Journal of Information Security and Applications 63 (2021) 103025

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

NyaYa: Blockchain-based electronic law record management scheme for judicial investigations

Ashwin Verma, Pronaya Bhattacharya, Deepti Saraswat, Sudeep Tanwar *

Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad, India

ARTICLE INFO

Keywords:
Blockchain
Electronic law records
Governance
Judicial investigations
Smart contracts

ABSTRACT

In digitization, judicial investigations have transitioned towards digital storage of forensic shreds of evidence as electronic law records (ELRs). The shift poses varied challenges of ELR preservation, homogeneity of case formats, chronology in recorded statements by suspects, time-stamping and digital signatures on ELRs, and the chain of transfer of cases to different law enforcement agencies (LEAs) over open channels. Thus, privacy and trust among judicial stakeholders-case appellant (CA), case defendant (CD), the police officer (PO), defence lawyer (DL), prosecutor lawyers (PL), LEAs, and court judge is a prime concern. Motivated from the aforementioned discussions, the paper presents a blockchain (BC)-based ELR management scheme, *NyaYa*, that operates as a four-phased scheme of judicial stakeholder registration in BC, case registration with meta-hash keys in public BC, that reference an external off-chain interplanetary file storage (IPFS), chronology of investigative updates among LEAs, and case hearing and settlement through smart contracts (SCs). In the simulation, *NyaYa* is compared to traditional ELR storage schemes for parameters like mining cost, query fetching time, block processing time, obtained IPFS throughput, signing latency, the effect of collusion attacks, ELR processing time, and corrupted indexes in IPFS. We also presented formal verification and proposed functionalities of SCs. In simulation, at 6 USD mining cost, *NyaYa* can append 22456 transactions, compared to 21497 and 3000 transactions respectively in existing schemes. The achieved query fetching time is 0.852 milli-seconds (ms), at 25 blocks, with cache support of 32 kilobyte (KB). The scheme has an average signing latency of 622.95 ms, and achieves a high-trust probability of 0.887 %, compared to 0.765 % in consortium BC, and 0.455 % in private BC, at 500, colluding nodes. An improvement of 6.77 % is achieved in ELR uploading latency, and the scheme has only 21 corrupted IPFS indexes for 350 fetched ELRs. The obtained results indicate the efficacy of the proposed scheme against conventional schemes.

1. Introduction

With the advent of digitization, the paradigm has shifted from physical papers to e-documents. In the judicial system, the law and governance have also adapted themselves to the change in filing records. The version, named electronic law records (ELRs), includes the recorded statements of different persons, and cases accused, and prime suspects based on legal trials. The legal trials, based on evidence, filed the first investigation report (FIR) by CA, form a chronology of investigative events, and records the updates on ELRs. However, due to strict policies, norms, and reduced manpower in investigative agencies appointed by the government, many cases are pending either for investigation or in fair trials. As per statistics by *National Judicial Data Grid (NJDC)* of India [1], by 2030, it is expected that overall pending cases (including all courts and talukas) would rise to 56.77%, for fair investigative trials, or case hearings. Fig. 1(a) presents the details of the same. The predicted statistics reflect the inherent load on the judicial sector and induced centralization of proceedings which adds up the processing latency, even in the presence of ELRs.

As mentioned, ELRs have induced faster file movements of case files, sequenced through case numbers, among law stakeholders like petitioner (CA), accused (CD), PO, lawyers (PL/DL), different LEAs, and courts. In addition, ELRs also save space over traditional paperwork, manual storage hassles, and thus increases staff efficiency. It improves the public access of records, that allows real-time updates during the investigative cycle by federal agencies. However, the ELR records in digital form are stored in centralized government servers and are thus prone to high-end user latency of querying cases (through case numbers), due to the bulk of stored cases in digital form. Moreover, centralized servers are prone to single-point failures, and also are

* Corresponding author.
E-mail addresses: ashwin.verma@nirmauni.ac.in (A. Verma), pronoya.bhattacharya@nirmauni.ac.in (P. Bhattacharya), deepti.saraswat@nirmauni.ac.in (D. Saraswat), sudeep.tanwar@nirmauni.ac.in (S. Tanwar).

https://doi.org/10.1016/j.jisa.2021.103025

Available online 26 October 2021

(a) Expected rise in pending trials and law cases by 2030 [1]

(b) Challenges faced by current law investigation agencies [2]

(c) Global Blockchain adoption in Law enforcement markets [3]

Fig. 1. Current challenges faced by law markets and potential blockchain based adoptions.

Table 1
Abbreviations and their meanings.

Abbreviations	Meaning
CA	Case Appellant
CD	Case Defendant
DL	Defence Lawyer
ELR	Electronic Law Records
FIR	First Investigation Report
IPFS	Interplanetary File System
LEA	Law Enforcement Agency
PL	Prosecution Lawyer
PO	Police Officer

Table 2
Notation and their meanings.

Notations	Meaning	Notations	Meaning
e_{lea}	LEA	Ds_{tx_k}	Digital signature on ith e_{la}
e_{ca}	CA	B_k	Block representation
e_{cd}	CD	W_e	Wallet entity
e_{po}	PO	A_e	Account detail of e
e_{dl}	DL	K_e	Public QR
e_{pl}	PL	S_e	Signature of e
e_{cj}	Chief Justice	T, t'	Timestamp
e_{la}	Law Admin	M_r	Merkle root
e	Entity set	R_e	Registration of e
C_{ELR}	Comprehensive ELR	DS_{po}	Digital sign of e_{po}
M_{cno_k}	Metadata on kth case	PK_{po}	Private key of e_{po}
H_k	Hash function on kth data	U_k^i	Update of kth data for ith e_{la}
K_{tx_k}	Hashed key	μ	Cache size

honeypots for security attacks by malicious adversaries like denial-of-service, alteration, and impersonation of public identity. As the ELRs are highly sensitive and confidential, centralized storage is not a viable solution. Moreover, the former is faced with inherent challenges of ambiguities of draft versions, authorization by digital timestamps, and format inconsistencies. Once CA files the case, the records are timestamped on centralized government servers that are prone to privacy and security attack vectors, as aforementioned. Also, there might be collusion among multiple parties, which might change the investigative discourse through ELR tampering. Thus, forensics requires a complete revamp and distributed ELRs servers are designed to mitigate the attack vectors and eliminate the end-user query latency. However, the issue of collusion attacks is yet not addressed that might result in the manipulation of the outcome of the investigation due to enormous data exchange between cloud, user and Internet-of-Things (IoT) devices [2].

Thus, ELRs suffer from authorized transparency, chronology, and trust issues of a sequence of investigative events, making the court hearings be manipulated against an honest identity. Lenk et al. [3] presented the challenges as case-study of foreign investments and settlements for European Union (EU). The details are presented in Fig. 1(b).

Thus, in law investigations and judicial proceedings, blockchain (BC) can be a potential solution provider to leverage immutably versioned controlling of ELRs, and ensure trust among judicial stakeholders [4]. BC also eliminates the issues of existing distributed storage ecosystems and addresses the vulnerabilities like data loss, disk failures, and trust against adversarial attacks. The capability to secure transactions authorized access, decentralization and immutability through blockchain ensures legitimacy and integrity of digital evidence for admissibility of the latter in the court of law. In judicial investigations, a public BC is a more preferred approach, so that it ensures maximum decentralization, trust, anonymity, and security. Public chains, owing to the large chain size, allows a collusion-free network as dishonest miners require a large amount of computational power to control more than 50% of the network. In Proof-of-Work (PoW) sense, this means that miners have to solve the difficulty problem, and have to modify block nonces in a short span, before a new correct block is ordered, to ensure validation in the network. Thus, public BC allows fairness in block ordering, compared to private and consortium chains, which can be controlled through collusion among a set of registered nodes A public chain will induce maximum security, anonymity, transparency and a corruption-free network with the freedom to maximize framework throughput.

As per the study of BC-adoption in legal issues [5], the use cases of BC-in law infrastructure is presented in Fig. 1(c). However, BC leverages trust and chronology to ELRs, but public chains are limited by storage limits. With the bulk of ELRs, storing data on a public chain is not scalable, and introduces high latency, and lowers the transactional throughput. For the same, authors have mentioned approaches of resilient network infrastructures [6], or storage as off-chain interplanetary file system (IPFS) storage [7]. Only meta-information can be stored in on-chain, with indexed reference to IPFS to maintain scalability. IPFS allows the exchange of large data using peer-to-peer communication and fast file searching using Merkle direct acyclic graph (DAG) approach [8]. This reduces the overall load, and more transactions can be processed in the BC network. Table 1 presents the list of abbreviations used in the paper and Table 2 presents the list of notations.

1.1. Motivation

Motivated from the aforementioned discussions, in this paper, we present a BC-based law-investigation scheme, *NyaYa*, that uses a public BC with IPFS off-chain storage to maintain and manage the ELRs. The

scheme addresses the gaps of earlier approaches by addressing an end-to-end transparent and chronological access system that allows all registered stakeholders to view and access data. The meta-information of ELRs is only stored in the public on-chain, and the record is referenced through case number, which is the hash object of the corresponding record in IPFS. All the users can view the public chain, the public chain, but only registered stakeholders, can access the IPFS record through the IPFS hash object, and private IPFS key, once they are registered and authorized by federal stakeholders. With the combination of the IPFS hash object, and IPFS private key, a registered user can access the case records, thereby maintaining the user privacy, and integrity of law records. Through proper authorization keys and access-control policies, stakeholders like PO and LEAs can update the ELR records to indicate new investigative findings. Thus, *NyaYa* ensures transparency and security, with data redundancy and disk failures eliminated through IPFS. In case of case closure and after case decision by the court of law, the evicted can be penalized for financial settlements through seamless SCs between CA and CD.

1.2. Contributions

- A BC-based ELR management and access scheme to ensure transparency and chronology in law investigations.
- A proposed FIR registration algorithm is proposed for CA that is digitally time-stamped by PO and meta-information is stored in BC, with case details stored in IPFS as off-chain, accessed through an indexed hash key and private credentials of authorized stakeholders. This improves the scalability of on-chain operations.
- A record-fetching algorithm is proposed for ELRs, starting from genesis block in BC, with updates in law investigations by LEAs managed through meta-information and reflected in IPFS. Also, SCs are proposed that ensure seamless payments by accused parties to the aggrieved party.

1.3. Article structure

The paper is organized into five sections. Section 2 presents the existing state-of-the-art schemes. Section 3 discusses the BC-based sign-encryption scheme for secure data exchange at gateway nodes. Section 4 presents the performance evaluation of the scheme against existing conventional approaches. Finally, Section 5 concludes the paper.

2. Key terminologies and related work

The section presents the discussion on the key terminologies and discusses the comparative analysis of the state-of-the-art schemes.

2.1. Key terminologies

The section presents a discussion on the basics of BC, Mining and consensus schemes, IPFS, and SC. The details are now presented as follows.

2.1.1. Blockchain

BC is a distributed ledger technology that maintains series of events or transactions chronologically. Fig. 2 presents the key entities and benefits of the BC-based ledgers. The added blocks are hashed through standard cryptographic hash primitives, making the ledger immutable and transparent for all authorized stakeholders. BC has found specific use-cases in finance, healthcare, cloud, and edge infrastructure security, tourism, and education sector [9]. As the entities in BC interact with the shared copy of the distributed ledger, there are no third-party intermediaries, and thus BC leverages trust in the overall ecosystem.

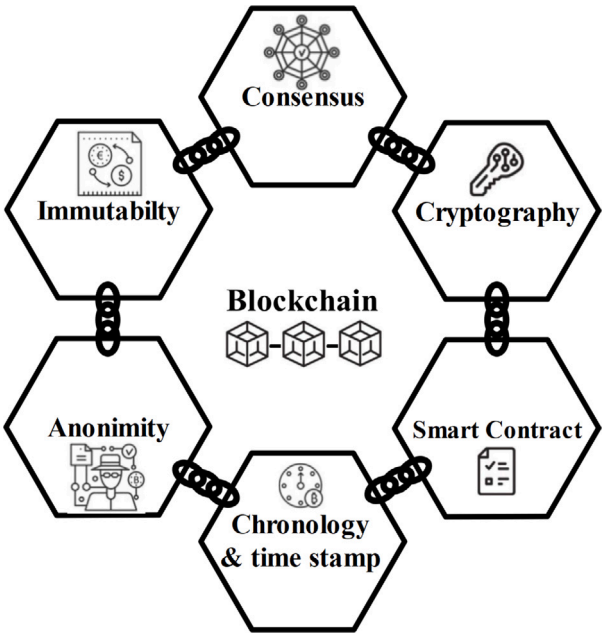


Fig. 2. Schematics of the BC Network.

An adversary in the shared network cannot manipulate the transactions recorded in blocks, as the current block hash is linked to the previous block hash. Thus any alteration would make the entire chain invalid. In such cases, the cryptographic hash of all the subsequent blocks would change, and the manipulation would be identified, and the source of change would be chronologically identified.

2.1.2. Mining and consensus protocols

In BC, miner entities validate the added transactions in blocks. To add the block, the miners must solve a difficult problem with the output random nonce value lower than the acceptable target hash. All the miner entities are in contention to solve the difficult problem. The miner entity capable of solving the problem first gets the chance to add the block and is rewarded through incentives. To solve the difficulty problem, miners require high computational power and storage to run the cryptographic algorithm. Once the block is added to the BC network, other network nodes verify the newly added block through a known consensus protocol. Through consensus, all the distributed peers in BC reach a common agreement about the chain ledger's current state, and thus it achieves trust in the ecosystem. The consensus mechanism guarantees that the newly added block is validated, and the chain state is global and the recent version is consistent across all nodes. Different types of consensus protocols exist like proof-of-work, proof-of-stake, proof-of-authority, and many others. Bodkhe et al. [10] presented a detailed discussion of the different consensus protocols along with the specific use cases and applications. The choice of a given consensus protocol depends explicitly on the network requirements, specific to the bandwidth and end-latency requirements, and the amount of achievable decentralization by the concerned application.

2.1.3. Interplanetary file systems

Centralized databases are faced with inherent challenges of single-point storage, high bandwidth, query-fetch latency, and security and privacy-based attack vectors by a malicious entity. Thus, to address these challenges, IPFS provides a distributed and consistent ledger to store and share data that leverages easy access and replications. To achieve the same, IPFS provides content-based addressing as compared to address-based addressing by the conventional databases. Through content-based addressing, location transparency is maintained, saving

Table 3
Comparison of different distributed data storage system.

Operation life cycle	IPFS	Sia	Filebase	Storj	Arweave
Data creation	No impediment	No impediment	No impediment	No impediment	No impediment
Data modification	Different versions created on updation that can be managed by file object pointer	No indication for version control	No indication for version control	No indication for version control	Separate file need to add for tracking different version
Metadata classification	Metadata for pointers only	No indication to add metadata	User defined metadata can be added	Metadata for pointers of file	No clear indication to add metadata
Storage security	Encryption on file	Encryption on file	Encryption on file	Encryption on file	Encryption on file
Data authenticity	Unique identifier confirms it	Unique identifier confirms it	Allows content to be private	Unique identifier confirms it	Unique identifier confirms it
Data retrieval	With unique hash	Files can be shared and downloaded	Unique URL as well share the file	User credential to share and access	By sharing the link of resources
Data retention	Console allows deletion	Console allows deletion	Console allows deletion	Can delete the file	Permanently stored

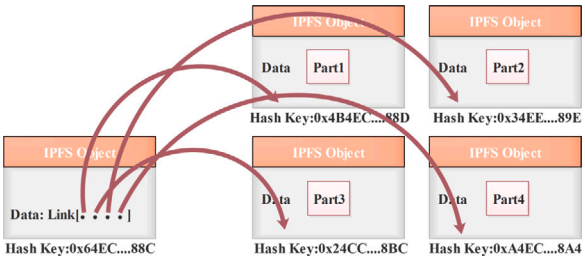


Fig. 3. Referencing of file objects in IPFS.

the sharing servers from denial-of-service and hijacking attacks. IPFS also reduces data redundancy as multiple users store the same data or file on the network, which would be considered the same resource for all users with the shared key. Files or data are stored in the IPFS as an object, and the IPFS object can store up to 256 KB of data. In case the size of data exceeds the defined limit, an IPFS linkage is created and the addresses of other IPFS object is stored through the pointer links. Fig. 3 presents the linkage structure of an IPFS file object. In multimedia streaming applications, audio and video contents are split into multiple frames, and frames are collected and added to an IPFS object. Once the frames' data size exceeds 256 KB, a reference pointer is set up to an empty IPFS object node, and the frames are stored in that node.

In essence, IPFS combines multiple technologies such as distributed hash tables (DHT), block exchange systems, and version control systems. Thus, IPFS objects can be combined with BC, where the IPFS hash can be stored as an external reference in blocks to fetch the data records. Moreover, BC leveraged with IPFS to address data storage challenges in cloud computing platforms, combined with advantages of low latency, improved data privacy, and accessibility.

Transactions on BC consists of files, audio, video and pictures, and with the rise of BC and distributed ledger technology there is a need to address what to store on-chain and off-chain. Data might contain some personal information that needs to keep safe and secure. Moreover, it is also important to check whether the information is disposed of after a specific retention time. The main challenges of BC include retention and disposition of data from the chain as the system is immutable, so we need a distributed system to store important information securely. Table 3 provides the comparison of IPFS with other distributed storage systems.

2.1.4. Smart contract

SC can be considered self-executing code logic executed between transacting entities based on predefined rules of agreements or events. As SCs only execute when the predefined criteria are met, they cannot be tampered with by an adversary.

Once a contract is executed, the transaction is validated and added to BC, visible to every node in the network. Thus, SC eliminates the requirements of third-part settlements, like in the case of financial ecosystems. SC allows users to effectively manage their assets and define the specified rights for their assets. Thus, it provides a public and verifiable logic, with easy maintenance and governance in a peer-to-peer network. In applications, such as supply-chain ecosystems, SC leverages automated payments among each transacting entity in the chain once the goods are transferred from one point to another. Thus, it forms a decentralized and automated ecosystem. SC can be deployed through Ethereum based platforms and is written in programming languages like C++, JavaScript, Solidity, and Rholang. In the case of permissioned BC, the contracts are executed in isolated containers or dockers to prevent public access and are referred to as chain codes. To execute an SC, a certain amount of transaction cost, or gas, is required. The users are charged for the gas according to the current gas price for each unit. SC is Turing complete and allows verifiable programming logic with no illegal branch and looping conditions. Thus, Turing complete SC has no movable parts and can be executed directly through decentralized platforms, or *DApps*, without the requirement of a separate deployment.

2.2. Related work

This section presents the existing state-of-the-art schemes proposed by the authors. Chopade et al. [15] proposed a BC framework to store the forensic shreds of evidence. The scheme uses Base64 encryption techniques to transfer the evidence securely from one participant to another. However, there is a series of evidence findings in the on-chain process, which unnecessarily increases the cost of mining operations. Hossain et al. [11] proposed *Probe-IoT*, a framework for IoT based system that collects information from the different entities in the network (IoT devices, clouds, users) and securely adds to BC. The entities use encryption to validate signatures between different processing entities. The approach eliminates single entity control of information in the system. Billard et al. [12] provides the details of the framework to store e-evidences in a digital inventory for ease of identification of the chain of records in a forensic investigation. The author utilizes three data structures that comprise BC, forensic rating and time-tagging events of the e-evidence. Hardwick et al. [13] describes BC and smart contract-based tender bidding process that will enhance transparency and fairness in all government tenders. The authors use ethereum BC application programming interface (API) and Truffle JS decentralized environment for computational performance and gas cost.

Table 4
Comparative table for existing approaches used in forensic investigation using Blockchain.

Authors	Year	Objective	1	2	3	4	5	Pros	Cons
Hossain et al. [11]	2018	IoT-based framework with public ledger for factual data in crime investigations	Y	N	N	N	N	A BC-based provenance scheme to store peer evidence transactions by IoT devices	Analysis and computational efficiency for energy and storage requirements is not mentioned.
Billard et al. [12]	2018	An evidence-based confidence rating scheme with forensic weights in BC	Y	N	N	N	N	Through weighted heuristics, able to check validity of the evidence	Does not provide the precise confidence rating for the evidence.
Hardwick et al. [13]	2018	Transparent and Fair technique for government e-tendering process on BC	Y	N	Y	N	N	Through BC, the e-tender becomes fully autonomous, and payment automation is carried through SC	Multi-party stakeholders consensus is not considered.
Hossain et al. [2]	2018	IoT-based framework for forensic investigation on public ledgers	Y	N	Y	Y	N	Through decentralized control, redundancy and resilience are added to failure storage points, increasing the framework availability	The authors have considered case-study with limited comparisons, more comprehensive evaluation is not presented.
Brotsis et al. [14]	2019	A Blockchain base framework for preservation of Forensic Evidence in IoT Environments	Y	N	N	Y	N	BC-based smart home evidence collection framework with data maintained in IPFS as off-chain that improves the scalability of chain transactions	Post evidence collection, consensus mechanisms are not proposed.
Chopade et al. [15]	2019	A BC-based framework based for forensic evidences	Y	N	N	N	N	Transparency and immutability of investigations	Data stored as on-chain, which is not scalable due to limited memory.
Liu et al. [16]	2020	A BC-based decentralized service computing paradigm for data governance life cycle	Y	N	N	N	Y	Service data are functionally application-independent to allow dynamic operations and is owner controlled	Simulation and evidence based results needs to be presented.
Li et al. [17]	2020	A BC-based privacy-preserving witness evidence collection scheme for judicial investigations	Y	Y	Y	N	Y	Short and efficient randomized signature algorithms are proposed for authorization, with voting among jury members to achieve case decision consensus	The model assumes a proof-of-stake consensus, so it not scalable with growing legal networks.
Zhang et al. [18]	2020	A light-weight consensus protocol for industrial IoT to ensure secure data transmission	N	N	Y	N	N	Two path routing strategy is used in the transmission to achieve data consistency and also ensure the safety and reliability of the data.	High routing and signing overheads.
Dorii et al. [19]	2020	A lightweight BC-based scalable scheme for security and anonymity in IoT	Y	N	Y	N	Y	Scheme provide end-to-end security and is highly optimized for IoT requirement	The algorithm assumes a distributed transaction throughput, that induces processing latency at end applications.
Kumar et al. [20]	2021	An IoT and BC based framework for evidence gathering and communications & digital forensic evidence management	Y	N	N	N	Y	Scheme is advantageous is terms of complexity, gas and energy consumption as well as resource utilization	The framework assumes the privacy aspects are preserved inherently.
Wang et al. [21]	2021	A privacy protection scheme for telemedicine diagnosis using double BC for remote medical diagnosis and data security	Y	Y	Y	N	Y	Access control scheme is optimum in terms of interaction, communication cost and throughput	The scheme assumes constant data retrieval and does not focus on energy, storage, privacy requirements.
Awuson-David et al. [22]	2021	A BC-based framework to forensically maintain integrity of log evidence in cloud	Y	N	Y	N	N	Advantageous in terms of trustworthiness, easy log retrieval from cloud, geolocations, time-stamping	Implementation on virtual platform and unavailability of global standard BC framework.

1 - Transparency, 2 - Off-chain (IPFS), 3 - Security Analysis, 4 - Chronology, 5 - Signing Latency, Y shows parameter is considered, N shows parameter is not considered.

Hossain et al. [2] proposed *FiF-IoT*, a public BC-based forensic investigation framework that collects and stores evidence from various IoT systems. The security analysis shows the efficiency of architecture against collusion detection. The experimental results dictate negligible impacts in overall delay as well as energy consumption which proves the efficacy of the proposed architecture. Brotsis et al. [14] proposed *Cyber-Trust blockchain* (CBT) to capture and store evidence (metadata) from IoT environment in smart homes, factories, offices etc. and interfaces with various law entities, ISPs using smart contract. The system is built on Hyperledger Fabric and replaces the chain-of-custody (CoC) process of recording the chronological history of digital shreds of evidence. Liu et al. [16] incorporates BC-based architecture in service computing and data governance applications mainly to decouple data generation points and enable decentralization in all data governance

activities. The authors propose a six-layer architecture and described the role/working principle as well as design challenges for each layer. Although BC provides security and anonymity in IoT, challenges such as bandwidth, latency, complexity, and scalability play an important role.

To handle such issues, Dorri et al. [19] presented a lightweight consensus algorithm for IoT, which nearly eliminate the need of solving the cryptographic puzzle before adding a block in the BC, and with this processing time for adding a new block in the chain can be reduced. Zhang et al. [18] proposed a consensus protocol for IoT and smart city applications that use distributed ledger and uses dual-path routing strategy to provide data consistency in data transmission. Li et al. [17] proposed an architecture to maintain the privacy of the witness in the process of collecting forensic evidence and the jury in the court trial at the time of voting for a specific case. A randomized signature algorithm is used to authenticate the witness and jury and maintain the identity as private. Kumar et al. [20] proposes *IoF*, an IoT based BC assisted digital forensic framework for evidence gathering, evidence management. The system utilizes the BC consortium for effective CoC handling and signcryption techniques to maintain transparency in handling digital forensic evidence. The performance is evaluated in terms of Gas consumption, latency, throughput, memory utilization, gas, complexity and energy consumption and shows that the proposed framework is highly efficient in IoT infrastructures. Wang et al. [21] proposes privacy protection scheme *DBTMD* for telemedicine diagnosis based on double blockchain technique. The identity is stored in IPFS and access control is ensured through key parameters and encryption. The communication cost outcomes show around 82.8% improvement against the traditional health chain scheme. The authors in et al. [22] implements *BCFL*, a BC framework to forensically maintain the trustworthiness, authenticity and integrity of log evidence in the cloud ecosystem (integration of BC with cloud). The framework resolves difficulties involved in cloud-assisted by providing transaction logs as forensic evidence. The former was also analysed for transaction rate, throughput and latency using open-source ELK software. The details of the state-of-the-art approaches are presented in Table 4.

3. *NyaYa*: System Model and the proposed scheme

The section presents the system model and the proposed *NyaYa* scheme.

3.1. System model

In this section we present a BC-based digital management scheme for judicial investigations. The system model is depicted in Fig. 4. In *NyaYa*, we consider the entity set $e = \{e_{lea}, e_{ca}, e_{cd}, e_{po}, e_{dl}, e_{pl}, e_{cj}, e_{la}\}$, where e_{lea}, e_{ca}, e_{cd}, e_{po}, e_{dl}, e_{pl}, e_{cj} denotes the LEAs, CA, CD, PO, DL, PL, federal court judge (CJ) and law admin respectively. e_{la} is responsible for deploying SC in the Ethereum chain network. *NyaYa* is a four-phased scheme.

In the first phase, we consider the registration of defined entity set e in BC through by executing the SC. All the SC are deployed by e_la, once e is registered, e_{ca} visits the nearest police station to file a first information report (FIR) against e_{cd} and the same time can view the status of its case via executing the *Fetch Metadata* SC, E-FIR is recorded and digitally signed by e_{po}. The recorded version of report is considered as ELR and is stored in IPFS as comprehensive ELR C_{ELR}. At this stage, we mark the investigation as *OPEN*. We consider k C_{ELR}, numbered as $\{C_1, C_2, \ldots, C_k\}$ are registered by n e_{ca}. For any kth ELR C_{ELR_k}, a hash is generated through H_k, that denotes the kth case registration. Any kth case is uniquely identified by a case number cno_k and the hashed metadata M_{cno_k} is stored in BC by e_{po}. The entry is mapped as a one-to-one relationship $V : e_{ca} \rightarrow \{cno_k, e_{po}\}$.

Through V, in the third phase, C_{ELR} is stored in IPFS, and is mapped to V through M_{cno_k}. Once M_{cno_k} is obtained, the case state changes to *ACTIVE*. In *ACTIVE* state, we consider i e_{lea} for investigative purposes assigned to M_{cno_k} as $\{lea_1, lea_2, \ldots, lea_i\}$ The case investigative updates are timestamped at different findings at w different time instances $\{T_1, T_2, \ldots, T_w\}$ on IPFS and M_{cno_k} is appended with wth timestamp accordingly in BC.

Investigative updates are sequenced chronologically . Any ith e_{lea} can access cno_k through associated wallets W_{lea} generated at registration phase and can append the new evidence if any, all the registered entities in the public BC can fetch the case status and its decision by executing the *Fetch Metadata* SC which results the hash key address of IPFS to locate the chronological updates of the case. Thus and transparency is maintained among entity set e and to all the publicly register users in the BC.

In the final phase, we consider the case investigations can be presented before e_{cj} by e_{pl}. To support e_{cd}, DL is appointed. e_{cj} records the statements of both e_{pl} an e_{dl} and updates the C_{ELR}. Once C_{ELR} is updated, the final verdict is recorded and case closure is appended to C_{ELR}, and appropriate legal actions, if any, is vented out on e_{cd}, depending on being proven guilty or non-guilty. The metadata M_{cno_k} references the C_{ELR} from IPFS, and is updated accordingly. Once the hearing process is over, the case number cno_k state is marked as *CLOSED* in BC. In case of court settlements, e_{cj} issues SC to be executed among e_{ca}, and e_{cd}, with the share of e_{pl}, and e_{dl}, mentioned in terms and conditions of settlement. Thus, BC leverages a systematic investigative process and assures transparency of events.

3.2. NyaYa: The proposed scheme

The section discusses the four-phased scheme of *NyaYa*, which is depicted as follows. The entity interaction of the proposed scheme is represented in Fig. 6.

3.2.1. PHASE I: Entity e registration in BC

As depicted in Section 3.1, C_{ELR} is stored in BC as meta-information M_{cno_k}, with hash H_k as external reference to case description in IPFS. We now propose the FIR block structure, that consists of case number cno_k, hashed key of data K_{tx_k}, digital signature of any ith investigating e_{lea}, denoted as Ds_{tx_k}, and a random nonce N. The block representation is denoted as follows.

$$B_k = (Cno_k, K_{(tx_k \| Ds_{tx_k})}) \tag{1}$$

Once B_k is generated, it is hashed with H_{B_k}, with the entry of previous block hash H_{prev}. The details of the FIR block are represented in Fig. 5. The block consist of the header and the main body, the header consist of the version number to track the current protocol/software upgrade, the hash of the previous block which makes it immutable, timestamp, nonce, and Merkle root which stores hash of all C_{ELR} in the tree structure and hash of subsequent leaf node is further computed to find Merkle root M_r of the tree which helps to verify the data on the chain. The body of the block consists of blocksize, transaction counter which counts the number of processed C_{ELR}, case number and hash address of C_{ELR} which is a hash key of IPFS. In the proposed scheme, we consider the registration of ith e_{lea}, and jth e_{cj}. For registration, the entities are presented with wallet details W_e that form BC's public addresses. The wallet details are presented as follows.

$$W_e = \{A_e, Pqr_e, S_e, t', M_r\} \tag{2}$$

where A_e, K_e, S_e, t', M_r denotes the account details of e, public QR, signature, timestamp and merkle root information respectively. Based on W_e, the registration R_e is carried out as follows.

$$R_e = \{A_e, W_e, T, M_r\} \tag{3}$$

where A_e, W_e, T, M_r denotes the account details, wallet identifier, timestamp and merkle root in BC.

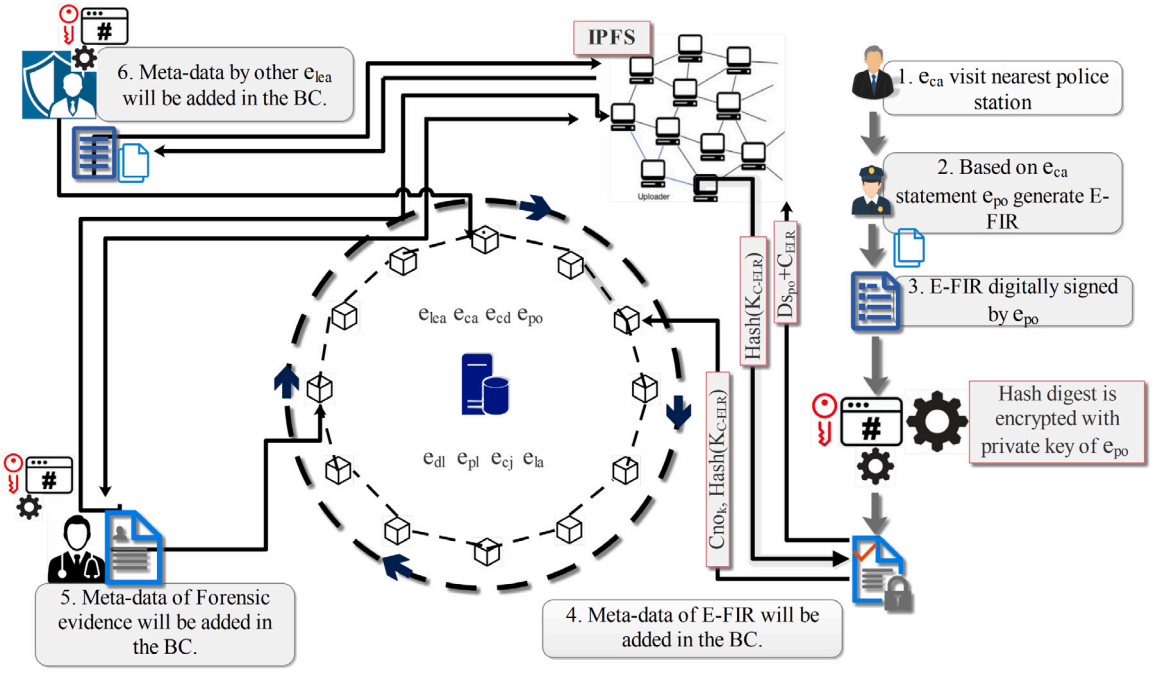


Fig. 4. *NyaYa*: System Model.

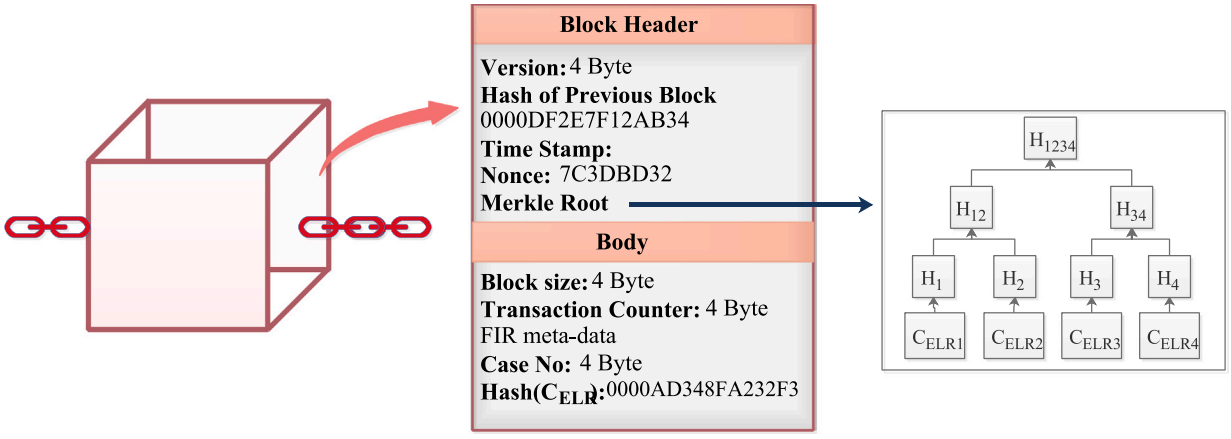


Fig. 5. Internal structure of FIR block.

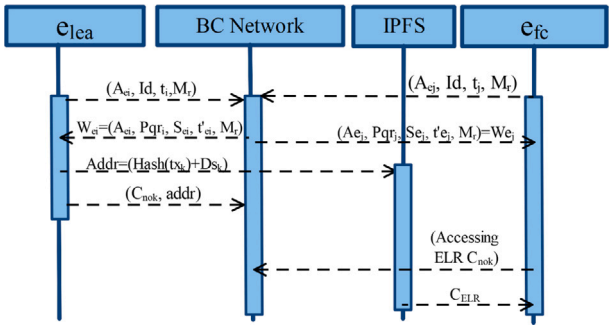


Fig. 6. Interaction between entities.

3.2.2. PHASE II: Filing of FIR by e_{ca} against e_{cd}

After the registration R_e, e_{ca} visits the nearest police station to file a FIR complaint against e_{cd}. Here, any kth particular e_{po} registers a report based on the verbal description of crime as a digital document tx_k. tx_k is then hashed and converted to hash digest Hd_k using a standard hashing H_a algorithm. The digest is signed by e_{po} and serves as signed hash. The signed record, referred as C_{ELR}, consists the following information as follows.

$$C_{ELR} = \{DS_{po}[PK_{po}(Hd_k), tx_k], STATE\} \tag{4}$$

where DS_{po} denotes the digital sign of e_{po}, PK_{po} denotes the private key of e_{po}, and *STATE* represents the case state, now marked as *OPEN*. The details of case is stored in IPFS. The process is followed for all k ELRs. We consider any kth ELR C_{ELR_k}, a unique identity (case no) cno_k is assigned, and meta-information M_{eno_k} is recorded in BC through V. At this point, the case state changes to *ACTIVE*, for investigative purposes by any ith e_{lea} e_i. For the same, e_{lea} sign the C_{ELR}, which consists of the following information

$$Ds(e_i) = \{tx_k, Ds_k\} \tag{5}$$

where $S(e_i)$ denotes the signature of ith e_{lea}. The details are stored on IPFS and is referenced through hash key based on $K_{(tx_k+Ds_{tx_k})}$. The hash key is use to update the chronological findings of case updation, and new blocks are added in chain. To ensure privacy and confidentiality

of case findings, the IPFS is referenced only through private key of ith e_{lea}, via Pk_i. The meta-information related to case cno_k is represented as follows.

$$M_{cno_k} = \{cno_k, Hash(tx_k + Ds_{tx_k})\} \tag{6}$$

where $Hash(tx_k + Ds_{tx_k})$ is stored in BC. The details of the FIR filing is presented in Algorithm 1.

Algorithm 1 Filing of FIR by e_{ca} through e_{po}

Input: Text data $data_k$.
Output: A boolean indicator $bool = \{0, 1\}$ that indicates the confirmation status of ELR registration.

1: $bool = false$
2: **for** $\forall$ $CA \leftarrow 1$ to k **do**
3: $tx_k = preprocessing(data_k)$
4: $Hd_k = (tx_k, H_a)$
5: $Ds_{tx_k} = (Hd_k, Pk_{po})$
6: $IPFS_{add} \leftarrow (K_{(tx_k, Ds_{tx_k})})$
7: $Bl_b = new(cno_k, IPFS_{add})$
8: $bool \leftarrow true$
9: **for** $\forall$ $lea_i \leftarrow 1$ to i **do**
10: **for** $cno_k \leftarrow 1$ to k **do**
11: $tx'_k = preprocessing(data'_k)$
12: $Hd_k = (tx'_k, H_a)$
13: $Ds_{tx'_k} = (Hd_k, Pk_{po})$
14: $IPFS_{add} \leftarrow (K_{(tx'_k, Ds_{tx'_k})})$
15: $Bl_b = new(cno_k, IPFS_{add})$
16: **end for**
17: **end for**
18: **if** $(bool == true)$ **then**
19: Output $e - FIR$ **is Registered,** cno_k **is allocated**
20: **else**
21: Output **error in registering** $e - FIR$
22: **end if**
23: **end for**

In algorithm 1, Lines 1–6 denotes the prepossessing of kth case data, and the addition to IPFS structure. For any kth e_{ca}, $data_k$ is pre-processed for inconsistencies to form tx_k, and hash and digital signature is computed. Lines 7–16 forms digital sign by of private key of entity Pk_{po} and C_{ELR} is stored in IPFS, with return hash to create new block B_k in on-chain structure. A boolean indicator flag $bool$, initially set to $FALSE$, is used to determine successful FIR filing. e_{lea} updates the case details in IPFS for same B_k block. The FIR filing is done $\forall$ k e_{ca}, and data is recorded in IPFS based on hash H_k. Thus, considering a total of q entries in hash table, the time complexity of algorithm 1 is $O(qk)$. For space complexity, the meta-information in BC is maintained as linear ledger, that consists of b keys. Thus, space complexity of algorithm 1 is $O(b)$.

Algorithm 2 C_{ELR} fetch from genesis block Bl_0

Input: cno_k and Bl_0.
Output: $C_{ELR} from IPFS$, $bool$ a $BOOLEAN$ indicator to indicate case match in IPFS

1: $C_{ELR} = null$
2: $bool = false$
3: **for** $\forall$ $Bl_b \leftarrow 0$ to b **do**
4: **if** $(Bl_b.cno == cno_k)$ **then**
5: $IPFS_{add} \leftarrow Get(Bl_b, Hash_{add})$
6: $C_{ELR} \leftarrow Append(data(IPFS, IPFS_{add}))$
7: $H_{d'} \leftarrow (C_{ELR}, H_a)$
8: $H_{d\varepsilon} \leftarrow Get(Bl_b, H_d)$
9: **if** $(H_{d'} == H_{d\varepsilon})$ **then**
10: $bool \leftarrow true$
11: $t \leftarrow Record(IPFS_{add})$
12: $State \leftarrow Update\_state(t(IPFS_{add}))$
13: **end if**
14: **else**
15: $State \leftarrow No\_change$
16: $bool \leftarrow false$
17: **end if**
18: **end for**
19: **if** $(bool == true)$ **then**
20: Output C_{ELR} **is generated**
21: **else**
22: Output cno_k **not found**
23: **end if**

3.2.3. PHASE III: C_{ELR} updation by e_{lea}

Any ith lea_i updates the incremental case investigative findings by storing $(tx'_k, Ds_{tx'_k})$ at off-chain IPFS. The returned hash is used to add a new block and meta-information and block sizes are updated. Also, to improve the scalability of on-chain records, M_{cno_k} is only used to update the case findings. For the same, we consider the new change as t'_x. The details are updated as follows.

$$U_k^i = \{cno_k, K_{(tx'_k, Ds_{tx'_k})}, STATE\} \tag{7}$$

where U_k^i denotes the update in M_{cno_k} by ith e_{lea}, and $STATE$ is marked as $ACTIVE$.

3.2.4. PHASE IV: C_{ELR} court proceedings and fetch from IPFS

Based on U_k^i, the case-details of any kth cno_k can be fetched by ith e_{lea} and any jth e_{cj} for court proceedings. For the same, the complete details of the case would be fetched from on-chain meta-data M_{cno_k}, where based on hash key H_k, it will sequentially scan all block header information to match with M_{cno_k}, starting from genesis block, Bl_0. In case of match at any bth block, a sequence of path is explored as $\{Bl_0, Bl_1, \ldots, Bl_b$. At Bl_t, hash key H_k and private key pair is used to fetch C_{ELR} from IPFS. Once case details are fetched, e_{pl} and e_{dl}, presents the facts against e_{cj}, and the court proceedings are recorded to cno_k. The final hearing outcome, or case-closure report, is marked in cno_k, and the case $STATE$ is changed to $CLOSED$ in BC. Based on decisions by e_{cj}, SCs are executed between e_{ca} and e_{cd}, based on indicated terms and final discourse of the dispute settlement. The SCs also includes the incentives for e_{pl} and e_{dl}, and automated funds are transferred from W_{ca}, and W_{cd} respectively. Algorithm 2 presents the query algorithm based on cno_k and information of Bl_0. Lines 1–8 of the algorithm presents the sequential scan of records based on M_{eno_k}, from Bl_0, to fetch the matching records. In case of match, lines 9–19 updates the boolean flag $BOOL$ to indicate the output of the fetched record from query. Since we have to run a sequential scan for b blocks, where each block contains k cases. The time complexity of algorithm 2 is $O(bk)$. To store the records, we require a list structure, hence the space complexity is $O(b)$.

4. Performance evaluation of *NyaYa*

The section discusses the performance evaluation of *NyaYa* based on parameters-mining cost of C_{ELR}, query fetching time based on associative cache μ, and obtained IPFS bandwidth for processed blocks B. For mining cost, the scheme is compared with Chopade et al. [15], and for IPFS bandwidth, the scheme is compared with Bragagnolo et al. [23], and Gupta et al. [24]. As only M_{eno_k} is stored in on-chain storage, we have considered mining cost and block processing time. Also, we have studied the effect of query latency by firing ethereum query language (EQL) tags on stored blocks by varying the cache size μ. The details of the same are now presented.

4.1. Experimental setup

For BC setup, we have considered a Linux Ubuntu LTS *v18.04*, running instance of node *npm v 6.7.1*. We have considered Intel core *i*5 machines with $4GB$ RAM and $500GB$ external-drive capacity. For smart contracts, we have considered Remix ethereum *v 0.10.3*, with injected *metamask* and *web3.js* libraries. For formal validation of smart contracts, we have considered *Mythril* open source tool [25] that tests transactional security flaws in SC like origin, re-entrancy, order-dependence, and time-stamp dependencies, that might be exploited.

4.2. Formal security verification of NyaYa

The section presents the formal verification of SCs proposed in *NyaYa* scheme. We use *Mythril* to test security vulnerabilities of all SC, it uses taint analysis, symbolic execution and satisfiability modulo theory analysis to detect security flaws in the SC. As indicated in Fig. 7, the proposed SC is devoid of any security loopholes and outputs *"No issues were detected"*.

Table 5
Accessibility and changeability rules for SC.

S.No	Smart Contract	Description	Execution rights								Changeability Rights
			e_{lea}	e_{ca}	e_{cd}	e_{po}	e_{dl}	e_{pl}	e_{cj}	e_{la}	
1	Entity Registration	Register entities in BC	✓	✓	✓	✓	✓	✓	✓	✓	e_{la}
2	FIR Registration	e-FIR is registered	✗	✗	✗	✓	✗	✗	✗	✗	e_{la}
3	Adding FIR Meta-Data	e-FIR meta-data is added in BC	✓	✗	✗	✓	✗	✗	✓	✗	e_{la}
4	Fetch Metadata	Read the data in BC	✓	✓	✓	✓	✓	✓	✓	✓	e_{la}

✗: No grant on Execution, ✓: Grant on Execution.

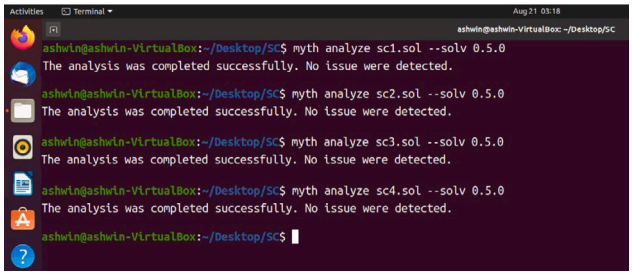


Fig. 7. Formal Security verification of SC's with MyThril Tool.

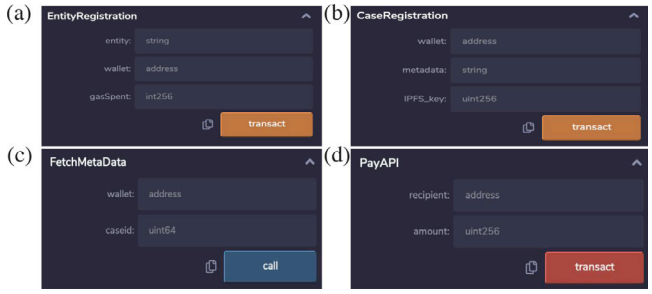


Fig. 8. The proposed SC functionalities in *NyaYa* scheme (a) User Registration, (b) Case Registration (c) Fetching Metadata, and (d) Payment Interface.

4.3. Functionality of smart contracts

The SC are developed using solidity language and compiled on ethereum virtual machine (EVM), through remix IDE. The considered functionalities are depicted in 8. In Fig. 8(a), we propose the functionality of entity e wallet W_e registration, and the amount of spent gas cost. In Fig. 8(b), we present the SC for case registration by any k^{th} e_{ca} with assigned case ID cno_k. The SC references the IPFS key to fetch case details and only meta-data M_{cno_k} is stored in BC. Fig. 8(c) presents the SC functionality of fetching C_{ELR} record, as depicted in algorithm 2, and Fig. 8(d) represents the functionality of wallet W_e once the case closure report is updated in BC, and case decision is provided by e_{cj}. Based on the quantum of the crime, e_{cd}, if proven guilty, has to pay penalty to e_{ca}. Thus, it shows the recipient e wallet and the amount for transfer. Similarly, lawyers e_{pl} and e_{dl} are paid there fees through the same functionality.

Table 5 shows the accessibility and changeability rules for SC deployed on Ethereum chain, all the SC are created and deployed by e_{la}. The SC are deployed with the constraints that only intended set of wallet address can execute the SC on chain and only e_{la} can upgrade the SC. Like first row of the table shows all any one on the network can register itself by executing the SC *Entity Registration*. Similarly only e_{po} have the permission to execute *FIR Registration* i.e. only e_{po} can register the FIR of e_{ca} in the system. Any one in the public network can fetch the details of case status by executing the *Fetch Metadata* SC.

4.4. Simulation results

To assess the simulation results, we consider the hash keys H_k to be uniformly distributed and mapped to the IPFS database. The details of the simulation results are now presented in Fig. 9.

In Fig. 9(a), we evaluate the mining cost of block b with respect to number of transactions per block. We compute the block information as 4 bytes, 80 bytes for entire block header, and 2 bytes for transaction count in M_{cno_k}. For E_{CLR} entry, it has 4 bytes case-ID cno_k, and 4 bytes IPFS hash address of the located file. For 1000 transactions, C_{ELR} size is $\approx$ 3.7 kilo-bytes (KB). The cost of mining ethereum data in the third quarter $Q3$ (2020) is $13.82USD/KB$. Based on the above points, we compute the mining cost compared to Chopade et al. [15] and Lone et al. [26]. At USD 6 cost, the proposed scheme can accommodate 22456 transactions, compared to 21497 transactions in Lone et al. [26], and 3000 transactions in Chopade et al. [15]. This is due to storing C_{ELR} as off-chain IPFS data.

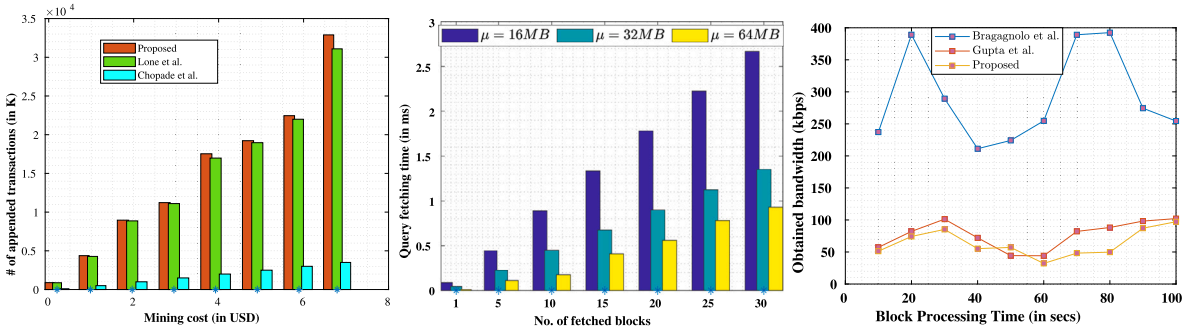
Fig. 9(b) shows the impact of query fetching time from IPFS, with cache inclusion. We have considered cache-size μ as 16 MB, 32 MB, and 64 MB, respectively. As evident, with an increase in cache size, the EQL time is significantly decreased. At 25 blocks and μ as $32KB$, the query time is 0.852 ms, due to more associated hits due to locality of reference.

Fig. 9(c) presents the network bandwidth of proposed scheme against conventional schemes [23,24]. As ethereum bandwidth is $\approx$ 200 – 300 Kbps. Due to off-chain record storage, more transactions are appended in blocks and thus improves the utilization bandwidth.

Fig. 10(a) presents the transactional throughput cost in the proposed scheme. In the traditional scheme, we consider normal databases for storing of ELRs. As indicated in Fig. 10(a), the query fetch time, with cache proximity for 64 MB cache allows a transactional throughput of $\approx 64KBps$ for one appended C_{ELR} is 101.27 Mbps, as depicted in Chopade et al. [15]. In the case of IPFS, the storage is a distributed off-chain storage, and with a cache size of 64 MB, the obtained throughput is $\approx 89KBps$ for single C_{ELR}. Thus, with IPFS, the obtained throughput is 139.45 Mbps. The increased throughput is due to the storage of meta-information in BC only, i.e., M_{cno_k}, which allows more transactions to be added time-quantum and increases the overall throughput.

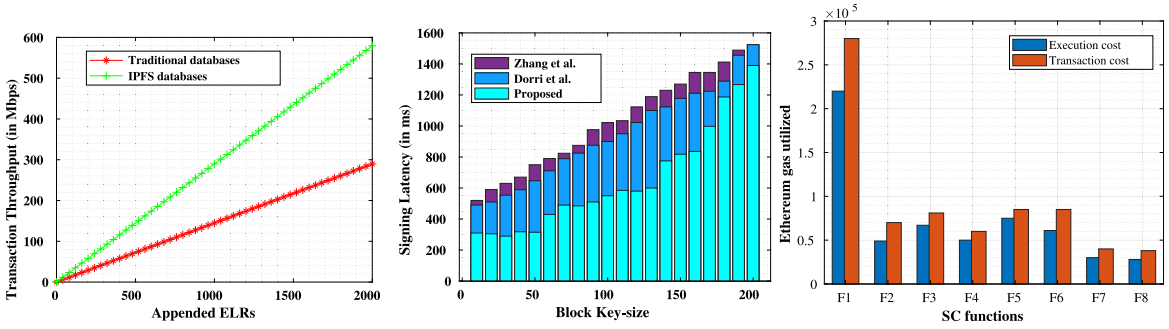
Fig. 10(b) presents the comparative analysis of signing latency with the schemes proposed by Zhang et al. [18], and Dorri et al. [19]. Zhang et al. scheme an average signing latency of 877.65 ms, and Dorri et al. scheme has an average signing latency of 948.4 ms. In the proposed scheme, the signature operation DS_{po} is validated and is stored in the hashed form in M_{cno_k}. Due to this, the verification process of generated signatures is faster. As evident, the scheme has an average signing latency of 622.95 ms, which outperforms the above schemes. At block key-size of 100 bytes, the signing latency of Zhang et al. is 1022 ms, and Dorri et al. is 900 ms. Compared to this, the signing latency is 550 ms. Thus, the scheme proposes an improvement in signing latency of 34.38% over the conventional approaches.

Fig. 10(c) presents the simulation results of proposed SC in terms of execution and transaction costs. We have proposed four key functionalities of SC in the *NyaYa* scheme. Each function is executed with a confirmation status recorded in a transactional ledger in BC in terms of state functions. Thus, 8 major functions $\{F1, F2, \ldots, F8\}$, are proposed in the scheme. $F1$ is the entity registration function, and $F2$ represents



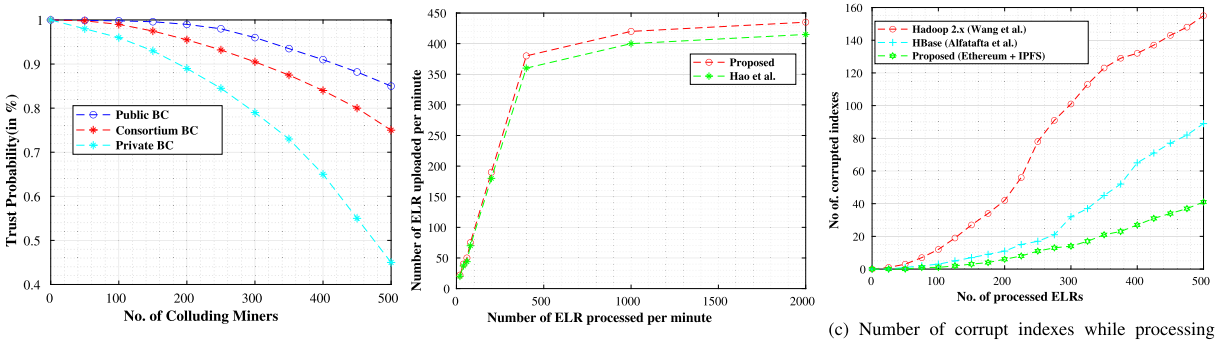
(a) Lower mining costs for E_{CLR} (b) Impact of cache μ to improve cno_k fetch (c) Obtained bandwidth and block processing time

Fig. 9. *NyaYa*: Simulation results on impact on BC-node characteristics through off-chain IPFS.



(a) Transaction Throughput of IPFS vs traditional distributed databases for C_{ELR} fetch (b) Signing Latency (in ms) of Pk_{po} against comparative schemes (c) Transaction and Execution costs of 8 SC functions

Fig. 10. *NyaYa*: Simulation results on IPFS node throughput, signing latency, and gas utilization of SC functions.



(a) Effect of no. of colluding miners on chain (b) ELR uploaded and processed per minute (c) Number of corrupt indexes while processing ELR

Fig. 11. *NyaYa*: Simulation results on impact on collusion attacks, C_{ELR} processing, and resilience of IPFS.

the transactional output of $F1$. $F3$ is the case registration, and $F4$ confirms the case status. $F5$ is the meta-information fetch from IPFS, and $F6$ represents the fetch status function output. $F7$ is the payment interface, and $F8$ confirms the notification of payments and associated historical transactions.

The transactional cost of data storage on ethereum based BC is depicted as follows [27]

$$USD6 \times ((x \times 20000)/1000000000) \times 129.34 \tag{8}$$

In *NyaYa* scheme, we have proposed the execution of data storage as off-chain in IPFS, which is fetched through reference hash H_k. This reduces the overall transaction and execution cost of SC executions. For ethereum BC, the storage cost is determined based on word size, where each word is 32 kilobytes. Thus, the amount of gas required to store the word length is $\approx$ 20000 Gas, and thus, for storage of 5 KB of data, the gas required is $5 \times \frac{2^{10}}{2^5} \times 20000$, i.e. $\approx 32,00,000$. Averaging over all the SC functions, we have computed the transaction and execution costs.

Next, we present the simulation results that measures the impact of collusion attack and corrupted IPFS indexes that fetches incorrect C_{ELR} in the proposed scheme. We measure the impact of collusion attacks against consortium and private chains, and the effect of processed ELRs against Hao et al. [28], and the resilience of IPFS against Hadoop 2.x scheme, as presented in Wang et al. [29], and HBase in Alfatafta et al. are presented [30]. Fig. 11 presents the details of the simulation results.

In Fig. 11(a), we present the comparative analysis of public, consortium, and private BC with trust probability. We measure the trust probability of a blockchain network as follows. We measure the probability through elected miners $\{m_1, m_2, \ldots, m_k\}$, through the proposal of fair block addition. In private chains, one vulnerable entity may collude with a group of a miner to reverse the transactions, through the generation of 50% of the total hash power in the network, which drastically reduces the trust, and allows m_k to add a favoured block in the chain. Thus, it allows the chain to grow according to dishonest

Table 6
Insights of comparative security analysis against existing schemes.

Scheme	CC	CCM	ME
Odelu et al. [31]	$7E_{ase} + 12H_{SHA} + 2T_{bpc} \approx 505.72$ ms	240 bytes	3
Kabra et al. [32]	$2E_{ase} + 1S_{gen} + 2S_{ver} + 1H_{SHA} \approx 192.14$ ms	203 bytes	4
Patel et al. [33]	$8H_{SHA} + 2nonce + 2merkle - root + 3T_{append} + 3E_{ase} \approx 20.96$ ms	121 bytes	7
Proposed *Nyaya*	$8H_{SHA} + 2nonce + 2merkle - root + 3T_{append} + 2E_{ase} \approx 6.56ms$	316 bytes	4

E_{ase}: Asymmetric encryption cost; H_{SHA}: Hash operation cost; T_{bpc}: Bilinear pairing cost; E_{se}: Symmetric encryption cost; D_{sc}:Digital Signing cost; V_c: Verification cost; S_{gen}: Signature generation cost; S_{ver}: Signature verification cost, *nonce*: Time-stamp cost; $merkle - root$: timestamp cost to refer genesis block hash; T_{append}: Cost of appending blocks to chain.

block proposal, and once the length of dishonest chain C_{dis}, becomes larger than honest chain length C_{hon}, i.e. $L(C_{dis}) > L(C_{hon})$, then C_{dis} is considered as valid chain, and all the nodes accept the same as it is through the elected consensus mechanism. While, in public BC, owing to the chain length, and difficulty problem, it is impossible to reduce trust in the network to less than 50%, as the miners, even if all collude, would not be able to generate a very high hash rate, as it would require significant resources, which is currently not feasible. Thus, as indicated, at 500 colluding miner nodes, public BC shows a trust probability of 0.887%, compared to 0.765% in consortium BC, and 0.455% in private BC.

Fig. 11(b) shows the effect of uploaded C_{ELR} per minute against the number of processed C_{ELR}. The scheme is compared against Hao et al. [28], which also proposes IPFS as off-chain storage of records. As we only include meta-information, which is 32 bytes long in the transaction field, the rate of C_{ELR} upload is significantly higher than conventional BC-based schemes. However, in *NyaYa*, we propose a single encryption and signing phase by e_{po} while it uploads C_{ELR} in the IPFS. On the other hand, Hao et al. [28] proposes a dual-signature based scheme, which improves the security model, but adds the additional overhead of the extra signature phase. The same is evident in Fig. 11(b), where at 475 transactions, the proposed scheme uploads 377 ELR records, compared to 354 records in Hao et al.. On an average, we obtain an improvement of 6.77% of ELR uploading latency over the previous scheme, however, the scheme *NyaYa* has a trade-off with *'good-enough* security, over tight-security, as in Hao et al. [28].

Finally, we present the impact of corrupted indexes in distributed storage schemes, as in Hadoop 2.x [29], and HBase [30]. Fig. 11(c) presents the results of the simulation. In Hadoop 2.x, data corruption usually occurs due to improper run-time checks, race conditions in multiple threads, and inconsistent state pointer information. Due to this, the block meta-information is updated, and the same is communicated to nodes in the network. On average, in Hadoop 2.x ecosystems, data integrity checks reveals that on an average out of every 1000 disk retrievals, there are 7 corrupted block information and a 1 system-file corruption state. Similarly, in HBase, due to partial network partitioning, and system configuration changes, there are corrupted meta-information blocks. On average, it is estimated that $\approx$ 2 meta-blocks are corrupted out of 1000 blocks, which is lower than Hadoop 2.x. However, in IPFS, the storage is resilient, and through redundancy via distributed hash tables, and block exchange, the failure rate is lower than both Hadoop 2.x and HBase systems. The same is measured in Fig. 11(c). We measure the processed C_{ELR} that are stored as multiple transactions and appended in blocks. At 350 C_{ELR}, Hadoop 2.x has 123 corrupted indexes (includes block, meta, system, and partition corruption) information, and HBase has 45 corrupted indexes. In comparison, IPFS has 21 corrupted indexes overall, which improves the scheme resilience against schemes that employ Hadoop 2.x and HBase ecosystems.

4.5. Computation and communication costs

In this section, we present the details of the computation and communication costs of different security identifiers in the scheme. The security cost parameters are taken from Bhattacharya et al. [34]. The details are presented as follows.

4.5.1. Computation cost

The computation cost of registering the FIR by adding the block by e_po will be calculated based on the algorithm 1. The algorithm includes pre-processing of $data_k$ followed by a digital signature which includes t_x to be converted into hash digest Hd_k. That digest is encrypted with asymmetric encryption, encrypted data will store in IPFS, and the hash address received from IPFS is finally stored in BC along with cno_k and other data called as M_{cno_k}. This involves 4 hash operation, 1 nonce, 1 asymmetric encryption and 1 *merkle_root* hash is present, Thus the time requires to register the $e - FIR$ in the BC is $0.00032 + 0.0056 + 0.00032 + 0.00032 + 0.00032 + 0.00032 \approx 0.00720$ s.

Then, to fetch C_{ELR} executes algorithm 2 that includes fetching the hashed address from the block and use that hash address to get the data from IPFS. We calculate the hash digest of received data to confirm there is no manipulation in transit. This include of 1 hash operation, asymmetric decryption E_{asym} of PO_{Pk}, 1 nonce identifier, 1 *merkle_root* hash. The cost required to search any FIR is $0.00032 + 0.0056 + 0.00032 + 0.00032 \approx 0.00656$ s. Thus, the overall computation cost for *NyaYa* scheme is $0.00720 + 0.00656 \approx 0.01376$ s or 13.76 ms.

4.5.2. Communication costs

The communication cost is evaluated with reference to algorithm 1, Hd_k involves 1 hash operation on tx_k of 32 bit to calculate the digest followed by asymmetric encryption of 384 bits for signing the hash digest Ds_{tx_k} and operation on IPFS involves 256 bit hash conversion, and block contains 32 bit timestamp, 32 bit version, 32 bit difficulty target, 32 bit nonce, 256 bit previous block hash, 256 bit *merkle_root*, 256 bit hashed IPFS address. Thus total exchange of data is $32 + 384 + 256 + 32 + 32 + 32 + 32 + 256 + 256 + 256 \approx 1556$ bits or 196 bytes. The number of message exchanges is 2. Similarly, to fetch C_{ELR} from IPFS concerning algorithm 2, block access requires 32 bit timestamp, 32 bit version, 32 bit difficulty target, 32 bit nonce, 256 bit previous block hash, 256 bit *merkle_root*, 256 bit hashed IPFS address and operation on IPFS involves 256 bit hash conversion and 324 bit to calculate the hash digest of received C_{ELR} and 32 bit digest from the block is accessed. Thus total exchange data is $32 + 32 + 32 + 32 + 256 + 256 + 256 + 256 + 32 + 32 \approx 960$ bits or 120 Bytes. The number of message exchanges is 2.

Table 6 presents the comparative analysis of computation, communication, and message exchange costs against existing related schemes.

4.6. Efficiency of Nyaya against conventional schemes

Next, we compare the proposed scheme in terms of chosen parameters against conventional schemes. Table 7 presents the comparative analysis. As indicated, the proposed scheme considers all the parameters and outperforms the similar and existing state-of-the-art schemes.

5. Conclusion

The paper proposes a BC-based digital law evidence scheme, named as *NyaYa*, for effective management of ELRs and future judicial investigations. In this paper, a BC-based case record storage, processing, retrieval, and update is proposed for judicial investigations. We proposed a four-phase scheme that highlights the need for chronology,

Table 7
Comparative Analysis with existing schemes.

Parameter	Billard et al. [12]	Hossain et al.[11]	Hardwick et al.[13]	Hossain et al. [2]	Brotsis et al. [14]	Patel et al. [33]	Chopade et al. [15]	Proposed *NyaYa*
A1	✗	✓	✗	✓	✗	✓	✓	✓
A2	✓	✓	✗	✓	✓	✓	✗	✓
A3	✓	✓	✓	✓	✓	✓	✓	✓
A4	✓	✓	✗	✓	✓	–	✓	✓
A5	✓	✓	✓	✓	✓	✓	✓	✓
A6	✗	✗	✗	✗	✗	✗	✓	✓
A7	✗	✗	✗	✗	✗	✓	✗	✓
A8	✗	✗	✗	–	–	✗	✗	✓
A9	✗	✗	✗	–	–	–	✗	✓
A10	✗	✗	✗	–	–	–	✗	✓

A1: Encryption; A2: Privacy; A3: Trust; A4: Confidentiality A5: Record Tampering; A6: Digital Signatures; A7: Smart Contracts; A8: Distributed Storage; A9; Formal verification; A10: ELRs; ✓ shows parameter is present; ✗ shows parameter is absent; & shows parameter is not considered .

transparency, and trust among judicial stakeholders and provides justice to the appellant in the court of law. The scheme stores case information in BC, with reference to IPFS off-chain, which improves the scalability of chain operations. In case of penalties and lawyer payments, SCs are executed. Finally, we compared the performance evaluation of *NyaYa* against parameters like-mining cost, EQL query fetching time, and obtained IPFS bandwidth, with proposed formal verification and deployment of SC functionalities.

As part of future work, the authors would investigate the effects of cache coherency in IPFS storage to optimize and reduce the query fetch time of ELRs.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] National judicial data grid (District and Taluka Courts of India). 2020, https://njdg.ecourts.gov.in/njdgnew/?p=main/pend_dashboard. [Accessed: 2020-10-16].

[2] Hossain M, Karim Y, Hasan R. FIF-IoT: A forensic investigation framework for IoT using a public digital ledger. In: 2018 IEEE international congress on Internet of Things (ICIOT), San Francisco, CA, USA. IEEE; 2018, p. 33–40.

[3] Lenk H. Investor-state dispute settlement: Constitutional challenges and pitfalls. In: Andenas M, Pantaleo L, Happold M, Contartese C, editors. EU external action in international economic law: recent trends and developments. The Hague: T.M.C. Asser Press; 2020, p. 121–51. http://dx.doi.org/10.1007/978-94-6265-391-7_6, URL https://doi.org/10.1007/978-94-6265-391-7_6.

[4] Reyes CL, Rosario NM, Cannon RM, Tall R. Blockchain and the law. J Marshall J Info Tech & Privacy L 2019;34:1.

[5] Blockchain and associated legal issues for emerging markets. 2020, https://openknowledge.worldbank.org/handle/10986/31202. [Accessed: 2020-10-16].

[6] Singh A, Tiwari AK, Bhattacharya P. Bit error rate analysis of hybrid buffer-based switch for optical data centers. J Opt Commun 2019;1(ahead-of-print).

[7] Shukla A, Bhattacharya P, Tanwar S, Kumar N, Guizani M. DwaRa: A deep learning-based dynamic toll pricing scheme for intelligent transportation systems. IEEE Trans Veh Technol 2020;69(11):12510–20. http://dx.doi.org/10.1109/TVT.2020.3022168.

[8] Kim D, Ihm S-Y, Son Y. Two-level blockchain system for digital crime evidence management. Sensors 2021;21(9). http://dx.doi.org/10.3390/s21093051, URL https://www.mdpi.com/1424-8220/21/9/3051.

[9] Bhattacharya P, Tanwar S, Shah R, Ladha A. Mobile edge computing-enabled blockchain framework—A survey. In: Singh PK, Kar AK, Singh Y, Kolekar MH, Tanwar S, editors. Proceedings of ICRIC 2019. Cham: Springer International Publishing; 2020, p. 797–809.

[10] Bodkhe U, Mehta D, Tanwar S, Bhattacharya P, Singh P, Hong W. A survey on decentralized consensus mechanisms for cyber physical systems. IEEE Access 2020;8:54371–401. http://dx.doi.org/10.1109/ACCESS.2020.2981415.

[11] Hossain MM, Hasan R, Zawoad S. Probe-IoT: A public digital ledger based forensic investigation framework for IoT. In: INFOCOM Workshops, Honolulu, HI, USA. 2018. p. 1–2.

[12] Billard D. Weighted forensics evidence using blockchain. In: Proceedings of the 2018 international conference on computing and data engineering, Shanghai, China. 2018. p. 57–61.

[13] Hardwick FS, Akram RN, Markantonakis K. Fair and transparent blockchain based tendering framework-a step towards open governance. In: 2018 12th IEEE international conference on big data science and engineering (TrustCom/BigDataSE), Beijing, China. IEEE; 2018, p. 1342–7.

[14] Brotsis S, Kolokotronis N, Limniotis K, Shiaeles S, Kavallieros D, Bellini E, et al. Blockchain solutions for forensic evidence preservation in IoT environments. In: 2019 IEEE conference on network softwarization (NetSoft), Paris, France. IEEE; 2019, p. 110–4.

[15] Chopade M, Khan S, Shaikh U, Pawar R. Digital forensics: Maintaining chain of custody using blockchain. In: 2019 Third international conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), Palladam, India. IEEE; 2019, p. 744–7.

[16] Liu X, Sun SX, Huang G. Decentralized services computing paradigm for blockchain-based data governance: Programmability, interoperability, and intelligence. IEEE Trans Serv Comput 2019;13(2):343–55.

[17] Li M, Lal C, Conti M, Hu D. LEChain: A Blockchain-based lawful evidence management scheme for digital forensics. Future Gener Comput Syst 2020.

[18] Zhang W, Wu Z, Han G, Feng Y, Shu L. LDC: A lightweight dada consensus algorithm based on the blockchain for the industrial Internet of Things for smart city applications. Future Gener Comput Syst 2020.

[19] Dorri A, Kanhere SS, Jurdak R, Gauravaram P. LSB: A lightweight scalable blockchain for IoT security and anonymity. J Parallel Distrib Comput 2019;134:180–97.

[20] Kumar G, Saha R, Lal C, Conti M. Internet-of-Forensic (IoF): A blockchain based digital forensics framework for IoT applications. Future Gener Comput Syst 2021;120:13–25. http://dx.doi.org/10.1016/j.future.2021.02.016, URL https://www.sciencedirect.com/science/article/pii/S0167739X21000686.

[21] Wang W, Wang L, Zhang P, Xu S, Fu K, Song L, et al. A privacy protection scheme for telemedicine diagnosis based on double blockchain. J Inf Secur Appl 2021;61:102845. http://dx.doi.org/10.1016/j.jisa.2021.102845, URL https://www.sciencedirect.com/science/article/pii/S2214212621000818.

[22] Awuson-David K, Al-Hadhrami T, Alazab M, Shah N, Shalaginov A. BCFL logging: An approach to acquire and preserve admissible digital forensics evidence in cloud ecosystem. Future Gener Comput Syst 2021;122:1–13. http://dx.doi.org/10.1016/j.future.2021.03.001, URL https://www.sciencedirect.com/science/article/pii/S0167739X21000807.

[23] Bragagnolo S, Rocha H, Denker M, Ducasse S. Ethereum query language. In: Proceedings of the 1st international workshop on emerging trends in software engineering for blockchain, Madrid, Spain. 2018. p. 1–8.

[24] Gupta R, Shukla A, Mehta P, Bhattacharya P, Tanwar S, Tyagi S, et al. VAHAK: A blockchain-based outdoor delivery scheme using UAV for healthcare 4.0 services. In: IEEE INFOCOM 2020 - IEEE conference on computer communications workshops (INFOCOM WKSHPS), Toronto, on, Canada. 2020. 255–60.

[25] Patel NS, Bhattacharya P, Patel SB, Tanwar S, Kumar N, Song H. Blockchain-envisioned trusted random oracles for IoT-enabled probabilistic smart contracts. IEEE Internet Things J 2021;1. http://dx.doi.org/10.1109/JIOT.2021.3072293.

[26] Lone AH, Mir RN. Reputation driven dynamic access control framework for IoT atop PoA ethereum blockchain. IACR Cryptol EPrint Arch 2020;2020:566.

[27] Wood G, et al. Ethereum: A secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper 2014;151(2014):1–32.

[28] Hao J, Sun Y, Luo H. A safe and efficient storage scheme based on blockchain and IPFS for agricultural products tracking. J Comput 2018;29(6):158–67.

[29] Wang P, Dean DJ, Gu X. Understanding real world data corruptions in cloud systems. In: 2015 IEEE international conference on cloud engineering. 2015, p. 116–25. http://dx.doi.org/10.1109/IC2E.2015.41.

[30] Alfatafta M, Alkhatib B, Alquraan A, Al-Kiswany S. Toward a generic fault tolerance technique for partial network partitioning. In: 14th {USENIX} symposium on operating systems design and implementation ({OSDI} 20). 2020. p. 351–68.

[31] Odelu V, Das A, Wazid M, Conti M. Provably secure authenticated key agreement scheme for smart grid. IEEE Trans Smart Grid 2018;9(3):1900–10. http://dx.doi.org/10.1109/TSG.2016.2602282.

[32] Kabra N, Bhattacharya P, Tanwar S, Tyagi S. MudraChain: Blockchain-based framework for automated cheque clearance in financial institutions. Future Gener Comput Syst 2020;102:574–87. http://dx.doi.org/10.1016/j.future.2019.08.035, URL http://www.sciencedirect.com/science/article/pii/S0167739X19311896.

[33] Patel S, Bhattacharya P, Tanwar S, Kumar N. KiRTi: A blockchain-based credit recommender system for financial institutions. IEEE Trans Netw Sci Eng 2020;1. http://dx.doi.org/10.1109/TNSE.2020.3005678.

[34] Bhattacharya P, Tanwar S, Bodkhe U, Kumar A, Kumar N. EVBlocks: A blockchain-based secure energy trading scheme for electric vehicles underlying 5G-V2X ecosystems. Wirel Pers Commun 2021;1–41.