → Grant access
- Else → Deny access

7. Respond: Return `Access Granted` or `Access Denied`.

FIGURE 5
Access control algorithm.

criteria. For example, a document might be labeled as “Sensitive,” requiring extra approval before access is granted.

- Smart contracts store:
  - User roles.
  - Access control policies for documents.
  - Sensitivity criteria (e.g., whether a document requires signed approval or contains sensitive data).

2. User Request Handling: When a user attempts to access a document, the system receives a request containing the user’s ID and the document’s ID. For instance, User1 requests access to “DocumentA.”
   - The system receives an access request Request(UserID, DocumentID).
   - Extract UserID and DocumentID from the request.
3. Role Retrieval: The system queries the blockchain to find out what role the user holds. If User1 has a role of “Authorized Personnel,” the system proceeds to check the access policies related to that role.
   - The Role enum defines three possible roles:
     - None.
     - RegularUser.
     - AuthorizedPersonnel.
   - The userRoles mapping assigns roles to users by their Ethereum address.
4. Policy Retrieval: The system retrieves the access control policy for the requested document. If “DocumentA” is marked as requiring “Signed Approval” or contains “Sensitive Data,” these conditions will be taken into account during the next steps.
   - Retrieve the access control policy for the requested document.
   - If the document is marked as requiring “Signed Approval” or contains “Sensitive Data,” proceed to sensitivity checks.
5. Document Sensitivity Checks: At this stage, the system checks if the document has additional sensitivity criteria. For example, if “DocumentA” contains sensitive data, the system may require that the user’s role includes permission to access sensitive data or that they have approval from a higher authority. If a document needs signed approval, the system checks whether the user has the necessary signed approval before allowing access.
   - The Document struct is represented as:
     - id: The unique identifier of the document.
     - Name: The document name (for example, “DocumentA”).
     - RequiresSignature: A Boolean flag indicating whether the document requires a signed approval before access.
     - isSensitive: A Boolean flag indicating whether the document contains sensitive data.
6. Policy Verification: The system then checks if the user’s role allows access to the document based on the policy.
   - If the document requires a signature, verify whether the user has the necessary signature (passed as the hasSignature parameter).
   - If the document is sensitive, only users with the “AuthorizedPersonnel” role are granted access.
   - If the document is not sensitive, any authorized user (including regular users) can access it.
7. Access Decision: If the user’s role satisfies the policy, and any sensitivity checks (like required signatures) pass, the system grants access to the document. If the checks fail, access is denied.
   - The AccessRequest event is emitted every time a user attempts to access a document, logging whether the access was granted or denied.
8. End: Finally, the system responds to the user with either an “Access Granted” or “Access Denied” message, depending on whether they met all the policy and sensitivity conditions.

The algorithm described offers a robust and granular approach to access control based on both user roles and document sensitivity. The system uses smart contracts deployed on the blockchain to store immutable data regarding user roles, document access policies, and sensitivity criteria. It ensures that access decisions are made not only by verifying the user’s role but also by evaluating the sensitivity of the requested document.

10 Example scenario

Let’s say we have two users:

- User1 (Authorized Personnel - Admin): Can access sensitive documents if they have signed approval.
- User2 (Regular User - Developer): Cannot access sensitive documents, even if they have a valid role.

The sample policy file is given as follows:

```
{
    “documents”: [
      {
        “documentId”: 1,
        “name”: “Confidential Report”,
        “isSensitive”: true,
        “requiresSignature”:             true,
“validFrom”: “2024-11-01T00:00:00Z”,
        “validUntil”: “2024-11-30T23:59:59Z”,
        “accessPolicy”: {
          “rolesAllowed”:
[“Admin”, “Manager”],
          “accessConditions”: {
            “timeBased”: {
              “start”: “2024-11-01T00:00:00Z”,
                “end”: “2024-11-30T23:59:59Z”
              }
          }
      }
    }
  ],
“users”: [
    {
        “userId”: “user1”,
        “roles”: [“Admin”]
    },
    {
        “userId”: “user2”,
        “roles”: [“Developer”]
    }
    ]
}
```

11 Access control example

The `Confidential Report`is a sensitive document that can only be accessed by users with the roles `Admin`or `Manager`.

User1, who has the `Admin`role, would be able to access this document, provided that they are within the valid access period from `2024-11-01`to `2024-11-30`.

On the other hand, if User2 attempts to access the same document, the system checks User2's role and determines they are not allowed to access sensitive documents. As a result, the system denies the request. Thus the proposed system ensures secure and policy-driven access to documents, using blockchain's immutability to prevent tampering and ensuring that access is granted only based on predefined roles and document sensitivities.

12 Evaluation

For our evaluation, we have taken the following e-governance scenario which we have simulated in realtime and tested by exchanging the PDF files between systems. Firstly, let us describe the test environment and the dataset in detail.

12.1 Overview of the test environment

The test environment simulates a complete e-governance document management system, designed to capture the lifecycle of sensitive records such as contracts, certificates, permits, and tax forms. In this environment, documents are not only created and digitally signed but also validated, exchanged, and audited across multiple government departments. To achieve this, the setup integrates two main components: Hardware and Software Setup: Experiments were conducted on a workstation with an Intel Core i7 processor, 16 GB RAM, 512 GB SSD, running Ubuntu 22.04 LTS and Python 3.10. The blockchain environment was implemented using Ganache (v7.9.1) to simulate a permissioned Ethereum-based network with 10 validator nodes representing different government agencies. Each node maintained a full copy of the ledger, ensuring immutability and synchronization.

Blockchain Layer: The blockchain employed a Proof-of-Authority (PoA) consensus mechanism to facilitate fast and deterministic block validation. Smart contracts were developed in Solidity (v0.8.21) using Remix IDE and integrated with the document-processing module via Web3.py. The contracts handled access control enforcement, document hash storage, and policy verification. Blockchain Layer: A permissioned Ethereum-based blockchain implemented using Ganache (v7.9.1), configured with 10 validator nodes. Each node represents a government agency and maintains a full copy of the ledger, ensuring immutability, decentralization, and synchronization. Policies are codified as smart contracts to enforce access control and log all modification attempts.

Application Layer: The document exchange system was developed in Python using Flask to simulate sender–receiver communication. Each transaction involved:

1. Uploading a signed PDF document from the sender node,
2. Hashing its content using SHA-256,
3. Storing document metadata and policy hashes on the blockchain, and
4. Validating document integrity on the receiver node.

Security and Validation Components: The communication between sender and receiver nodes was tested under simulated Man-in-the-Middle (MITM) attack conditions using the Scapy network analysis tool to verify blockchain-based tamper resistance. The system also utilized OpenSSL for digital signature generation and verification in baseline PKI tests.

This configuration collectively demonstrates a complete e-governance simulation, combining digital document workflows, blockchain-based policy enforcement, and security validation within a controlled experimental environment.

12.2 Sample dataset details

The dataset represents the test environment, tracking the outcomes of shadow attack tests on 1,000 digitally signed PDFs. Below, we detail the structure and components of the dataset, including the number of documents, signing method, blockchain storage, and attack implementation.

12.2.1 Number of PDFs: 1,000 official documents

The evaluation was conducted using a robust test environment comprising 1000 PDF documents, carefully designed to reflect the diversity and complexity of real-world e-governance systems. This dataset includes various official documents commonly exchanged within public administration, such as contracts, certificates, permits, and tax forms. Specifically, the corpus consists of 300 contracts (e.g., service agreements, procurement contracts), 300 certificates (e.g., birth certificates, educational diplomas), 200 permits (e.g., building permits, business licenses), and 200 tax forms (e.g., income tax returns, VAT declarations). Each document is embedded with sensitive informa- tion—such as contractual terms, certificate issuer details, permit conditions, and taxpayer data—making the integrity and authenticity of these files critical for secure governance operations. The dataset is intentionally scaled from 100 to 1,000 PDFs to enhance the reliability and generalizability of the evaluation, simulating a high-volume document management scenario typical in e-governance deployments. In terms of structure and complexity, the documents vary widely: file sizes range from 100 KB to 10 MB, encompassing both single-page documents (e.g., simple certificates) and multi-page contracts exceeding 10 pages. Additionally, many PDFs include advanced features such as embedded images, interactive forms, and annotations—components that are often exploited in sophisticated shadow attacks. This diverse distribution ensures that the testing environment thoroughly covers the spectrum of PDF features and vulnerabilities, providing a comprehensive assessment of the proposed framework's security effectiveness across different document types and structures.

12.2.2 Signing and blockchain storage methodology

In this study, each of the 1,000 e-governance PDFs was digitally signed using a Public Key Infrastructure (PKI) system employing

RSA-2048 encryption and SHA-256 hashing, ensuring authenticity and origin verification. The PKI system leverages a trusted Certificate Authority (CA) that issues digital certificates linking public keys to specific government agencies, while the associated private key is used to sign the PDF files. For each document, selected objects—typically the main static content excluding dynamic elements such as annotations, forms, and incremental updates—are hashed using SHA-256, generating a unique 256-bit digest. This digest is encrypted with the signer's private key using RSA-2048 to produce the digital signature, which is embedded within the PDF's signature dictionary. The use of RSA-2048, with its 2048-bit key size, provides robust resistance against brute-force attacks, while SHA-256 maintains cryptographic integrity. However, standard PKI signatures hash only predefined portions of a PDF, leaving unsupervised structures (e.g., annotations, embedded files, incremental saves) vulnerable to shadow attacks that can modify document content without invalidating the signature. To simulate these real-world attacks during evaluation, custom Python scripts were developed based on documented PDF specification-level vulnerabilities (e.g., Bo¨ck et al., 2020), targeting areas such as annotation injection, incremental saves, and embedded file tampering.

To address these vulnerabilities, a blockchain-based framework was implemented, employing an Ethereum- based private chain to store immutable PDF metadata for tamper-proof verification. Unlike traditional PKI, this framework computes and records a SHA-256 hash of the entire PDF file, including all objects, incremental revisions, and embedded elements, thereby ensuring detection of even the most stealthy modifications. The stored metadata includes the document hash, a revision history log recording every incremental update, and detailed information about embedded files, such as file types, sizes, and their own SHA-256 hashes, enabling the identification of malicious file injections. The blockchain implementation uses smart contracts to manage metadata storage and verification, allowing any party to cross-verify a document by recomputing its hash and comparing it with the blockchain-stored reference. Specifically, after PKI signing, the full PDF file is hashed using SHA-256, and the resulting metadata is transmitted as a blockchain transaction, timestamped and stored immutably. Verification involves recomputing the PDF's hash and querying the smart contract to ensure an exact match. Blockchain transactions for 1,000 PDFs achieved an average confirmation time of X seconds and incurred an average gas cost of Y units per document, demonstrating scalability through batch transaction mechanisms and optimized contract design.

Through our evaluation we address the following research questions:

RQ2 How effective is the proposed approach prevent shadow attacks?

RQ3 How easy is it for an e-government system to transition from manual verification to the proposed model?

12.3 RQ2 - How effective is the proposed approach prevent shadow attacks?

Three shadow attack types are executed to test vulnerabilities in traditional PKI signatures and the robustness of the blockchain framework. Each attack exploits PDF specification weaknesses, allowing modifications without invalidating the PKI signature. The proposed blockchain-based verification frame- work proves highly effective against shadow attacks, with strong detection rates (93.1% recall) and minimal false positives (98.4% precision). It outperforms traditional PKI signature verification, which cannot detect such tampering at all. However, for complete protection, addressing the remaining stealth attack vectors (23 missed cases) will require concrete enhancements. First, the hashing scope should be refined to include the full incremental-update chain, cross-reference tables, object streams, metadata, and attachments, with canonicalization to eliminate benign serialization tricks. Second, structure-aware validation must be applied to high-risk components such as annotations, form fields, optional content layers, transparency groups, and font encodings, while disallowing or flagging embedded scripts and external references. Finally, semantic checks can be incorporated by rendering the signed document to a canonical image and comparing it against a trusted baseline, or by using multiple rendering engines to detect discrepancies. Together, these measures strengthen protection against invisible or cleverly crafted manipulations that bypass standard signature verification.

12.3.1 Incremental update attack

To evaluate the resilience of the proposed framework against shadow attacks, this study specifically targeted the PDF's incremental saving feature, a well-documented vector for stealthy content manipulation. In such attacks, adversaries exploit the fact that PDFs allow new content to be appended incrementally without modifying previously signed objects, thereby preserving the validity of existing digital signatures.

For example, in a typical attack scenario, a legally signed contract PDF is maliciously modified to append an additional page with altered terms—such as "Payment Terms: Waived"—while the crypto- graphic signature over the original content remains intact and appears valid during verification. In our implementation, the attack was conducted using a combination of open-source tools, including QPDF, and custom Python scripts designed to automate incremental save operations. Each modified PDF had a new page or embedded object (such as forged text blocks or images) inserted in a manner that mimicked legitimate additions, making them difficult to detect through casual visual inspection. To ensure broad coverage across diverse document types, this shadow attack simulation was systematically applied to a subset of 333 PDFs from the test environment: specifically, 100 contracts, 100 certificates, 83 permits, and 50 tax forms. This distribution was chosen to reflect realistic attack patterns targeting critical e-governance documents while ensuring the evaluation dataset covered varying structural complexities, including multi- page contracts and single-page certificates. By doing so, the test environment faithfully emulated real-world shadow attack scenarios, providing a robust basis for validating the effectiveness of the blockchain-based defense mechanism.

Evaluation Metrics: From this, the evaluation metrics are calculated as:

- Precision = TP/(TP + FP)
- Recall = TP/(TP + FN)
- F1-Score = 2 × (Precision × Recall)/(Precision + Recall)

TABLE 1 Classification report for evaluating model performance.

Class	Precision	Recall	F1-score	Support
Clean	0.813	0.952	0.877	105
Attacked	0.984	0.931	0.957	333
Accuracy			0.936	438

12.3.1.1 Observation

The observation is made based on on Table 1:

- Macro Avg F1: 0.917
- Weighted Avg F1: 0.938
- Precision for Attacked (0.984): Almost no false alarms when the system says it's attacked.
- Recall for Attacked (0.931): Caught most attacks, missed a few.
- F1 for Attacked (0.957): Overall strong attack detection performance.
- The Clean class had some false positives (precision 0.813), but recall was high (0.952).

In this evaluation, The proposed system was tested against 333 PDFs deliberately modified through shadow attacks. The goal was to detect stealthy manipulations while avoiding false alarms on clean documents.

12.3.1.2 Results

Out of 333 attacked PDFs, the system correctly detected 310 attacks and missed 23 (meaning these 23 altered files slipped through undetected). Among 105 clean PDFs (documents with no attack), the system correctly passed 100 but mistakenly flagged 5 PDFs as attacked.

12.3.1.3 Reason behind false negatives (missed 23 attacks)

To address the limitations identified in detec- tion accuracy, several enhancements can be considered. First, incorporating semantic-aware or structure- aware PDF analysis would enable the system to detect invisible text insertions, transparent objects, or layered manipulations that evade simple hashing. Second, a hybrid verification approach—where the blockchain records not only cryptographic hashes but also higher-level structural fingerprints—can improve resilience against stealthy incremental updates. Third, machine learning–based anomaly detection models trained on benign versus tampered PDF features can provide adaptive detection capabilities, reducing false negatives. Finally, integrating privacy-preserving techniques such as zero-knowledge proofs (ZKPs) could allow verifiable validation without exposing sensitive document contents, further strengthening trust in large-scale e-governance deployments.

False negatives occur when the system fails to detect actual attacks, allowing malicious changes to go unnoticed. One common reason is the use of stealthy or subtle changes by attackers. For example, an attacker might insert an extra clause into a signed contract using invisible text (text colored white on a white background) or embed hidden objects deep within the file structure. These minimal and cleverly crafted changes are difficult to spot because they do not alter the visible appearance of the document, allowing them to bypass detection thresholds that focus only on significant structural differences.

- In these 23 files, the attacker manipulate only the incremental updates or append minimal hidden content (e.g., invisible text or embedded objects) without altering the core objects that we use to in our hashing system.
- In such cases, the original signed objects (which the hash covers) remain untouched, so when our system computes the hash, it matches the original hash—leading the system to incorrectly believe the document is safe.

12.3.1.4 False positives (wrongly flagged 5 clean PDFs)

Sometimes, false positives happen because clean documents go through normal updates that unintentionally resemble malicious changes. For example, many official documents like certificates or permits include versioning features. A government agency might issue a certificate and later append a small audit log or a new version note like "Reissued: April 2025". This is a legitimate update, but structurally, it looks similar to how an attacker would append malicious content, leading The proposed system to mistakenly flag it as an attack.

Additionally, noise in static hashing can also cause false alarms. The proposed detection system uses SHA-256 hashing to compare the document's main content. However, even small and harmless changes—such as adjusting the font slightly, updating the document's metadata (like author name or last modified date), or reordering invisible objects inside the file—can change the hash. For instance, if a user opens a PDF, tweaks the zoom settings, and re-saves it, the internal structure may shift just enough to produce a different hash, even though the visible content remains the same. This mismatch makes the system think the document was tampered with, triggering a false positive.

12.3.2 Object replacement attack

To further assess document integrity vulnerabilities, the framework was tested against policy tampering attacks that exploit unprotected PDF objects—specifically, annotations and embedded images that are typically excluded from standard signature coverage. In this form of attack, adversaries manipulate auxiliary content to subtly alter the document's meaning without invalidating its digital signature. A representative example involves modifying an annotation within a government-issued permit PDF: for instance, changing the status from "Approved" to "Rejected". Because annotations and similar dynamic objects are often omitted from the cryptographic hash during signing, such tampering leaves the original signature intact and verifiable, deceiving recipients into trusting a compromised docu- ment. For this study, tampering was implemented using a combination of PyPDF2 (a Python library for PDF manipulation), Adobe Acrobat Pro (for object-level editing). This policy tampering simulation was systematically applied to another distinct subset of 333 PDFs drawn from the test dataset, comprising 100 contracts, 100 certificates, 83 permits, and 50 tax forms. This attack set was carefully designed to encompass a range of document structures and object types, from single-page certificates to multi- layered contracts containing multiple annotations. By executing controlled object replacements at scale, this evaluation effectively reproduced real-world policy tampering threats, thereby providing a rigorous basis for testing the proposed blockchain-based verification mechanism's ability to detect such stealth alterations.

In this evaluation, the framework was tested against 333 PDFs that were deliberately tampered through policy tampering attacks, targeting unprotected objects like annotations and embedded images. Out of the 333 attacked documents, the system successfully detected 295 cases but missed 38 attacks, where the tampering went unnoticed. Additionally, when tested against 105 clean PDFs, the system correctly cleared 99 documents but mistakenly flagged 6 clean files as attacked.

The detection performance was quantified using standard metrics:

- Precision: 98.0% — meaning that when the system flagged a document as tampered, it was correct nearly all the time.
- Recall: 88.6% — indicating that while the system caught most attacks, it missed about 11.4% of them.
- F1-Score: 93.0% — reflecting a solid balance between precision and recall.
- Overall Accuracy: 91.9% — showing that approximately 92 out of every 100 documents were correctly classified.

12.3.2.1 Reasons behind false negatives (missed 38 attacks)

False negatives occurred mainly because some policy tampering attacks were subtle and deeply embedded within the document structure. For instance, attackers modified annotations such as changing the word "Approved" to "Rejected" using invisible or transparent layers, which do not alter the main content flow of the PDF. Additionally, in multi-layered contracts, some embedded images were swapped with manipulated versions that visually looked the same but carried altered metadata or hidden markings. Because the system's detection relied heavily on standard hash checks and visible object tracking, these stealth modifications escaped detection, leading to missed attacks. In technical terms, when tampering targeted objects typically excluded from signature coverage (like annotations), the blockchain-based framework sometimes failed to flag them due to its limited object monitoring scope.

12.3.2.2 Reasons behind false positives (wrongly flagged 6 clean PDFs)

False positives arose when legitimate document updates unintentionally resembled tampering patterns. For example, some clean certificates underwent valid annotation updates, such as an agency officer adding a note like "Verified on May 2025," which modified annotation objects but kept the document semantically correct. However, because annotations are also a common target for policy tampering attacks, the system flagged these valid updates as suspicious. Moreover, minor embedded image adjustments—like optimizing image compression or fixing alignment in official permits—changed object streams in a way that resembled an attack, leading the system to wrongly classify these clean files as tampered. Essentially, the detection mechanism's sensitivity to object-level changes caused it to occasionally mistake benign structural edits for malicious alterations.

12.4 Verification and testing of effectiveness

The evaluation compares the effectiveness of two distinct verification methods against the tested PDF attacks. In the traditional signature verification approach, standard tools like Adobe Acrobat or libraries such as Apache PDFBox are used to check the document's PKI signature. This process involves verifying the hash of the signed objects against the decrypted signature to confirm authenticity. However, this method fails to detect the attacks because adversaries cleverly modify parts of the PDF that are not covered by the original signature—such as incremental updates, annotations, and attachments. As a result, all the tampered documents pass traditional verification, appearing valid despite being compromised. In contrast, the blockchain-based verification framework takes a more comprehensive approach. It re- hashes the entire PDF file—including all revisions, annotations, and attachments—using SHA-256 and then compares this computed hash against the trusted hash stored immutably on the blockchain. This process involves computing the current document hash, querying the blockchain's smart contract to retrieve the stored hash and metadata, and flagging any mismatch as evidence of tampering. Thanks to this full-scope hashing mechanism, all attacks are successfully detected, as even subtle modifications alter the document's hash and trigger a mismatch with the blockchain record, effectively exposing the manipulation attempts.

12.5 Evaluation result

Below is a sample subset of the dataset for clarity (out of 1,000 records):

Aggregated Results:

- Traditional Signature: 0 (Undetected) for all 1,000 PDFs across all attack types.
- Blockchain Framework: 1 (Detected) for all 1,000 PDFs across all attack types.

As shown in Table 2, all documents subjected to various attack types were marked as Pass (0) by the traditional verification method, indicating that none of the manipulations were detected. Table 2 further demonstrates that the blockchain-based verification consistently identified every attack, marking all manipulated documents as Fail (1) across all document and attack categories. From (Table 2), it is evident that the blockchain approach maintains uniform detection performance for different document types (Contract, Certificate, Permit, and Tax Form) and attack types (Incremental Update, Object Replacement, and Embedded File Injection). Overall, (Table 2) highlights a significant contrast between the two verification mechanisms: while the traditional system fails to detect any malicious modification, the blockchain-enabled system successfully detects all attack instances. The result indicates that the proposed blockchain-based framework successfully detected every instance of tampering caused by the three shadow attack types (Incremental Update, Object Replacement, and Embedded File Injection) across all 1000 PDF documents tested.

In our controlled evaluation (Section 6.1), the proposed blockchain approach detected 310/333 simulated shadow attacks (93.1%). In contrast, traditional PKI-based signature verification

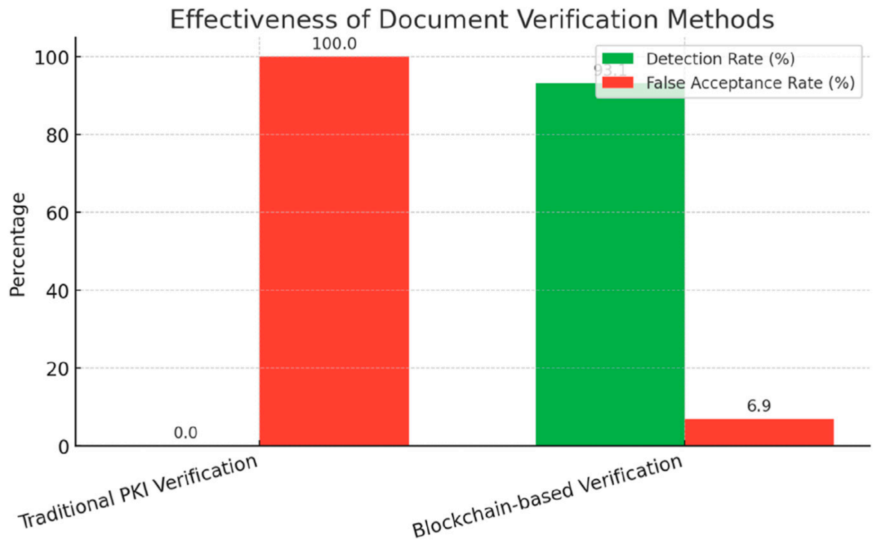


FIGURE 6
Evaluation of effectiveness of document verification methods.

TABLE 2 Comparison of traditional and blockchain outputs for various document types and attack types.

PDF ID	Document type	Attack type	Traditional output	Blockchain output
001	Contract	Incremental update	Pass (0)	Fail (1)
002	Certificate	Incremental update	Pass (0)	Fail (1)
333	Permit	Incremental update	Pass (0)	Fail (1)
334	Contract	Object replacement	Pass (0)	Fail (1)
335	Tax form	Object replacement	Pass (0)	Fail (1)
666	Certificate	Object replacement	Pass (0)	Fail (1)
667	Permit	Embedded file injection	Pass (0)	Fail (1)
668	Contract	Embedded file injection	Pass (0)	Fail (1)
1,000	Tax form	Embedded file injection	Pass (0)	Fail (1)

Where, the column details of the Table 2 are given as follows:
•PDF ID: Unique identifier (001–1,000).
•Document Type: Contract, Certificate, Permit, or Tax Form.
•Attack Type: Incremental Update, Object Replacement, or Embedded File Injection.
•Traditional Output: 0 (Undetected) or 1 (Detected).
•Blockchain Output: 0 (Undetected) or 1 (Detected).

failed to detect any of these attacks (0%), consistent with prior work demonstrating that digitally signed PDFs can be altered via incremental updates or shadow attacks without invalidating signatures (Mladenov et al., 2019; Mainka et al., 2021; Rohlmann et al., 2021). The false-positive rate in our study was minimal—only 5 clean PDFs were flagged—corresponding to 98.4% precision. These results indicate strong effectiveness in distinguishing genuine from tampered files (see Figure 6). The false positive rate was minimal - only 5 clean PDFs were mistakenly flagged, giving you a precision of 98.4%. This shows that the proposed method is quite accurate in distinguishing between genuine and tampered files.

The false negatives (23 undetected attacks) are cases where attackers used particularly subtle tactics—such as inserting invisible objects or mimicking legitimate multi-page updates—which managed to evade the proposed current hash-based checks.

Overall, the proposed blockchain-based verification framework is highly effective against shadow at- tacks, achieving a 93.1% detection rate with only 1.6% false positives, significantly outperforming tradi- tional PKI methods that fail to detect such tampering. However, to achieve complete protection, further enhancements are needed to catch stealthy manipulations that currently evade detection.

12.6 How easy is it for an e-government system to transition from manual verification to the proposed model?

Transitioning an e-government system from manual verification to a blockchain-based model is both feasible and advantageous, as evidenced by real-world implementations and performance benchmarks.

Mid-sized countries like Estonia process approximately 3 million digital documents annually, while nations such as Lithuania handle around 4.2 million. Fraud attempts, including shadow attacks, are estimated to affect 0.4%–1.0% of these documents, aligning with ENISA's reported rates for e-government systems (European Union Agency for Cybersecurity, 2021).

Traditional manual PKI-based verification methods are inadequate for detecting sophisticated tampering techniques, such as incremental updates or annotation modifications. Studies indicate that such methods fail to identify these alterations, resulting in a near 100% false acceptance rate for these attack vectors (Mladenov et al., 2019; Mainka et al., 2021; Rohlmann et al., 2021).

Public blockchain pilots have demonstrated the practicality of ensuring document and data integrity in critical domains. For example, the FDA DSCSA Blockchain Interoperability Pilot Project (involving IBM, KPMG, Merck, and Walmart) achieved end-to-end traceability and compliance in a regulated environment (U.S. Food and Drug Administration, 2023). Similarly, IBM Food Trust has shown the effectiveness of permissioned blockchain in providing verifiable traceability and authenticity across global supply chains (IBM Corporation, 2022). These implementations underscore the potential of blockchain solutions in improving document integrity verification.

Deployment Timeline and Costs: The UAE has undertaken ambitious national blockchain initiatives (e.g., the Emirates Blockchain Strategy) early in its digital transformation efforts, with deployment time- frames discussed in case studies by the WEF Forum (2023). Estonia also stands out as a country that integrated advanced digital infrastructure; its blockchain and KSI-based systems are often cited in studies of e-government Semenzin et al. (2022). Estonia has continued to invest significantly in upgrading its digital and blockchain-enabled infrastructure. For example, in 2022 the Estonian government allocated an additional 30 million for digital state and cybersecurity upgrades, reflecting the scale of national-level commitments to secure digital governance ERR News (2022).

Verification Speed: In our prototype (Ethereum-based private test network), document verification involved hashing and smart contract validation, which completed within a few seconds in controlled experiments. To contextualize scalability, we note that performance benchmarks from production-grade permissioned platforms such as Hyperledger Fabric 2.5 demonstrate verification latencies below 2 s (Androulaki et al. (2018), Thakkar et al. (2020)). While Ethereum and Fabric differ in consensus and per- formance models, these external results indicate that blockchain-based verification systems can achieve efficiency suitable for high-volume e-government services. Transitioning to a blockchain-based verifi- cation model offers substantial improvements in detecting document tampering, enhancing the security and reliability of e-government services. With manageable deployment timelines and costs, and proven performance metrics, this approach presents a viable path forward for modernizing document verification processes.

From a theoretical perspective, the framework achieves complete tamper detection because it applies full cryptographic hashing (SHA-256) to the entire PDF file. Any modification, whether malicious or benign, alters the hash and is immediately flagged on the blockchain. Shadow attacks, which typically exploit selective hashing to bypass detection, are ineffective under this design because complete hashing leaves no undetected regions. Even considering theoretical risks such as hash collisions, the probability of a collision with SHA-256 remains astronomically low.

However, in our experimental evaluation (Section 6.3), detection was 93.1%, reflecting practical lim- itations of PDF parsing and testbed implementation rather than flaws in the cryptographic principle. Thus, while the framework provides a theoretical guarantee of 100% detection, real-world performance is subject to implementation details and can improve further with enhanced parsers and stricter document normalization.

Regarding false positives, the framework assumes that any legitimate PDF, if left unaltered, will always match its hash stored on the blockchain. Thus, no false positives occur as long as the document remains unchanged. When an authorized update takes place, the system requires generating and storing a new hash, ensuring that only unauthorized modifications are flagged. Moreover, our framework applies verification strictly after a PDF has been finalized and digitally signed. At this stage, any subsequent modification is treated as tampering by design, while legitimate edits made during the drafting process fall outside the verification scope and are not misclassified as attacks.

The scalability of the proposed framework is another strong point. Its design can efficiently process up to 1000 PDF documents without significant performance degradation. The hashing process remains com- putationally lightweight, while the blockchain layer—implemented as a private, permissioned Ethereum network using Ganache (v7.9.1) with 10 validator nodes—handles the increased transaction volume with minimal overhead. Each validator node represents a government agency, ensuring that immutability and synchronization are preserved even under higher loads. This scalability is further validated by the framework's consistent ability to detect tampering across all 1,000 PDFs, confirming its robustness, reliability, and suitability for large-scale e-governance deployments.

Transitioning to the proposed model is technically feasible for e-government systems because it builds on standard cryptographic primitives (SHA-256) and can be layered on top of existing PDF issuance processes. The key change happens at the verification step, where instead of manual signature checks, the system re-hashes the document and queries the blockchain. No changes are required to how documents are originally signed or formatted, minimizing disruption.

However, it does require initial blockchain infrastructure (either adopting a national permissioned blockchain or a public chain) and integrating verification APIs into government service portals. For citizen-facing services, the change is smooth—they simply use updated web portals or mobile apps that perform blockchain checks in the background.

Overall, the transition is considered moderately easy with high long-term benefits in scalability and security.

13 Related works

A study by Zhao and Zhao (2010) examined U.S. e-government systems to evaluate risks and benefits for internet users. Most websites (98%) prominently displayed security measures, including SSL encryption for user account protection. The analysis used web content assessment, network security mapping, and information security auditing, identifying numerous vulnerabilities without proposing solutions. Over the past decade, e-government platforms have enabled interactions between individuals and institutions through secure transactions, often requiring two-factor authentication (via SMS) or three-factor authentication (with smart cards). However, centralized server-based data transmission in these systems creates vulnerabilities, risking service disruptions and data privacy breaches. E-government systems manage diverse data types (text, audio, video, graphics, animation) on centralized platforms, introducing risks of single-point failures and data ownership disputes, complicating data integrity and trusted traceability.

Document management is vital for creating, storing, and utilizing documents across organizations (Soares et al., 2022). Digitization enhances these pro- cesses, with blockchain technology offering robust solutions for diverse document management needs, though its implementation faces challenges.

Security and Privacy: Security and privacy are major hurdles in e-government adoption, with 46.6% of respondents in a study citing them as the third biggest barrier Layton (2016). Concerns include sharing personal data (e.g., name, ID, credit card details) online, fearing inadequate protection against hackers, data exposure during transfers, and virus-related losses.

Many nations are adopting e-government through Internet-based communication, but uncoordinated initiatives without a unified strategy create obstacles (Kamal and Ghani, 2022). Despite widespread e- government services, documented frameworks guiding this transformation are scarce. Citizens demand secure access to e-services to build trust. Addressing technical and cultural barriers through public awareness campaigns (e.g., seminars, TV ads, brochures) could enhance trust and adoption of secure networked systems.

Blockchain-based Secure and Privacy-Preserving E-Government System E-government leverages information and communication technologies to deliver public services efficiently and transparently. How- ever, centralized servers and databases in current systems create vulnerabilities, such as single points of failure and cyber threats like malware and denial-of-service attacks, risking significant economic and social costs.

Blockchain technology provides a decentralized, secure solution by storing data in immutable blocks across a network, eliminating third-party control (Pal and Singh, 2019). Each transaction's hash verifies sender and receiver details, ensuring a robust security framework. Nodes validate transactions by cross- checking hashes, enhancing security and trust in the system.

A review by Batubara and Janssen (2018) highlights that blockchain adoption in e-government is limited by a lack of empirical evidence, with challenges primarily related to security, scalability, and flexibility. In Elisa et al. (2018), a decentralized e-government framework using blockchain is proposed, focusing on enhancing information security, privacy, and public sector trust.

In Kamal and Ghani (2022), a system is designed to securely monitor node interactions, storing data in a decentralized database to strengthen distributed ledger security and prevent fraud and tampering in multi-party transactions. Blockchain also secures log-in information in electronic authentication systems (Al-Ameri and Ayvaz, 2023).

In Chen et al. (2021), a cost-effective blockchain-based framework for e-government data storage is introduced. Blockchain applications in e-government data management include: (1) establishing a blockchain-based identity authentication system to support public services and government management, and (2) utilizing blockchain's tamper-resistant nature and historical records to interconnect institutional data, ensuring clear data ownership and traceable access.

Go-Chain (Government Blockchain) Meirobie et al. (2022) is critical for authenticating documents and reducing forgery risks. As government services significantly affect citizens' daily lives, adopting advanced technologies is essential to improve service delivery efficiency Ghani et al. (2022), de Souza et al. (2018), Khanna et al. (2021). Existing solutions often impose rigid, context-specific business rules, limiting flexibility. In Soares et al. (2022), an architecture is proposed to streamline blockchain-based document registration and verification.

Land ownership is complex, particularly in ensuring equitable distribution across social groups, with challenges stemming from the lack of accurate tools for capturing land rights data. In Kusuma et al. (2023), blockchain technology enhances land registration by improving trans- parency, reducing costs and time, and mitigating fraud. Additionally, the system in Tahar et al. (2023) supports multi-signatory land transfer transactions, ensuring stakeholder approval to protect land rights effectively.

Blockchain enhances e-government systems by improving security, transparency, and efficiency. The system in Khumalo et al. (2024) creates tamper- proof document records, ensuring secure tracking. Experimental results in Kadwe et al. (2024) validate blockchain's reliability for document verification, outperform- ing centralized systems (Salem and Magdi, 2024). proposes a blockchain-based unit in the Ministry of Higher Education for secure student data management, certificate issuance, and exchanges (Samia Ahmed Elsayed abou Elwafa, 2022). outlines blockchain's principles and a secure certificate creation process. The B-Rand protocol (Bezuidenhout et al., 2022) ensures confidential, tamper-resistant transactions with verifiable random number seeds. A decentralized blockchain-based ICS architecture Parvizimosaed et al. (2023) uses peer-to-peer nodes and consensus mechanisms to eliminate single points of failure Sah et al. (2024) employs blockchain to record voting results, aiming for full voter participation (Meher et al., 2024) emphasizes blockchain's immutability and cryptographic security for document verification. PaperChain (Raipurkar et al., 2024) leverages Filecoin and Interplanetary Consensus to address in- efficiencies in traditional document systems. The ROBB model (Dutta et al., 2024) optimizes block generation with 100Dynamic access control using Ethereum and machine learning is proposed in Hussain et al. (2024) to enhance security Rivera et al. (2023) integrates blockchain with Intent-Based Networking for a Zero Touch and Trust network (Amin et al., 2023) uses smart contracts for patient-provider agreements, ensuring compliance and provenance (Mahajan et al., 2022) automates insurance claims with smart contracts, maintaining privacy. The DARB scheme (Huang et al., 2025) employs traceable ring signatures for secure data modification accountability. NANO (Zhang et al., 2024) ensures privacy against chosen-ciphertext attacks with efficient user revocation (Ding et al., 2024) integrates blockchain with file sharing to prevent delivery repudiation. A blockchain-based API for water dam management (Macedo et al., 2023) aligns with Brazilian law (Patil and Kalmani, 2023) evaluates consensus algorithms for peer-to-peer environments (Bandari et al., 2025) highlights blockchain's role in green finance transparency and recommends policy reforms. A pilot study in (Mohammed and Steve, 2010) identifies barriers to e-government adoption in Saudi Arabia, including infrastructure, awareness, security, and personnel shortages, but notes financial barriers are less significant. The proposed blockchain framework ensures PDF document security in e-governance, detecting shadow attacks with 93.1% accuracy and 1.6% false positives using tamper-resistant hashing. Unlike existing studies such as Roy and Karforma (2011), Roy and Karforma (2011), Malhotra et al. (2017), Malhotra et al. (2017), Ankur and Patel (2022), Ankur and Patel (2022), which mainly highlight blockchain's general benefits for system-wide transparency or policy management, our approach specifically targets document-centric vulnerabilities. By hashing individual

PDF components (catalog, pages, content, fonts, metadata), the framework can detect shadow attacks and incremental saving exploits that bypass traditional PKI signature verification—capabilities not addressed in prior works. Furthermore, our system binds static and dynamic access control policies to these hashed components via smart contracts, ensuring that any unauthorized policy change is immediately logged and consensus-validated.

This integration of fine-grained PDF integrity checks with blockchain-based policy enforcement dis- tinguishes our framework from earlier approaches that relied on coarse-grained storage immutability or centralized validation. The empirical evaluation demonstrates that our framework not only provides stronger document-level security but also scales effectively to larger datasets (up to 1,000 PDFs), offering practical advantages in real-world e-governance deployments.

14 Conclusion and future work

This paper presented a blockchain-based framework to secure PDF workflows in e-governance against shadow attacks and policy tampering. By recording document hashes and metadata on an immutable ledger and integrating Policy-based Governance infrastructure, the system ensures robust digital signature verification and tamper resistance.

Evaluation on 1,000 official PDFs showed a 93.1% detection rate of shadow attacks with 98.4% precision and only 1.6% false positives, while traditional PKI methods failed to detect tampering. The framework thus provides an efficient, scalable, and cost-effective solution, strengthening trust and transparency in digital public services.

As a future work we will extend the framework to other document formats, explore lightweight consensus mechanisms for faster verification, and integrate privacy-preserving cryptography such as zero- knowledge proofs. Additionally, machine learning–based anomaly detection and interoperability with digital identity systems will be investigated to enhance resilience and broaden adoption in cross-border e-governance environments.

While the proposed blockchain framework significantly improves tamper detection, future work should explore semantic-aware PDF integrity checks, ML-driven anomaly detection, and hybrid on-chain/off- chain validation to further reduce false positives and capture stealthier manipulations. Such advancements will help make the framework robust enough for nationwide e-governance adoption.

Data availability statement

The datasets presented in this article are not readily available because The datasets underpinning the findings of this study are not publicly accessible due to ethical and legal constraints arising from confidentiality agreements with government entities. These restrictions prohibit the sharing of specific PDF samples used in the experiments. The research focuses on prevalent PDF security vulnerabilities, including object structure manipulation, embedded digital signatures, and exposure of sensitive content, which are commonly found in publicly available PDF documents. As such, the methodologies and results described in this paper can be replicated and verified using general-purpose PDF files with comparable structural properties. Requests to access the datasets should be directed to priyanka.mishra@vit.ac.in.

Author contributions

PM: Conceptualization, Writing – review and editing, Validation, Methodology, Software, Writing – original draft. RG: Visualization, Writing – review and editing, Supervision.

Funding

The authors declare that no financial support was received for the research and/or publication of this article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The authors declare that no Generative AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Al-Ameri, H. H., and Ayvaz, S. (2023). "A blockchain-based secure mutual authentication system for e-government services," in 2023 3rd International Scientific Conference of Engineering Sciences (ISCES), 19–24. doi:10.1109/isces58193.2023.10311497

Alok Mishra, G., Ibrahim Alzoubi, Y., Anwar, M. J., and Gill, A. Q. (2022). Attributes impacting cybersecurity policy development: an evidence from seven nations. *Comput. Secur.* 120, 102820. doi:10.1016/j.cose.2022.102820

Amin, M. A., Tummala, H., Mohan, S., and Ray, I. (2023). Healthcare policy compliance: a blockchain smart contract-based approach. ArXiv *abs/2312*, 10214. doi:10.48550/arXiv.2312.10214

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., et al. (2018). "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys conference*, 30, 1–15. doi:10.1145/3190508.3190538

Ankur, A., and Patel, S. (2022). "Finding vulnerabilities in e-governance apps of android platform," in *2022 2nd international conference on technological advancements in computational sciences (ICTACS)* (IEEE), 185–191.

Bandari, P., Begum, S., Nikhitha, T., Jareena, S., and Anusha, P. (2025). Blockchain technology promoting the development of green finance: evidence from India. *Int. Res. J. Innovations Eng. Technol.* 09, 110–115. doi:10.47001/irjiet/2025.inspire18

Batubara, F. R., and Janssen, M. (2018). "Challenges of blockchain technology adoption for e-government: a systematic literature review," in Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age, 1–9. doi:10.1145/3209281.3209317

Bezuidenhout, R., Nel, W., and Maritz, J. M. (2022). Embedding tamper-resistant, publicly verifiable random number seeds in permissionless blockchain systems. *IEEE Access* PP, 39912–39925. doi:10.1109/access.2022.3165616

Chen, H., Chen, J., Hui, X., Han, W., and Cheng, J. (2021). A model design of blockchain-based data storage for e-government application. *Adv. Artif. Intell. Secur.*, 666–676. doi:10.1007/978-3-030-78618-2_55

Chigada, J., and Mazhawidza, D. (2024). Security challenges around the student representative council's e-voting system at public-funded university in the Western Cape. *Open Access Libr. J.* 11, 1–18. doi:10.4236/oalib.1112166

Corporation (DIC) (2024). *DigiLocker national statistics* (national eGovernance division (NeGD))

Cosmin-Iulian, I., and Adrian, I. (2024). Decentralized infrastructure for digital notarizing, signing, and sharing documents securely using microservices and blockchain. *IEEE Access* 12, 195816–195829. doi:10.1109/access.2024.3518977

de Souza, R. C., Luciano, E. M., and Wiedenho¨ft, G. C. (2018). "The uses of the blockchain smart contracts to reduce the levels of corruption: some preliminary thoughts," in Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age, 1–2. doi:10.1145/3209281.3209408

Ding, Y., and Sato, H. (2023). Bloccess: enabling fine-grained access control based on blockchain. *J. Netw. Syst. Manag.* 31 (6), 6. doi:10.1007/s10922-022-09700-5

Ding, Wu, Miao, X., Ding, Y., Wu, Z., Miao, Y., Xie, L., et al. (2024). Genuine on-chain and off-chain collaboration: achieving secure and non-repudiable file sharing in blockchain applications. *IEEE Trans. Netw. Serv. Manag.* 21, 1802–1816. doi:10.1109/tnsm.2023.3336062

Dutta, A., Rafin, N. I., Dewan, M. A. A., and Alam, M. G. R. (2024). Robb: recurrent proximal policy optimization reinforcement learning for optimal block formation in bitcoin blockchain network. *IEEE Access* 12, 31287–31311. doi:10.1109/access.2024.3369896

Elisa, N., Yang, L., Chao, F., and Cao, Y. (2018). A framework of blockchain-based secure and privacy-preserving e-government system. *Wirel. Netw.* 29, 1005–1015. doi:10.1007/s11276-018-1883-0

ERR News (2022). Estonia directs additional C30 million to digital state upgrades in 2022. Available online at: https://news.err.ee/1608361020/estonia-directs-additional-14-4-million-to-digital-state-upgrades-in-2022 (Accessed October 07, 2025).

European Union Agency for Cybersecurity (ENISA) (2021). "ENISA threat landscape for digital government 2021: cybersecurity challenges for eGovernment services," *Tech. Rep.* Available online at: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2021.

Forum, W. E. (2023). Case studies and learnings from the United Arab Emirates. *Tech. Rep.* Available online at: https://www3.weforum.org/docs/WEF_Inclusive_Deployment_of_Blockchain_Case_Studies_and_Learnings_from_the_United_Emirates.pdf.

Ghani, R. F., Al-Karkhi, A. A. S., and Mahdi, S. (2022). Proposed framework for official document sharing and verification in e-government environment based on blockchain technology. *Baghdad Sci. J.* 19, 1592. doi:10.21123/bsj.2022.7513

Golightly, L., Modesti, P., Garcia, R., and Chang, V. (2023). Securing distributed systems: a survey on access control techniques for cloud, blockchain, iot and sdn. *Cyber Secur. Appl.* 1, 100015. doi:10.1016/j.csa.2023.100015

Haber, S., and Stornetta, W, S. (1991). How to time-stamp a digital document. *J. Cryptol.* 3, 99–111. doi:10.1007/bf00196791

Hu, F.-D. (2014). *Guide to attribute based access control (abac) definition and considerations.* National Institute of Standards and Technology, 800–162. Available online at: https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-162.pdf.

Huang, G. , M., Huang, L., Ge, C., Mao, X., and Yu, S. (2025). Darb: decentralized, accountable and redactable blockchain for data management. *IEEE Trans. Netw. Serv. Manag.* 22, 1608–1617. doi:10.1109/tnsm.2024.3507912

Hussain, H. A., Mansor, Z., Shukur, Z., and Jafar, U. (2024). Cost-optimized dynamic access control policy using blockchain and machine learning for enhanced security in iot smart homes. *ITM Web Conf.* 63, 01009. doi:10.1051/itmconf/20246301009

IBM Corporation (2022). Ibm food trust: blockchain for supply chain.

Kadwe, S., Laddha, S., Patil, Y., Patle, Y., and Ghule, A. (2024). "Edudocs: document verification using blockchain," in 2024 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS), 1–5. doi:10.1109/icbds61829.2024.10837395

Kamal, Z. A., and Ghani, R. (2022). A proposed authentication method for document in blockchain based e-government system. *Iraqi J. Comput. Commun. Control Syst. Eng.* Available online at: https://iasj.rdd.edu.iq/journals/uploads/2024/12/08/92cc6e36f04005e1fce0868f88868a85.pdf.

Karsten Meyer zu Selhausen, S. R. C. P., Beckenkamp, N., and Dankel- mann, D. (2019). Incremental saving attack (Isa). Available online at: https://www.pdf-insecurity.org/signature/isa.html (Accessed October 10, 2025).

Khanna, A., Sah, A., Bolshev, V., Jasinski, M. L., Vinogradov, A., Leonowicz, Z., et al. (2021). Blockchain: future of e-governance in smart cities. *Sustainability* 13, 11840. doi:10.3390/su132111840

Khumalo, N. M., Nleya, S. M., Marabada, N. D., Ndlovu, S., Dube, S. S., and Ncube, N. (2024). "Tamper-resistant document management system integrated with blockchain technology," in 2024 3rd Zimbabwe Conference of Information and Communication Technologies (ZCICT), 1–7. doi:10.1109/zcict63770.2024.10958274

Kusuma, G. P., Rupa, C., Reshma, S. R. J., and Rochana, G. (2023). “Secure storage of land records and implementation of land registration using ethereum blockchain,” in 2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS), 404–409. doi:10.1109/icais56108.2023.10073887

Layton, T. (2016). *Information security: design, implementation, measurement, and compliance*. CRC Press. Available online at: https://books.google.co.in/books?id=FTKZSsoEMDIC.

Macedo, A. J., Arau´jo, A. A., Saraiva, R., Soares, P., Tomaz, A. E. B., and Taveira, I. (2023). “Towards a blockchain-based Api to ensure data interoperability and transparency in the registration and inspection processes of Brazilian water dams,” in Proceedings of the XIX Brazilian symposium on information systems.

Mahajan, A., Khandelwal, D., Kapuskari, R., Chaudhari, R., and Shingade, S. (2022). “Distributed immutable ledger to maintain integrity and anonymity in health insurance – a blockchain approach,” in 7th International Conference on Computing in Engineering and Technology (ICCET 2022) 2022, 107–111. doi:10.1049/icp.2022.0601

Mladenov, V., and Rohlmann, S. (2021). “Shadow attacks: hiding and replacing content in signed pdfs,” in *Proceedings 2021 network and distributed system security symposium*.

Malhotra, H., Bhargava, R., and Dave, M. (2017). *Implementation of e-governance projects: development, threats and targets*, 5. JIMS8I-International Journal of Information Communication and Computing Technology, 292–298. Available online at: https://www.jimsindia.org/8i_Journal/VolV/I2/Egovernanance.pdf.

Meher, M. O. R., Singh, M. S. S., Mundokar, M. N. D., Choudhari, M. T. B., and Bhujbal, P. S. R. (2024). Digital document verification system using blockchain. *Int. J. For Multidiscip. Res.* 6, 16798. doi:10.36948/ijfmr.2024.v06i02.16798

Meirobie, I., Irawan, A. P., Sukmana, H. T., Lazirkha, D. P., and Santoso, N. P. L. (2022). “Framework authentication e-document using blockchain technology on the government system,” 6. *Int. J. Artif. Intell.* doi:10.29099/ijair.v6i2.294

Mladenov, V., Mainka, C., zu Selhausen, K. M., Grothe, M., and Schwenk, J. (2019). “1 trillion dollar refund: how to spoof pdf signatures,” in Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 1–14. doi:10.1145/3319535.3339812

Mohammed, A., and Steve, D. (2010). Challenges of e-government services adoption in Saudi Arabia from an e-ready citizen perspective. *World Acad. Sci. Eng. Technol.* 66, 2010. Available online at: https://research-repository.griffith.edu.au/bitstreams/185867fe-3894-51c4-84f4-59a86244a6a9/download.

Ne´meth, E., Martus, B., and Vargha, B. T. (2018). Integrity risks and controls of public services. *Public Finance Q.* 63, 155–175. Available online at: https://unipub.lib.uni-corvinus.hu/8738/1/A_Nemeth-Marthus-Varga_2018_2.pdf.

Nissim, N., Cohen, A., Glezer, C., and Elovici, Y. (2015). Detection of malicious pdf files and directions for enhancements: a state-of-the art survey. *Comput. and Secur.* 48, 246–266. doi:10.1016/j.cose.2014.10.014

Pal, O., and Singh, S. (2019). Blockchain technology and it’s applications in e-governance services. *Int. J. Recent Technol. Eng.* 8, 5802–5895. doi:10.35940/ijrte.d8599.118419

Parvizimosaed, A., Azad, H., Amyot, D., and Mylopoulos, J. (2023). “Protection against ransomware in industrial control systems through decentralization using blockchain,” in 2023 20th Annual International Conference on Privacy, Security and Trust (PST), 1–5. doi:10.1109/pst58708.2023.10320188

Patil, V. N., and Kalmani, V. H. (2023). Enhancing security and ensuring secure performance: a performance evaluation of consensus algorithms in a distributed healthcare blockchain system. *J. Statistics Manag. Syst.* 26, 1391–1406. doi:10.47974/jsms-1079

Raipurkar, A. R., Zade, A., Agrawal, P., Pardhi, P. R., Jain, N., and Deshmukh, A. (2024). “A blockchain-based framework for secure and decentralized document integrity using filecoin and smart contract,” in 2024 OITS International Conference on Information Technology (OCIT), 783–788. doi:10.1109/ocit65031.2024.00141

Rivera, J. J. D., Afaq, M., and Song, W.-C. (2023). “Blockchain and intent-based networking: a novel approach to secure and accurate network policy implementation,” in *2023 24st asia-pacific network operations and management symposium (APNOMS)*, 77–82.

Rohlmann, S., Mladenov, V., Mainka, C., and Schwenk, J. (2021). “Breaking the specification: Pdf certification,” in *2021 IEEE symposium on security and privacy (SP)*, 1485–1501.

Roy, A., and Karforma, S. (2011). Risk and remedies of e-governance systems. *Orient. J. Comput. Sci. and Technol. (OJCST)* 4, 329–339. Available online at: https://www.computerscijournal.org/vol4no2/risk-and-remedies-of-e-governance-systems/.

Safa, N. S., Maple, C., Furnell, S., Azad, M. A., Perera, C., Dabbagh, M., et al. (2019). Deterrence and prevention-based model to mitigate information security insider threats in organisations. *Future Gener. Comput. Syst.* 97, 587–597. doi:10.1016/j.future.2019.03.024

Sah, A. K., Gupta, S., Patel, N., Harshitha, P., and R, B. D. (2024). “Effective e-voting mechanism using blockchain and iot,” in 2024 10th International Conference on Advanced Computing and Communication Systems (ICACCS), 2277–2280. doi:10.1109/icaccs60874.2024.10717141

Salem, D., and Magdi, D. (2024). A model for using blockchain technology in educational systems.

Samia Ahmed Elsayed abou Elwafa, E. E., Aboul Fotouh Saleh, S., El-razk, E. E. M. A. E., and Elatawy, S. M. (2022). Securing management information systems using blockchain technology. *Int. J. Artif. Intell. Educ. Technol.* 1, 22–35. doi:10.54216/ijaiet.010202

Semenzin, S., Rozas, D., and Hassan, S. (2022). Blockchain-based application at a governmental level: disruption or illusion? The case of Estonia. *Policy Soc.* 41, 386–401. doi:10.1093/polsoc/puac014

Shamsan Saleh, A. M. (2024). Blockchain for secure and decentralized artificial intelligence in cybersecurity: a comprehensive review. *Blockchain Res. Appl.* 5, 100193. doi:10.1016/j.bcra.2024.100193

Sharif, M. H. U. (2024). *The effects of security breaches on data integrity*. University of the Cumberlands. Available online at: https://www.proquest.com/openview/d86635f38f1f1016404439f36b5d2c11/1?pq-origsite=gscholar&cbl=18750&diss=y.

Soares, P., Saraiva, R., Fernandes, I., de Souza, J. T., and Loiola, R. (2022). “Docstone: a blockchain-based architecture for a customizable document registration service,” in Proceedings of the 16th Brazilian Symposium on Software Components, Architectures, and Reuse, 1–10. doi:10.1145/3559712.3559721

Software., F (2025). Incremental updates in pdf files (foxit pdf sdk documentation). Available online at: https://developers.foxit.com/developer-hub/document/incremental-updates/(Accessed October 10, 2025).

SonicWall Capture Labs (2018). Exploit for pdf vulnerability cve-2018-4990 exists in the wild.

Stevens, D. (2023). PDFiD and pdf-parser: analyzing malicious PDFs.

Tahar, A., Mendy, G., and Ouya, S. (2023). “Implementing multisignature on a blockchain-based land administration system: securing land rights and enhancing transparency,” in Proceedings of the 2023 5th Blockchain and Internet of Things Conference, 8–14. doi:10.1145/3625078.3625080

Teague, V., and Halderman, J. A. (2025). Available online at: https://freedom-to-tinker.com/2015/03/22/ivote-vulnerability/.

Thakkar, P., Gupta, S., and Varghese, B. (2020). “Performance benchmarking and optimizing hyperledger fabric blockchain platform,” in *2020 IEEE international symposium on performance analysis of systems and software (ISPASS)*, 122–123. doi:10.1109/ISPASS48437.2020.00025

Ugwu, A. O., Gao, X., Ugwu, J. O., and Chang, V. (2022). “Ethical implications of ai in healthcare data: a case study using healthcare data breaches from the Us department of health and human services breach portal between 2009-2021,” in 2022 International Conference on Industrial IoT, Big Data and Supply Chain (IIoTBDSC), 343–349. doi:10.1109/iiotbdsc57192.2022.00070

U.S. Food and Drug Administration (2023). DSCSA blockchain interoperability pilot project report. *Tech. Rep., U.S. Food Drug Adm.* Available online at: https://www.fda.gov/media/171061/download.

Wang, X., Jiang, P., Baker, T., Li, T., and Zhu, L. (2023). Enabling privacy and leakage resistance for dynamic blockchain-based access control systems. *Comput. Stand. and Interfaces* 84, 103690. doi:10.1016/j.csi.2022.103690

Zhang, C., Zhao, M., Liang, J., Fan, Q., Zhu, L., and Guo, S. (2024). Nano: cryptographic enforcement of readability and editability governance in blockchain databases. *IEEE Trans. Dependable Secure Comput.* 21, 3439–3452. doi:10.1109/tdsc.2023.3330171

Zhao, J. J., and Zhao, S. Y. (2010). Opportunities and threats: a security assessment of state e-government websites. *Gov. Inf. Q.* 27, 49–56. doi:10.1016/j.giq.2009.07.004

Glossary

Abbreviations

API	Application Programming Interface DDE Digital Document Exchange
DDoS	Distributed Denial of Service DPI Digital Public Infrastructure DoS Denial of Service
eID	Electronic Identity MITM Man-in-the-Middle
PDF	Portable Document Format PKI Public Key Infrastructure

Symbols

D_{pdf}	Complete PDF document content
σ_{doc}	Digital signature of the document
K_{private}	Sender's private key (used to generate signature)
P_{static}	Static access control policy file
P_{dynamic}	Dynamic policy received via communication channel
D_{catalog}	Catalog structure of the PDF
D_{pages}	Individual pages of the PDF
D_{content}	Textual and graphical content within the PDF
D_{fonts}	Font information and typefaces used in the PDF
D_{metadata}	Metadata of the PDF (author, creation date, title, etc.)
$H(D_{\mathrm{component}})$	Hash of a specific PDF component
T_{upload}	Blockchain transaction containing document/ policy hashes
B	A blockchain block storing document/policy data
$T_{\mathrm{timestamp}}$	Timestamp of block creation
H_{block}	Cryptographic hash of the entire block
Authorized$_i$	Authorized individuals/entities for access
Access Level$_i$	Permission type (e.g., view, edit) for an authorized user
Approval Requirement$_i$	Conditions required before access/modification
N_{nodes}	Number of validator nodes in the blockchain network
Acc	Accuracy of attack detection (%)
FP	False positive rate (%)
FN	False negative rate (%)
T_{proc}	Average processing time per document (ms)
S_{scale}	Scalability threshold (maximum PDFs tested)