future internet

MDPI

Review

Redactable Blockchain: Comprehensive Review, Mechanisms, Challenges, Open Issues and Future Research Directions

Shams Mhmood Abd Ali [1,2], Mohd Najwadi Yusoff [1,*] and Hasan Falah Hasan [1,3]

[1] School of Computer Science, Universiti Sains Malaysia, Gelugor 11800, Penang, Malaysia
[2] College of Literature, Aliraqia University, Hayba Katoon, Street 22, Avenue 308, Adhamyh, Baghdad 7366, Iraq
[3] College of Engineering, Aliraqia University, Hayba Katoon, Street 22, Avenue 308, Adhamyh, Baghdad 7366, Iraq
* Correspondence: najwadi@usm.my

Abstract: The continuous advancements of blockchain applications impose constant improvements on their technical features. Particularly immutability, a highly secure blockchain attribute forbidding unauthorized or illicit data editing or deletion, which functions as crucial blockchain security. Nonetheless, the security function is currently being challenged due to improper data stored, such as child pornography, copyright violation, and lately the enaction of the "Right to be Forgotten (RtbF)" principle disseminated by the General Data Protection Regulation (GDPR), where it requires blockchain data to be redacted to suit current applications' urgent demands, and even compliance with the regulation is a challenge and an unfeasible practice for various blockchain technology providers owing to the immutability characteristic. To overcome this challenge, mutable blockchain is highly demanded to solve previously mentioned issues, where controlled and supervised amendments to certain content within constrained privileges granted are suggested by several researchers through numerous blockchain redaction mechanisms using chameleon and non-chameleon hashing function approaches, and methods were proposed to achieve reasonable policies while ensuring high blockchain security levels. Accordingly, the current study seeks to thoroughly define redaction implementation challenges and security properties criteria. The analysis performed has mapped these criteria with chameleon-based research methodologies, technical approaches, and the latest cryptographic techniques implemented to resolve the challenge posed by the policy in which comparisons paved current open issues, leading to shaping future research directions in the scoped field.

Keywords: blockchain; security; distributed ledger; immutability; redaction mechanism; chameleon hash

Citation: Abd Ali, S.M.; Yusoff, M.N.; Hasan, H.F. Redactable Blockchain: Comprehensive Review, Mechanisms, Challenges, Open Issues and Future Research Directions. *Future Internet* **2023**, *15*, 35. https://doi.org/10.3390/fi15010035

Academic Editors: Christoph Stach and Clémentine Gritti

Received: 27 November 2022
Revised: 27 December 2022
Accepted: 31 December 2022
Published: 12 January 2023

1. Introduction

Blockchain technology has recently acquired significant attention from academicians and the industry as evidenced by multiple applications, including copyright dispute resolution [1], product traceability [2], electronic voting [3], storage services [4], healthcare services [5], product tracking throughout the entire supply chain [6], and data management [7] in the internet-of-things (IoT) [8]. Prior researchers [9] predicted global expenditures on blockchain solutions to expand from 1.5 billion dollars in 2018 to 15.9 billion dollars annually by 2023. Blockchain technology provides pertinent data solutions through decentralization, collection, storage, and processing by recording transactions into respective blocks, which are appended to one another through cryptology to provide high security and validity levels. Specifically, each block is equipped with a reference number, known as a hash, to be attached to the subsequent block [10,11] to maintain technical immutability, which requires all attached blocks to be edited if a specific block is to be amended [12]. Blockchain functions as a distributed database, which is regulated by a peer-to-peer network wherein all stakeholders (nodes) comply with a stipulated software protocol (consensus) to communicate and validate existing records or blocks [13,14]. With every node possessing

all copies of other blocks without centralized and official replicas, blockchain serves as a 'trustless' system without a third party to validate every performed transaction and facilitate interactions between corporations and clients [15,16]. Immutability is a fundamental attribute of conventional blockchain technology to maintain high data integrity, which prevents transaction data modification while expediting the data auditing process [17,18]. Notwithstanding, establishing procedures to authorize blockchain data redaction is also essential owing to three factors. Correspondingly, immutability might be abused for malevolent motives, such as illegal information storage and dissemination [19]. In [20], eight records related to sexual content were discovered in the Bitcoin blockchain, of which two comprised 274 child pornography website links, with 142 associated with darknet services. Therefore, the available arbitrary data storage is exploited due to child pornography, improper content, and illicit material violating intellectual rights [21]. With the misusage, users might be reluctant to adopt blockchain technology to avoid unintentional possession of illicit or inappropriate digital content and law violation. If the negative situation persists, thousands of blockchain nodes might be affected, consequently diminishing the Bitcoin ecosystem and functionality. Hence, continuously appending the latest digital information would not be feasible without illegal content being removed from the blockchain, which poses an essential prerequisite to further adoption by being obligatory for law enforcement agencies, including Interpol [22]. Another rationale for requiring the implementation of a redactable blockchain is triggered by imposed laws and regulations. According to the European General Data Protection Regulation (GDPR) [23,24], users' data contain the "right to be forgotten", which allows all users to remove personal details and relevant duplicates while simultaneously amending personal data anonymously attached to former blocks [25]. Furthermore, blockchain immutability might not be pertinent to emerging blockchain-based applications, which adamantly request a certain flexibility level for data redaction. Examples include, but are not restricted to, the data stored on the blockchain, that may concern users' confidential and sensitive information, such as healthcare and insurance records [26,27]. The users might prefer removing sensitive details from the platform while updating the information if required, including the contract amendment service, and deleting redundant data to free more space in the IoT-based blockchain systems [28,29].The current study would allow blockchain developers and researchers to capture a holistic view and ease establishing future blockchain redaction. Summarily, this study contributed to the existing knowledge corpus by reviewing present redaction techniques utilizing the chameleon and non-chameleon hashing function, analyzing blockchain security and performance features thoroughly, outlining blockchain information redaction challenges, and differentiating redactable blockchain systems comprehensively before providing open issues and future directions. The following content of this paper is organized as follows: the present study scrutinizes in Section 2 the blockchain technology research model with the respective characteristics elucidated. Section 3 represents blockchain construct features discussed in depth. Section 4 illustrates blockchain types. Section 5 thoroughly discusses the security properties of blockchain technology before examining multiple design challenges for redactable blockchain in Section 6. In Section 7, approaches and mechanisms are primarily illustrated, analyzed, and heavily discussed. Section 8 represents comparisons to the state of the art discussed in Section 7. Open issues and future research directions are explained in Section 9, which draws the path for researchers to further investigations. Section 10 contains conclusions and, finally, Appendix A contains Table A1 that contains a list of acronyms.

2. Blockchain Technology

Satoshi Nakamoto introduced Bitcoin as an innovative, decentralized, and peer-to-peer cryptocurrency system founded in 2008 [30], which seeks to develop a trustless yet credible financial platform to facilitate transactions through Bitcoins without existing financial institutions or third parties as intermediaries. The fundamental technology is blockchain, which harnesses the strength of peer-to-peer networks, Merkle trees, consensus, and public-key

cryptology to maintain high assurance and validity degrees of every transaction and resolve multiple issues [31]. Essentially, a blockchain could be defined as a decentralized, transparent, and transactional database shared among all network nodes (computers) to smoothen the exchange process of various values and media [32,33]. A conventional database, either relational or object-oriented, employs referential integrity methods to maintain satisfactory data validity and synchronization across numerous copies. Conversely, a blockchain is equipped with a consensus algorithm that determines the block orders and relevant transactions before being included in the ledger. The algorithm automates conflict resolutions, such as double spending, by accepting solely one value as a valid transaction [34]. The latest transaction present in the blockchain repository would be authenticated by the entire node before including the record in a pool of pending transactions. All pending transactions would subsequently be categorized into respective blocks by miners, which are computers that classify and publish the transactions on the blockchain by competing with peers to complete a cryptographic puzzle in acquiring the publication priority. As the puzzle difficulty elevates gradually, feigning transactions on the public ledger would be significantly unlucrative with all nodes ably revalidating the transactions upon receiving the latest blocks. Hence, the blockchain validation feature ensures the mining process is highly secure and fraud-resistant, which contributes high integrity to the public ledger. Each block is timestamped and connected to the previous blocks through hashing to create a sequence of blocks attached to the genesis block (the first block in the network). Hashing is a cryptographic technique that converts all information with various file sizes to a fixed output, namely the hash [35–37]. Correspondingly, alterations to the original blocks, regardless of minimal or significant, would result in a different hash [38,39]. Specifically, all stored blockchain data would be subject to several verification stages, which render the information virtually irrevocable and immutable, thus forbidding data tampering even by the initial network entities which publish, process, and store the data. Summarily, the process is performed by miners employing software to decode a relevant cryptogram to publish blocks on the blockchain in guaranteeing network governance [40]. A block is constituted of two main parts, header and body, in which the header contains a previous block hash, timestamp, nonce and Merkle root while the body contains a list of transactions stored in the block. The genesis block is the father of the following blocks in the chain in which it does not contain any previous block hash value nonetheless, genesis block hash is generated by SHA256 hash function, and all subsequent blocks can be traced through it. The previous block's header is hashed via a hashing function in order to be stored in the current block and, thus, blocks are linked, chain growth is increased and simultaneously integrity is preserved against tampering. A timestamp indicates the block creation time, whereas nonce is utilized in the block's production and verification. Merkle tree is a binary tree that contains leaf nodes represented by hashed transactions wherein non-leaf nodes result from the concatenation of its hashed child transactions leading to the production of a Merkle root. Merkle root represents a unique single hash value attained from hashing Merkle tree as the advantage gained to ease transaction verification in the block; however, any amendments performed in any transaction will invalidate its hash and hence, a mismatched Merkle root hash value is denoted [41,42], as portrayed in Figure 1.

An eclectic range of consensus protocols is available contingent on respective network sources, including proof-of-work (PoW), proof-of-stake (PoS), proof-of-activity (PoA), Byzantine fault tolerance (BFT), and hybrid BFT algorithms. Particularly, the PoW protocols function by each node independently decoding the PoW cryptogram to create blocks with authentic transactions, wherein the blocks could be validated in the main chain. Furthermore, honest nodes would frequently mine on the main chain, which was selected based on the protocol rules. For instance, the longest sequence is chosen as the main chain in the Bitcoin blockchain protocol, although the PoW protocol and relevant variants pose exorbitant computation costs and low throughput [43,44]. Contrarily, the PoS blocks are produced by stakeholders [45,46], in which a PoS miner's probability of proposing a block is corresponding to the stake value. The PoA protocol [47] is a hybrid approach

composed of both PoW and PoS protocols, in which blocks are created by stakeholders associated with a pseudo-random series, with their probabilities in the series equivalent to the possessed stake volumes. Meanwhile, each consensual participant in a BFT protocol could suggest an alternative block, such as a common set of transactions, to be consented to by a group of participants [48,49]. The protocol could accomplish a larger transaction throughput, although a communication overhead explosion might be induced. Summarily, every consensus protocol aims to ensure that the most honest nodes could concur on a unified blockchain history.

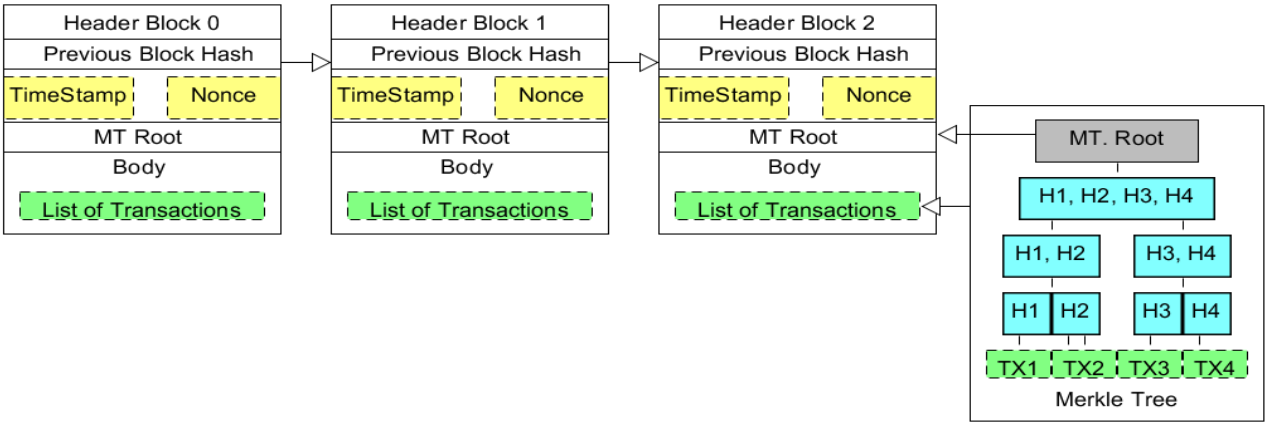

Figure 1. Blockchain construction.

3. Blockchain Construction Features

Different lower-level blockchain technology construct features. In [50,51], decentralization allows the blockchain to serve as a decentralized data repository (ledger) duplicated and disseminated across all users of a specific network. Moreover, neutrality facilitates continuous valid transaction registration in a blockchain regardless of the origination, wherein all individuals with adequate credibility in terms of payments or trust levels could include the transaction records in the public ledger. Concurrently, the replication feature would authorize the consensus models to replicate all valid transactions across all network nodes, while audibility ensures all transactions are publicly displayed and audited due to every performed transaction being appended to the genesis block. Furthermore, integrity is defined as transactions being verified through a hashing algorithm before being published to the distributed ledger. Specifically, the SHA256 algorithm is frequently employed to provide a digital fingerprint which cannot be reverse-engineered [52]. Thus, every amendment could not be executed without invalidating the signature before eventually invalidating the transaction. Anonymity allows users to conduct transactions through pseudonyms generated by respective secret keys rather than actual identities or real-life addresses to prevent exposing personal particulars [53]. Traceability provides detailed records between transactions to trace all transactions in the blockchain and reveal the entire flow of transactions. Simultaneously, immutability issues signatures to guarantee transaction authenticity and integrity, and the Merkel tree (MT) structure maintains an efficient integrity check process, prohibiting malicious block tampering with the hash [54]. As displayed in Figure 2, the transaction TX3 highlighted in red was modified on the nth block, which caused the hash values of the current branch from TX3 to the MT root to become divergent from other nodes' duplicates. Particularly, the header hash of the altered block was also inconsistent with the copy in the (n + 1)th block. As such, the modification was invalid and difficult to be approved without other nodes possessing over 51% hashing power to accept the modification and regenerate an alternative chain from the (n + 1)th block. Resultantly, data immutability is ensured.

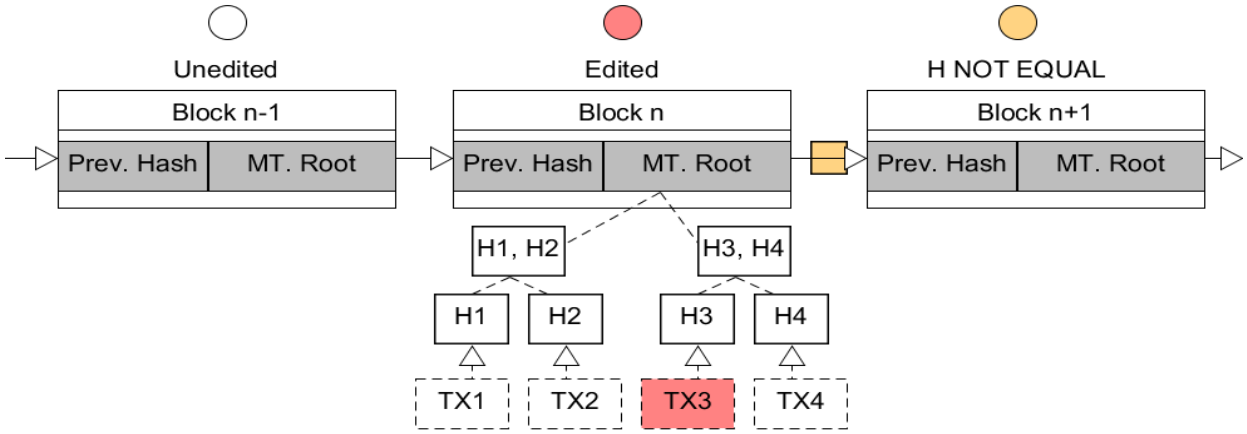

Figure 2. An instance of blockchain immutability.

4. Blockchain Types

Blockchains are categorized into numerous types depending on their respective purposes and distinctive features as illustrated in Table 1. The permissionless or public blockchain does not necessitate platform users to participate in the network [55], which is fully decentralized as the participation is based on the process of consensus and perusal before sharing the transaction record to maintain the distributed ledger [56]. Additional blocks are posted, examined, and verified by all users by possessing a copy of the entire blockchain, which ensures high security in establishment and operation as the blocks are corroborated by computationally challenging consensus procedures, such as decoding cryptograms or investing more personal cryptocurrency. Therefore, any form of data tampering in the blockchain content is prevented by hashes and decentralized concord with other benefits of anonymity and privacy [57]. Nonetheless, permissionless blockchains pose multiple research issues. For example, efficiency is negatively impacted by numerous participants and computationally expensive consensus procedures. Another type is permissioned or private blockchains developed for a sole organization, wherein participants join the network through invitation and are obliged to sustain the blockchain decentralization characteristic [55]. Permissioned blockchains contrast with permissionless blockchains in that only approved entities are permitted to participate in the network and maintain the blocks, which provides higher security and efficiency levels by preventing data tampering through hashes and consensus. Nevertheless, private blockchain nodes are not anonymous [56] as the networks could be breached by internally authorized users. Meanwhile, consortium blockchains serve as private blockchains for various organizations, wherein merely invited and approved users are permitted to participate and support the network. The consensus process is comparatively more time-consuming than in permissioned blockchains, although swifter than in permissionless platforms. In terms of security, consortium blockchains could process data in a more protected manner for averting alterations and hacking activities than permissioned blockchains, owing to the monitoring by different organizations [55].

Table 1. Blockchain types.

Type	Public (Permissionless)	Private (Permissioned)	Consortium
Network	Decentralized	Decentralized Partially	Hybrid Among Public and Private
Access	Any participant	Predefined Participant	Multiple Predefined Participants
Concept	1. Read and write transactions. 2. Vote to add pooled transactions. 3. Validate transactions and consequently they are secured.	1. Conditional read and write operations. 2. Conditional verification. 3. Public read might be allowed.	1. Permission read/write by multi controlling nodes. 2. Controlling nodes selection differ among participated entities. 3. Public read might be allowed.
Consensus	PoW/PoS	Multi party	Multi party
Approval Time	10 Minutes	100 ms	100 ms
Scalability	Slow	Fast, Light	Fast, Light
Security, Privacy	Lack of privacy and anonymity.	High Privacy	High Privacy
Cost	Costive.	Costive.	Costive.
Energy	High Consumption	Low Consumption	Low Consumption
Efficiency	Non-Efficient	Efficient	Efficient
Immutability	Non-tempered	Can Be Tampered	Can Be Tampered
Centralization	No	Yes	Partially
Use Cases	Cryptocurrency	Supply Chain	Banking, Insurance
Application	Bitcoin	Ethereum	Edexa

5. Security Properties

A mature blockchain system should fulfill three security attributes established [58,59]. The security properties guarantee blockchain accuracy, consensus, and validity with a high probability, whereas redaction designs must not impact them. They are classified as follows:

- **Chain Quality:**

It refers to the ratio of the blocks controlled by an adversary in a chunk of an honest party's chain that cannot exceed a certain fraction X where it represents the total amount of adversary-dominated resources.

- **Common Prefix:**

Formally common prefix property represents two honest node chains S1 and S2, whereas the shortest chain is a common prefix for the longest chain at different time segments T1 and T2. The common prefix $K \in \mathbb{Z}$ indicates the number of blocks required to be removed from a timely older chain. If $S1 \leq S2$ where $T1 \leq T2$, it means removing k blocks at the end of S1 and becoming a common prefix to S2.

- **Chain Growth:**

The growth of an honest chain must be compatible with the number of blocks produced regardless of adversarial employed methodology.

The security properties guarantee blockchain accuracy, consensus, and validity with a high probability. Specifically, accuracy maintains the "healthy" degree of a redactable

blockchain, while consensus necessitates all honest nodes to concur on a particular sequence. Meanwhile, validity specifies that most blocks emanate from honest nodes [59]. A redaction technique is required to not impact the aforementioned blockchain security properties.

6. Redactable Blockchain Implementation Challenges

Redaction contradicts the principle of blockchain immutability, thus requiring the process to be conducted under stringent rules. Accordingly, the redaction procedure is restricted to authorized personnel and contingent on specific situations without compromising system reliability and constancy. Contemporarily, multitudinous blockchain data redaction approaches and models are developed by suggesting different pertinent methods. Similarly, every relevant redaction approach should prioritize validating all redacted transactions or blocks. Figure 3 presents several redaction challenges to thoroughly investigate and differentiate every proposed model and they are as follows.

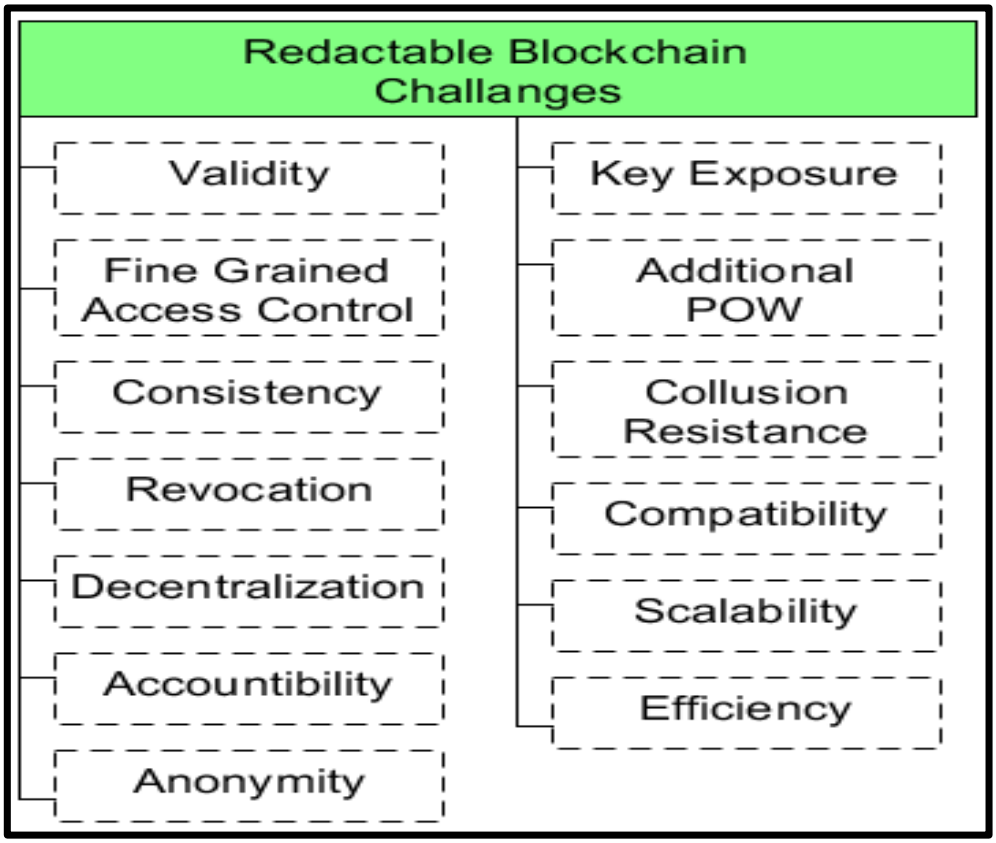

Figure 3. Redactable blockchain implementation challenges.

6.1. Validity

Consensus protocol approved policy must be met and applied by redacted transaction/block to acquire validity. Once redaction is verified, then it is considered honest by other nodes.

6.2. Consistency

Authentic nodes observe the chain consistently. Connected transactions/block validity must not be affected by redaction at any period. Thus, avoiding consistency and traceability damages at some points caused by honest redaction.

6.3. Fine Grained

Who has the right to modify it remains an unsolved question. Fine-grained access control authorizes various rights to diverse users to modify contents, however coarse-grained access control cannot identify who made the certain modification.

6.4. Decentralization

Centrality remains a bottleneck, which has been solved by decentralization through transferring power and control to users; consequently, system security is improved as consistency and corruption are fairly reduced.

6.5. Accountability

Redacting transactions by authorized users requires certain accountable mechanisms to observe their activities.

6.6. Revocation

Access control methods play a vital role in controlling users' privileges where they must act effectively once the user's rights are revoked, left, or degraded without affecting system efficiency in terms of cost and computation overhead.

6.7. Anonymity

Anonymity aims to achieve privacy to protect transaction contents via authentication and preserve personal identifications of modifiers from any adversarial activities.

6.8. Collusion Resistance

Collusion is the illegal accumulation of rights of cooperated parties who aim to gain unlawful access to data and the system must prohibit these actions.

6.9. Scalability

User amount growth must not affect system performance.

6.10. Efficiency

Time is the efficiency factor in admitting/ratifying certain redactions.

6.11. Computation Overhead

Low cost must be achieved while applying all the above measures.

6.12. Additional PoW

Redaction mechanisms at the block level might be required to resolve PoW puzzles again, and hence it leads to being undesirable to miners due to its high hashing power consumption.

6.13. Key Exposure

It is an obstacle that prevents CH from being used. Trapdoor keys can be disclosed by finding several collisions for a certain function.

7. Redaction Mechanisms in Blockchain

This section represents comprehensively the taxonomy of two main different categories, namely chameleon and non-chameleon-based redaction mechanisms in blockchain, which involves analyzing emerging challenge-related issues in order to produce feasible perception. Figure 4 shows a detailed discussion in the following subsections.

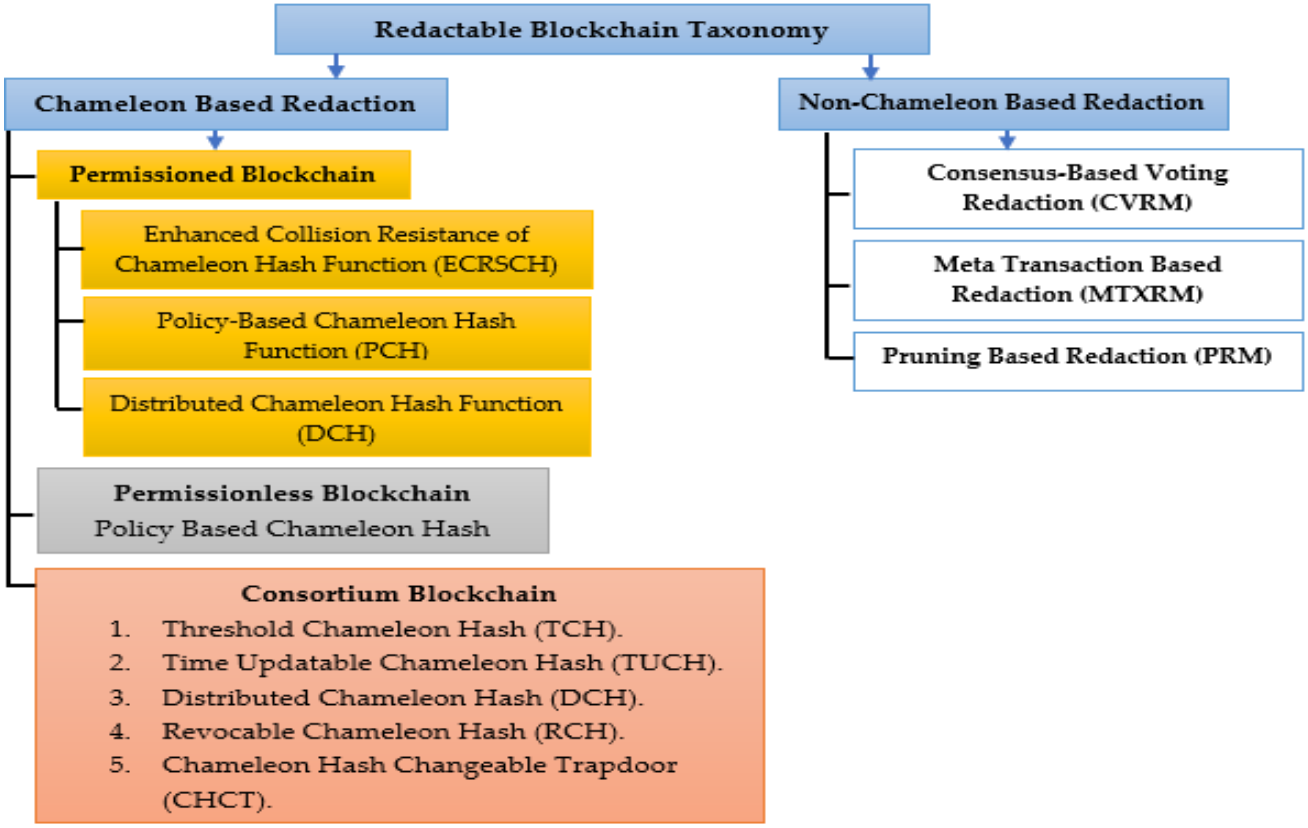

Figure 4. Paper taxonomy analyzing redactable blockchain state of art.

7.1. Redactable Blockchain-Based Chameleon

Redactable blockchains leveraging chameleon hash functions, which contain a trapdoor key, function as a standard cryptographic hash under conventional circumstances. A chameleon hash-based redactable blockchain is collision-resistant, which allows the trapdoor key holder to effortlessly perform data amendments by seeking the same hash as the original version. The holder could be either a centralized entity or belong to a defined set of independent entities. Summarily, a chameleon hash function allows the following procedures to be executed: (i) each function contains a couple of trapdoors and hashing keys, (ii) individuals with hash keys could create the hash function, (iii) trapdoor key holders could locate collisions in the function domain, (iv) the function is collision-resistant for individuals without trapdoor keys [60]. Table 2 provides an overview of all proposed methods discussed below.

Table 2. Chameleon redaction methods outlines.

Schema	Methods	Security Assumption	Setting (Blockchain Type)	Redaction Type
[61], (2017)	CH and PKE	DLP	Private	BL
[62], (2020)	CH and ZR	SXDH	Private	BL
[63], (2020)	CH and ZR	SXDH	Private	BL
[64], (2019)	CH and ABE	DLP	Private	TL
[65], (2020)	CH and ABE and DS	DLP	Private	TL
[66], (2021)	CH and DS	DDH	Private	TL
[67], (2022)	CH and ABE and DS	DLP	Private	TL\BL
[68], (2021)	CH and MA-ABE and DGS	DLIN	Private	TL
[69], (2021)	CH	DLP	Private\Public	TFL
[70], (2021)	CH and ABE and DS	DLIN	Private	TL
[71], (2021)	CH and MA_ABE	DLP	Private	TL
[72], (2022)	CH and ABE and DS	DLP	Private	TL
[73], (2021)	CH and ABE	DBDH	Private	TL
[74], (2021)	CH and ABE and DS	DLP	Private	TL
[75], (2021)	CH and TEE	DLP	Private	BL
[76], (2021)	CH and Lattice	SIS	Private	BL
[77], (2022)	CH	DLP	Private\Public	TL
[78], (2021)	CH and MA_ABE and DS	DLP	Public	TL
[79], (2019)	TCH and DS	CDHP	Consortium	TL
[80], (2019)	RCH	CDHP	Consortium	TL
[81], (2020)	TTCH	CDHP	Consortium	TL
[82], (2020)	CH	DLP	Consortium	TL
[83], (2021)	TCH and DS	CDHP	Consortium	TL
[84], (2021)	TCH and DS	CDHP	Consortium	TL
[85], (2021)	CH	DLP	Consortium	TL
[86], (2022)	CH	CDHP	Consortium	TL

7.1.1. Redactable Permissioned Blockchain

According to the taxonomy, permissioned blockchain redaction mechanisms have been divided into three main sets in which are analyzed and discussed thoroughly as follows:

- Enhanced Collision-Resistance of Chameleon Hash Function (ECRSCH)

Ateniese et al. [61] have proposed the first redactable blockchain with an enhanced standard chameleon hash function (ECRSCH) by incorporating chameleon hash function (CH), public-key cryptography (PKE), and non-zero knowledge proof (NIZKP), which is adopted by Accenture. Every block structure in the blockchain will extend to include an additional field to efficiently record the randomness (r), also termed as a chameleon hash check value, whereas solely trapdoor key holders consequently are able to compute (r′), as the computation of additional redacted block (block *n*) randomness (r) is highly challenging without a trapdoor key. The randomness maintains the identicality of redacted blocks and header hashes (prior hashes) as before reduction to ensure the blockchain is intact. Figure 5 represents Ateniese's proposal. The approach is limited as the procedure is constrained

to rewriting the blockchain based on the block level where it impacts system security and efficiency regardless of mechanism employed, whereas chameleon hash function is utmost risky where the whole block privacy depends on a single trapdoor key for all the transactions maintained; once a modifier granted permission to alter any transaction, the rest can be modified too. Simultaneously, the procedure diminishes transaction validity in the blocks, including other relevant transactions. In addition, coarse-grained decision-making in the function does not define individuals entitled to compute the collision when the hash is frequently produced by the public key.

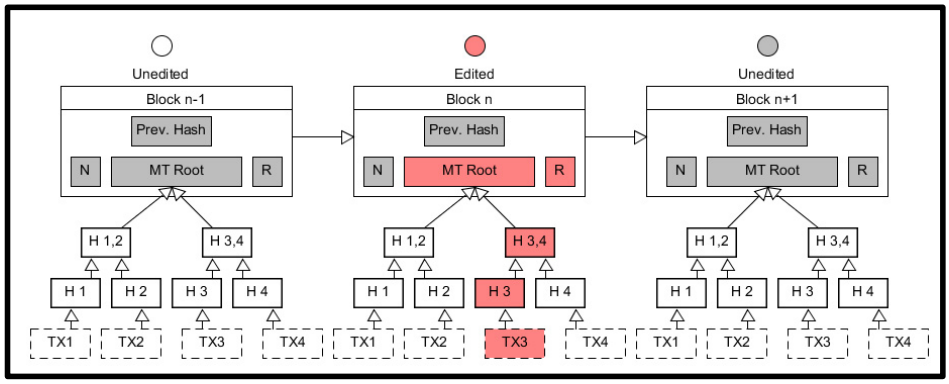

Figure 5. Ateniese's proposal: randomness update in redacted block (edit).

Conversely, secret keys are employed to search for collisions without hash makers being aware of the individual performing the collision. Meanwhile, block traceability is not supported, and accountability is excluded from the chameleon hash function, thus posing the possibility of malicious modifications. With the presence of the trapdoor key and centralized storage, distributed management is not feasible due to the missing decentralization characteristic. Subsequently, Khalili et al. [62] and Derler et al. [63] enhanced the collision resistance feature of the CH corresponding to the Ateniese proposal [61]. However, better efficiency is achieved while similar construction is utilized. Both proposals sought trapdoor exposure prevention; meanwhile, they still suffer from similar Ateniese's proposal drawbacks.

- Policy-based Chameleon Hash Function (PCH)

Authority regulation to redact data in a fine-grained model, Derler et al. [64] proposed the policy-based CH function (PCH) as shown in Figure 6.

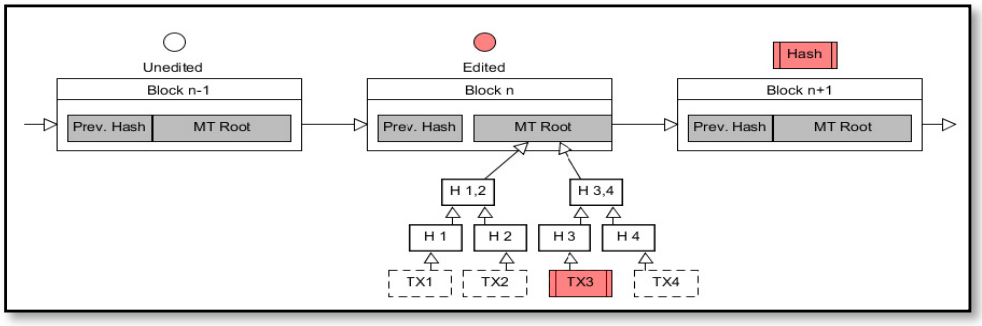

Figure 6. Derler proposal.

It is inspired by the CH with an ephemeral trapdoors (CHET) function [87], and cyphertexts policy attribute-based encryption (CP-ABE) [88] to serve as a hybridization function. The PCH function is associated with the user's attributes to control redaction privileges. The present study fulfilled two main questions by determining the individual authorized to perform a redaction and the specific information that could be redacted.

Specifically, users of private blockchains are permitted to perform redaction activities as their participation is authorized before becoming involved in the network. The CP-ABE provides the PCH function with high indistinguishability, collision, and resistance levels due to the security measures from indistinguishability under the non-adaptive and adaptive chosen cyphertexts attack (IND-CCA2), which prevents attackers from recovering trapdoor keys in conducting double-spending and tampering attacks. Meanwhile, the PCH chain allows transaction-level (TX-level) redaction without negatively impacting the system security, although collusion attacks are highly possible between participants by unifying personal attributes and updating or sharing access to an MT to be consistent with a specific strategy. Nonetheless, non-traceable evidence modifications are a security issue as the changes allowed by the PCH function would not be visible to other users. ABE is not guaranteed to encrypt valid one-time trapdoor keys without NIZK inclusion. The comprises of CHET base RSA in PCH is ineffective in terms of cost and time overhead. Other limitations of Ateniese's proposal [61], including non-accountability, distributed key management, trapdoor centralized storage, and central control of secret keys wherein decentralization remain unresolved, as in Figure 6 showing a delineated Derler's proposal [64]. Tian et al. [65] propounded accountability to be the main contributing solution by proposing the policy-based CH function with black-box accountability (PCHBA) by combining the CHET, the CP-ABE, and the digital signature. The PCHBA is advantageous in prohibiting a transaction modifier to perform malicious rewriting with the accountability property to halt any exploitation form of rewriting privilege. Nonetheless, the PCHBA violates user anonymity as the function supports key generation by central authority, while privilege revocation and decentralization are not considered. PCHBA deprives an effective penalty mechanism to stop malicious redaction activities. As such, Xu et al. [66] proposed the k-time modifiable and epoch-based redactable blockchain (KERB), which consists of a monetary penalty to regulate rewriting privileges and punish malevolent demeanors. The proposal improved the [64] algorithm by ensuring sufficient accountability and traceability to account for the lack of punishment mechanisms and negligence of collusion attack possibility. The KERB comprises the central authority (CA), the miner, and the modifier. The modifier sends a key creation request, which includes the transaction to be modified, the required modification, and the key expiry time. The CA approves both the modifier's identity and request by signing the request with a conditional deposit to produce a token key before being sent to the modifier. A violated modifier triggers punishment by withdrawing agreement on the deposit. Meanwhile, it is committed to any conducted modification approvals wherein modifications are not stored immediately to facilitate checking by the miner. Only approved transactions will be stored in the blockchain public ledger. Nevertheless, security weakening, scalability and flexibility are less common due to CA centralization. Concurrently, computation overhead and malicious modifiers' privileges are not minimized as the deposit amount might be smaller than the damage cost. The authors in [67] have followed the previous suggestion by proposing a blockchain rewriting approach (CDEdit) that allows modifiers to acquire controllable redaction privileges and diversified editing types based on [66] frameworks. The CDEdit model, which is identified as a privilege token service (PTS), authorizes multilevel editing in on-chain permissioned blockchains, such as Hyperledger, to manage and disseminate redaction prerogatives. The entire process is triggered by modifiers upon sending a rewriting request. The modifiers are divided into four main categories, including one-time modifiers on the transaction and block level, and multi-time modifiers on the transaction and block level, without producing conflicts. Specifically, the process commences with requesting a token before receiving a token key (TK) which comprises the request of required rewriting times, expiry times, object indexes, and modifier public key (MPK) issued by the PTS. Before sending the TK to the modifier, a deposit is necessary to account for potential malicious actions after the PTS validates and signs the modifier's request. The PTS remains vulnerable due to centralization, which would negatively affect scalability with the presence of latency and computation overhead, and the absence of an effective punishment mechanism. Panwar et al. [68] adapted the [64]

model by developing a revocable and traceable blockchain rewriting framework (ReTRACe) from the revocable CHET (RCHET) and revocable attribute-based encryption (RABE). Two primary dissimilarities exist wherein the RCHET hashing algorithm is a private procedure permitting only trapdoor holders to generate the CHs and subsequently conduct relevant adaptation, hence being inconsistent with the CH definition. Meanwhile, the hashing algorithm is intended to be a public procedure in the RABE to allow attribute revocation. Decentralization is attained via replacing ABE with MA-ABE; nonetheless, a public RABE is unrealistic as a user is required to acquire another decryption key through a secure medium when the user's attribute is repealed by the attribute authority (AA), although the group manager's (GM) attributes could not be revoked. Subsequently, Jia et al. [69] developed a fine-grained blockchain rewriting mechanism to support hierarchical revocation via two stages. The first two data types in a mutable transaction, which were personal information (required to be recorded in a chain) and security-related details (senders, receivers, and transaction amounts), were considered. Users would be authorized by a semi-trusted regulator to amend existing personal records and modify security-related data. As the entire process concentrated on certain mutable transaction fields, the procedures would be more precise compared to the block-level and transaction-level blockchain redaction. Simultaneously, the regulator possessing a master key could repeal a user's rewriting privilege by generating an alternative subkey to disable the original key. Notwithstanding, the model did not establish a decentralization concept or indistinguishability. Xu et al. [70] established a revocable policy CH function (RPCH) adapted from the PCH [64] to aid in fine-grained and revocable blockchain rewriting approaches. The RPCH development referred to CHET based on the Rivest, Shamir, and Adleman (RSA) [89] proposal with the RABE method [90], and the fast attribute-based message encryption (FAME) process [91]. The RPCH also did not include a blockchain decentralization feature in its design via permitting a trusted party to practically repeal a chameleon trapdoor possessor's redaction prerogative, outsource the decoding process, and transmit confidentiality without another semi-trusted entity however CHET performs based on RSA, which affects scalability issues. Meanwhile, Zhang et al. [71] and Ma et al. [72] proposed a multi-authority policy-based CH (MAPCH) function, wherein the CA in the [64] model is replaced with multiple authorities to manage the attributes applied to a permissioned blockchain. The performance in both proposals has been improved due to the decentralized technique proposed, however the reliance on RSA remains a scalability drawback and the absence of revocation requires further investigation. Guo et al. [73] suggested a hybrid blockchain rewriting process both online and offline via an auditable outsource computation (OO-RB-AOC) scheme, which was adapted from [64]. The hybrid model is an enhancement to produce two PCH types, namely online and offline. Particularly, high-value transaction operations could be performed offline to provide adequate credibility by incorporating multiple authorities to generate rewriting keys. Beneficiaries who acquired the keys would be verified on whether the keys were issued by legitimate authorities. The previous model has been built based on ring signature [92] and is employed to create a rewriting key in which external resources could be continuously applied due to the limited device capacity. Nevertheless, the limitations include the lack of a pertinent mechanism for revocation and misbehavior observation. Meanwhile, Hou et al. [74] established a fine-grained and governable redactable blockchain model to allow destructive information forced erasure. Specifically, the transaction creator possesses a delicate regulation over selective individuals to conduct data redaction by specifying the components to be edited, after receiving sufficient miners' votes. All miners could also disclose the block index with detrimental data generated by malevolent users to be subsequently recompensed after authoritatively erasing the deleterious information based on the index. Malevolent users would be prohibited from performing further transactions if the penalty was not paid as miners' compensation within a stipulated period. Moreover, the fine-grained framework facilitates additional information and unexpended transaction output (UTXO) redaction concurrently. Nonetheless, revocation is unavailable

for any misdemeanors and decentralization is not recognized with the presence of a trusted authority, significantly influencing the model's performance.

- Distributed Chameleon Hash

Liu et al. [75] have proposed the inclusion of the distributed key management function by using distributed CH function and trusted execution environment (TEE) [93]. Distributed key management is employed to guarantee high trapdoor security levels, which could be strengthened by the TEE to elevate asset confidentiality and integrity degree in a secure TEE enclave. Nevertheless, as the rewriting procedure is performed throughout the blockchain level, weak security is contributed by the exclusion of collusion, accountability, and tractability. Wu et al. [76] further improved the existing CH function. In the present study, the development was based on enhancing the collision resistance characteristic which was perceived to be highly vital to managing redactable blockchain. Particularly, quantum-resistant key-exposure-free CH functions via the concrete single trapdoor were the foundation to optimize the existing CH function based on the proposed framework. Two operations were included, namely distributing trapdoor keys on a semi-honest secure part and voting on the redaction to provide public accountability. However, collusion is not prevented. Matzutt et al. [77] proposed Redact Chain, which comprises several parallel local jury committees that are rotated periodically to perform redactions instead of utilizing external tools. The committee includes solely authorized members who can conduct redactions jointly by employing the CH function and threshold cryptographic method. Each committee will be located at a randomly selected node to act immediately when certain content is reported. Disputed transaction cases are solved by the jury committees voting in an off-chain mode to approve the modifications. The CH function is applied to grant redaction authority while limiting the possessed authority from being abused. Keys will be distributed amongst the jury members respectively to equalize the trapdoor key control. All modifications could only be conducted once before the committee is replaced by another batch selected from recently successful miners. As such, Redact Chain supports global redaction to organize respective mining activities. Nonetheless, attackers could dominate the jury committee as no accountability mechanism is enforced on the committee.

7.1.2. Redactable Permissionless Blockchain

Developing a redactable public blockchain protocol is challenging yet engaging as the accomplishment requires highly sophisticated solutions. Correspondingly, researchers in [65,74] stated that their proposed redactable mechanisms would be realized in the permissionless blockchain as their future work. In 2021, Tian et al. [78] created an enhancement protocol by expanding the current blockchain rewriting ability with a fine-grained access control policy in permissionless blockchains, including Bitcoin and Ethereum, which required no trusted parties to provide access privileges. To achieve strong security, dynamic proactive secrete sharing (DPSS) as a decentralized set of procedures were implemented to remove all forms of trusted parties, while a committee of several users was established with each user holding a segment of trust. All users are authorized to join the committee whenever available to utilize the KP-ABE to safeguard fine-grained access control. The previous author proposes another algorithm that contains two major phases of fine-grained and public traceability. Subkeys are divided from the master key and distributed amongst committee users, which ensures all committee users provide unanimous agreement before granting access privileges requested by modifiers through master key recovery. Any shift in the committee structure would not affect the master key as the algorithm remains intact. Subsequently, public traceability, digital signature, and the KP-ABE, which constitute the second phase, necessitate the modifier to sign a modified transaction with the private key to guarantee the modification is publicly verifiable. The public traceability function is enabled by the attribute-based encryption traceability (ABET) through a set of rewriting privileges produced from the interaction with the access black box to preclude unauthorized access. Simultaneously, the presence of the public key proves that the modifier was authorized to rewrite a block by the committee. Nonetheless, committee members' collusion behav-

ior could not be effectively averted. Similarly, security issues remain present from the high flexibility provided to the modifier and the lack of effective misbehavior punishment mechanisms and revocation methods.

7.1.3. Redactable Consortium Blockchain

Huang et al. [79] demonstrated that [61] blockchain protocol was not optimally applicable to the conventional industrial IoT (IIOT) scenario and, therefore, a redactable consortium blockchain (RCB) was established to increase redacted transaction validity. The RCB concentrates on forfeiting transaction redactions when modifiers are offline while upgrading the CH function as threshold CH (TCH) through linkage with participants' identities (authorized sensors). Authorized sensors with genuine identities could effectively calculate additional randomness assisted by holders with trapdoor keys. By adhering to the hash-and-sign model, the TCH function hashes.

The transactions before being endorsed by an accountable-and-sanitizable chameleon signature (ASCS) algorithm [94] as portrayed in Figure 7. Participants with authentic identities could also assess the same signature to authenticate redacted transactions. Nevertheless, the scalability issue exists owing to the high cost of generating trapdoor key holders' pertinent signatures, which increases proportionally to the number of nodes. Redaction power exploitation could also occur concurrently. Huang et al. [80] developed smart-chained redactions supported by the self-redactable blockchain proposal (SRB), which required no cooperation to perform transaction redaction in the blockchain. The revocable hash function (RCH) optimizes the CH function by linking identities within a cyclic time and being relied on by the revocable chameleon signature (RCS) to generate transactions. Participants are only allowed to perform periodic redactions based on the defined cyclic time in the blockchain. Specifically, private redactions are conducted with trapdoor keys contained, and (X-1) public redaction only if any X transactions are attached beyond the performed redaction. Meanwhile, public reaction is conducted through the signature to facilitate every individual performing transaction authentication. When a trapdoor expiration period is stipulated, the trapdoor will be updated periodically. Nevertheless, due to high cryptographic complexity, the SRB is rarely adopted in IoT devices. In addition, low adoption is due to the SRB and the RCB sharing a fixed expiration period, which renders the trapdoor key management inflexible while allowing the old trapdoor to be continuously utilized. Hence, key exposure would pose security issues. Zhang et al. [81] proposed the reusable and redactable blockchain (Re-chain) for the RCB, which was a redactable and reusable blockchain established by the threshold trapdoor CH (TTCH) function to address blockchain storage issues in accounting for information explosion. The proof-of-concept consensus was also realized when Re-chain achieved maximum storage size to rewrite previous blocks with security and authority, while maintaining block connection safety. Consensus guarantees robustness even with the presence of N-node acceptance (N/2-1) faults in the edge nodes. Furthermore, Re-chain is beneficial for the IIOT which frequently encounters serious storage limitations. Past experiments revealed that the rewriting process was efficient when performance achieved a medium scale within the limit of 30 nodes. Nonetheless, the deficiency of scalability due to high computation overhead would not support user accountability and anonymity on the platform. Time could be defined as a physical entity in terms of hours and days, or as a virtual being in terms of the number of blocks in the blockchain. As such, time is the main factor in generating an efficient key to redact and generate transactions. Correspondingly, LV et al. [82] suggested a decentralized CH function to gain enhanced security, time efficiency, and reduced space overhead in offering higher RCB flexibility. Modifiers would be necessitated to sign redaction requests to be validated by authorized nodes on personal identities by comparing the signatures and redaction requests. The requests are stored in modified transactions while the signature key is stored in the local redaction record. Thus, accountability is maintained as redaction records could be compared with the stored signatures. Nonetheless, an additional storage cost and low anonymity would be the issues. Huang et al. [83] improved the previous work

of [79] by elevating the anonymity and scalability degrees with the time-updatable CH (TUCH) and linkable-and-redactable ring signature (LRRS) as theoretical fundamentals to develop a decentralized, scalable and redactable algorithm with sufficient anonymity similar to the SRB. The TUCH enables automatic ring configuration without interacting with other entities when producing an access key. As the chameleon randomness in holding a hash collision is subject to cyclical expiration, redaction could fulfill verification criteria only within a pre-determined period. Meanwhile, the LRRS ensures modifiers anonymously provide personal signatures on a message without trusted parties. The process is completed by forming a ring of users automatically, without the users being informed that their public keys are employed in producing the signature. Nevertheless, increasing scalability would lower the anonymity level owing to the requirement of the signature, which simultaneously renders the process to be costly. Conversely, Gao et al. [84] criticized the [79] TCH proposal after manifesting that the TCH was vulnerable to malicious redactions. Correspondingly, an enhanced TCH was established to elevate the collision n-resistance degree, which was recognized as one-more preimage resistance. The enhanced protocol ensured sufficient resistance even when preimages with the same hash values and identities were exposed. Nonetheless, similar to the RCB, the protocol was seldom adopted in IoT devices due to high cryptographic complexity. Security issues, such as collusion and scalability, were not adequately considered. Zhang et al. [85] discovered that credible industrial blockchain data management could optimize blockchain reducibility by enhancing public key generation through the CH function in three phases, namely off-chain setup, on-chain setup, and data management. During the setup different data types are included in the blockchain where trapdoor partial authenticity verification is conducted and aided by smart contracts to determine accountability levels. Particularly, the off-chain setup deploys blockchain parameters to create trapdoor segments and perform trapdoor distribution to support trapdoor holders. The on-chain setup is executed by employing the redactable blockchain to collect various data before gauging the authenticity degree. Subsequently, data management separation from other transactions is performed by two different blockchain approaches to realize management supervision on transaction redaction, while reducing coupling in the blockchain-based trapdoor recovery accountability mechanism. Smart contracts would verify the validity of trapdoor segments published by the holders, which are concurrently monitored by supervision nodes before jointly approving editing requests. Trapdoor performers would redact information upon receiving approval. Nonetheless, the entire process is costly, which is unsuitable for the IIoT environment. Accordingly, Wei et al. [86] proposed the federal learning model to be applied in the IIOT circumstances, such as the medical blockchain (RMB), by incorporating the CH changeable trapdoor (CHCT) function. When collisions are identified, the CHCT would be updated to govern the process without revealing the trapdoor. The CHCT could be canceled in in all circumstances to avoid key exposure owing to the highly restricted trapdoor contributed by the owner's autonomy to define the expiration time.

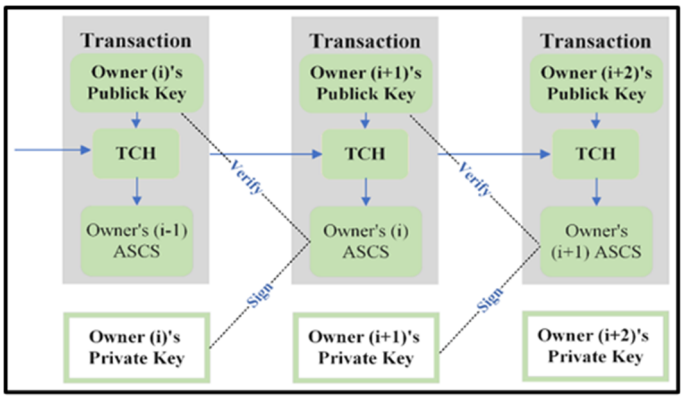

Figure 7. The Huang et al. (2019) RCB signature chain.

7.2. Non-Chameleon Hash-Based Redactable Blockchain

In this section, non-chameleon hash-based mechanisms have been thoroughly discussed and further analyzed as illustrated in Table 3 in terms of methods, security assumption, blockchain type, and finally redaction level type.

Table 3. Non-chameleon redaction approaches outlines.

Schema	Methods	Security Assumption	Setting (Blockchain Type)	Redaction Type
[95], (2019)	CVRA	N/A	Public	TL
[96], (2020)	CVRA	N/A	Private/Public	TL
[97], (2017)	MTRA	N/A	Private/Public	TL
[98], (2020)	MTRA	N/A	Private	TL
[99], (2019)	PRA	N/A	Public	BL
[100], (2020)	PRA	N/A	Public	BL

7.2.1. Consensus-Based Voting Redaction Approach (CVRA)

The CVRA leverages standard regulations to concur on the eventual condition of redacted transactions or blocks via on-chain voting. A redaction would only be approved when all honest nodes consent to the altered state. Past studies [101] employed the hard fork to amend the information history, resolve crucial security risks in codes, include additional functionality, restore previous transactions, and overcome negative hacking consequences. For instance, Ethereum applied the hard fork to restore past transactions in recovering substantial funds lost to "The DAO" attack. When a fork simultaneously appears in at least two block versions of the same height, an alternative chain will be produced. Although accidental forks could be resolved via identical consensus principles, attackers' chains are more difficult to be removed due to the longer series than the honest version (51% attack). Consequently, the hard fork adhering to the latest consensus principles would generate an irreversible sequence independent of the original chain, which is typically employed to divide blockchain cryptocurrency. For example, Bitcoin Gold and Bitcoin Cash were separated from Bitcoin [102]. Nonetheless, several limitations exist as the hard fork would not erase exact data history as both original and alternative chains would be maintained by respective nodes. Moreover, a high cost is incurred as the procedure consumes significant computation bandwidth.

Deuber et al. [95] suggested authorizing data redaction in the public blockchain with the CV (consensus voting) sequence expanding the block header structure to provide additional fields encompassing previous states or hash values, which enabled connectivity between redacted and subsequent blocks. All blocks are connected by two hash chains to represent the latest prevHash and prior hashes, with the following block constantly being attached to the previous block after referring to the prior condition. For multiple redactions, the old state consists of all states of all revisions. Moreover, the CV series adequately sustains consistency through honest nodes, holding a unified monitor of redaction activities by publicly validating before voting. The redaction request on a specific transaction would be announced in the peer-to-peer (P2P) network, where miners securing a subsequent block could vote for approving a valid revision while recording the redacted block hash in personal blocks. The revision would be approved when the editing request on a particular block received sufficient votes. Resultantly, honest nodes would update personal copies with the performed redaction. Scalability and efficiency degrees are low in the process owing to the significant amount of time required for approval of valid redactions. Marsalek et al. [103] formulated an amendable blockchain protocol in two hash chains, wherein the standard chain stored the original data, whereas the correction chain comprised numerous blocks to store correction data. The CV approach was also applied to perform a decentralized decision on requested corrections, in which a correction block consisting of the

actual correction data replaced the redacted block with the same height in the standard chain after the voting session. Nevertheless, several challenges persist, including a long voting period and poor redaction efficiency. Meanwhile, Thyagarajan et al. [96] proposed the Reparo approach to conduct redactions on a repair layer in both public and private blockchains. Particularly, redaction would be sought through a repair message to specify the block hash, the object, and the chain state to be amended in an off-chain for validation. A repair witness would be voted on-chain by miners. Similar to the procedures in the CV series, the witness obtaining adequate votes would be approved to maintain transaction consistency by Reparo recording the original version and the approved repair message in data repositories. The repositories would also be announced and managed by the P2P network nodes. Furthermore, the repair message requires monetary expenses to be stored in the blockchain. Limited storage size, extended voting time, large bandwidth requirement, and computation overhead contributed to low process efficiency.

7.2.2. Meta-Transaction-Based Redaction Approach

The meta-transaction-based redaction approach (MTRA) is regulated by fiat and mate-transaction (mate-T), which is an alternative type to authorize redaction activities while preventing heavy cryptographic primitives and an extended voting period. Puddu et al. [97] recommended extracting coins from performed transactions, namely μchain, to enable blockchain data redaction by providing additional meta-T as mutants and extending transactions. The process commences with μchain separating the entire transactions, such as currency and smart contract deployment transactions, into mutable and immutable versions. The mutable version would be identified as active in each stage while the redaction would be subject to the stipulated rules delineating the redacted objects, the redactor, and the time window. The redactor would produce the meta-Tx to activate redaction procedures. When the extending transaction is permitted in an efficient off-chain voting method, the mutant transaction would substitute the active transaction. Subsequently, μchain encodes the mutable transaction to conceal the different transaction history. The secret key is distributed via the DPSS protocol [104] to fulfill the security threshold. Nevertheless, μchain violates consistency owing to high computation overhead and bandwidth in the enciphering procedure. Blockchain transparency is also diminished as limited verification operations would engender low audibility. Dorri et al. [98] established flexible memory management in the IoT, namely the memory-optimized and flexible blockchain (MOF-BC), to allow certain blockchain transaction deletion or summarization within a period in extensive networks. Concurrently, the deleted and previous transaction hashes would remain to guarantee record consistency through the redactor's signature to corroborate that personal editing legitimacy MOF-BC records essential data about the summarized transactions, including the MT root, timestamp, and order. Nevertheless, the CA verification is not robust owing to key management ineffectively preventing denial-of-service (DOS) and double spending attacks. Florian et al. [99] proposed the functionality preserving local erasure (FPLE) to delete improper information of transaction outputs in the local node storage as transaction outputs, including arbitrary information, would be improbably further employed, which enables rewriting in the local UTXO database. To cautiously delete the data, the FPLE adjusts the transaction output, which was the script public key (ScriptPubKey), to develop a removal database to store relevant data before performing redaction, while inspecting unedited inputs. The database could also validate the following transactions which refer to the amended records. After inspecting and validating, the original record would be substituted with the amended block before being completely removed. Meanwhile, drawbacks are present due to the unguaranteed credibility of the removal database source and a trusted entity is necessitated to manage the database, which violates decentralization and security principles.

7.2.3. Pruning-Based Redaction Approach (PRA)

The PRA is devised to be applied in specific situations in circumventing massive cryptographic primitives. Pyoung et al. [105] established two different PRAs to allow more edge node storage space in the IoT ecosystem. As edge nodes possessed finite amounts of storage and computing power in the IoT ecosystem, memory would be swiftly depleted and complete blockchain generation would be impeded. Accordingly, the blockchain LifeTime (LiTiChain) resolves the limitations by removing expired blocks, in which edge nodes achieve a distributed concord based on the BFT algorithm [106] through the corruption tolerance threshold (t) fulfills $n \geq 3t + 1$ for n nodes. Each LiTiChain block persists from the creation time until the stipulated end time and would be removed from the chain when the block lifetime expires. When an obsolete block consists of unexpired transactions, LiTichain restores and attaches the unexpired blocks to the latest block. Subsequently, LiTichain formulates a graph congruent with the block end-time order, wherein the child block possesses a shorter lifespan than that of the parent block. To maintain block consistency, every block header comprises two reference hashes, namely the prevHash of the former block and the Parent BlockHash of the parent block, in the graph. Although certain blocks would be erased, the remaining blocks remain in connection with one another. Removing expired blocks would minimize the percentage of honest blocks in the chain, while compromising the chain quality and common prefixes. Moreover, LiTiChain could not effectively sustain block consistency as the deletion process would negatively impact previous and upcoming transactions, which might pose security vulnerabilities. For instance, malevolent nodes could effortlessly perform double spending and transaction forgery on blocks legally removed from the series. LiTiChain is also vulnerable to the DoS attack as the removed block hashes would not be stored, therefore leading to low chain traceability and integrity. Matzutt et al. [100] suggested a snapshot method, which was CoinPrune, to minimize blockchain size in optimizing the storage and bandwidth criteria. CoinPrune generates a snapshot for constant intervals, such as every 10,000 blocks, which contains block headers and serialized UTXOs in a group of pruned blocks. The snapshot is openly broadcasted to be authenticated within a period. If the authentication process exceeds the stipulated threshold, the snapshot is regarded as genuine and would be subsequently accepted. Contrarily, the snapshot is considered invalid, and the pruning process would be postponed when the threshold was not satisfied. The latest joining nodes necessitate only the validated snapshot and several complete blocks to be concurrent with the system. CoinPrune could decrease the required device storage and synchronization period, although continuous pruning could render a costly overhead. As data are not entirely removed from the blockchain, CoinPrune enables pertinent resolutions in preventing certain nodes from storing improper information in local devices. The CoinPrune nodes generating the blockchain via the snapshot technique are comparable to the lightweight counterparts in requiring complete nodes to store the entire blockchain. As such, blockchain consistency and security are ensured when decentralization is facilitated with all nodes composed of a low network percentage. Resultantly, pruning is a feasible approach to performing redaction, which serves as a high-interest research topic.

8. Comparisons and Discussion

Blockchain data redaction studies have been started by using the [61] proposal, which has been the core for the aftercoming suggestions. This section is dedicated to comparing applied implementation challenges and security properties for previously analyzed literature respectively for chameleon based, as illustrated in Table 4, and non-chameleon redaction mechanisms in Table 5. The discussion below illustrates the advantages and disadvantages of the proposed methods in both the chameleon and non-chameleon approaches analyzed above.

Table 4. Chameleon redaction mechanism challenge comparisons.

	Security Properties						**Challenges**										
Schema (Year)	**Chain Growth**	**Chain Quality**	**Common Prefix**	**Validity**	**Block Consistency**	**Transaction Consistency**	**Scalability**	**Additional POW**	**Compatibility**	**Fined Grained**	**Secret Sharing**	**Accountably**	**Anonymity**	**Decentralization**	**Revocation**	**Anti-Collusion**	**Efficiency (High (H)/Medium (M), Poor (P))**
[61], (2017)							√				√					√	H
[62], (2020)					√		√				√					√	H
[63], (2020)					√		√				√					√	H
[64], (2019)	√	√	√		√		√		√	√						√	H
[65], (2020)	√	√	√	√	√	√	√		√	√		√				√	H
[66], (2021)	√	√	√	√	√	√	√		√	√		√				√	M
[67], (2022)	√	√	√	√	√	√	√	√	√	√		√		√		√	H
[68], (2021)	√	√	√	√	√	√	√		√	√		√	√	√	√		M
[69], (2021)	√	√	√	√	√	√			√		√	√	√		√	√	M
[70], (2021)	√	√	√	√	√	√	√	√		√		√			√	√	H
[71], (2021)	√	√	√	√	√	√	√		√	√				√		√	H
[72], (2022)	√	√	√	√	√	√	√		√	√		√		√		√	H
[73], (2021)	√	√	√	√	√	√	√		√	√		√					H
[74], (2021)	√	√	√	√	√	√	√		√	√		√				√	H
[75], (2021)					√		√		√		√						H
[76], (2021)	√	√	√	√	√			√	√		√			√		√	H
[77], (2022)	√	√	√	√	√	√	√	√	√		√	√					M
[78], (2021)	√	√	√	√	√	√	√		√	√		√		√		√	H
[79], (2019)	√	√	√	√	√	√					√	√		√			M
[80], (2019)	√	√	√	√	√	√					√						M
[81], (2020)	√	√	√	√	√						√						M
[82], (2020)	√	√	√	√	√	√					√	√		√			L
[83], (2021)	√	√	√	√	√	√					√	√	√	√			M
[84], (2021)	√	√	√	√	√	√					√	√		√		√	M
[85], (2021)	√	√	√	√	√	√	√	√	√		√						M
[86], (2022)	√	√	√	√	√	√					√	√			√	√	M

Table 5. Non-chameleon redaction approaches challenges comparisons.

Schema (Year)	Security Properties						Challenges									
	Chain Growth	Chain Quality	Common Prefix	Validity	Block Consistency	Transaction Consistency	Scalability	Additional POW	Compatibility	Fined Grained	Secret Sharing	Accountably	Anonymity	Decentralization	Revocation	Anti-Collusion	Efficiency (High (H)/Medium (M), Poor (P))
[95], (2019)	√	√	√	√	√	√		√						√			P
[96], (2020)	√	√	√	√	√	√		√	√					√			P
[97], (2017)	√	√	√	√				√						√			M
[98], (2020)				√		√		√									M
[99], (2019)								√	√								P
[100], (2020)	√	√	√	√	√			√	√								P

9. Open Challenges and Future Research Direction

Recent advancements in designing mutable blockchains are still immature, and consequent modeling redaction techniques require further investigations. This section is dedicated to enriching interested researchers' knowledge and shaping future research direction through open issues listed below:

- **Permissionless settings.** Permissioned and consortium blockchain concepts have adopted most of the proposed solutions where permissionless settings remain a challenge due to their openness and unrestricted characteristics and, hence, it is still unclear how to solve chameleon-based redaction in public blockchain. However, there are several suggestions for a non-chameleon method which proved to be inefficient due to time overhead. Non-chameleon is out of the scope of this research.
- **Redaction exceptional circumstances.** Redaction in total is a delicate case due to violating major blockchain immutability features. The balance requires special supplements, and careful procedures in order to succeed in redaction without any contradiction.
- **Revocation scalability consequences.** Centralization revoking mechanisms lack efficiency and effectiveness. Consequently, it overwhelmingly poses cost, time, communication overhead and even other equipment that might be required.
- **Trapdoor key exposure.** The trapdoor key is the main weakness of the chameleon hashing function; meanwhile, the entire collision being indistinguishable relies on its secrecy.
- **Revocation centralization authorities.** Studies in the current domain have employed central authorities to perform redaction; however, centralization is considered a drawback and decentralization is highly recommended.
- **Punishment methodologies.** Punishment has been replaced by accountability where violators remain safe without any further actions taken against them.
- **IOT-based redaction in blockchain performance.** Scalability is an ineffective factor due to a lack of blockchain redaction in the IoT domain wherein performance criteria are still low due to edge device limitations not being considered. According to authentic requirements, blockchain-based IOT/IIOT designs cannot provide networking resources. Current designs mainly suffer from a dispute among network resources, security, and redaction.

- **The balance between accountability and anonymity.** Privacy is a legal right to preserve identity concealment. Proposals mainly prioritize accountability over personal privacy. However, GDPR legislation strictly states that hidden identities must be maintained which is acutely confronted by the recent redaction mechanisms concepts.
- **Rewriting flexibility limiting.** Absolute power offered to the rewriting modifiers sabotaged data integrity and confined rewriting abilities to themselves.
- **Redactor granting privileges agreement.** The cooperation between owners and modifiers demands prearranged agreements.
- **Collusion resistance:** Colluding must be prevented due to illegal privilege accumulation among different colluded, revoked users who are willing to either access authentic data or try to falsely redact data.
- **Consistency preservation:** Redacted chain stability remains a major obstacle in designing any redaction mechanism. Removal operation performed after storing transaction/block state prior redaction in the current proposal's methods is as yet vulnerable due to verifications and transaction chain failure.

10. Conclusions

The current study aimed to investigate blockchain redaction concepts which emerge as an urgent need to legally violate immutability features due to blockchain technology employment in the wide business sector. Immutability ensures data persistence once approved and published. Current scenarios include adding illicit content that is essentially permitted according to the nature of public blockchain, where immutability becomes an issue that cannot be removed from the blockchain conceptual architecture. Mutability was suggested as an alternative solution to immutability to be applied under special circumstances and highly monitored atmosphere to remove malicious added content; however, it remains difficult to be achieved. The latter has attracted both academia and industry to invest in this research direction to increase blockchain growth effectiveness, wherein several suggested hypotheses were deployed in order to solve previously mentioned drawbacks without effecting either security or scalability. The state of the art is thoroughly analyzed in this paper as it has been classified based on chameleon-based and non-chameleon-based redactable blockchain mechanisms. Non-chameleon suffers mainly from poor efficiency, while chameleon based keeps better efficiency records but at certain domains, such as IoT sectors, its efficiency still needs more enhancements. Future directions must focus on chameleon hash based on its simplicity and it does not add any complication to the blockchain architecture; rather, no certain requirements need to be amended in blockchain infrastructure. However, key exposure, blockchain type and cryptographic methods used must be noted extensively in order to establish balanced solutions among traditionally conflicted security and efficiency.

Author Contributions: Visualization, S.M.A.A.; supervision, S.M.A.A. and H.F.H.; project administration, M.N.Y.; funding acquisition, M.N.Y.; writing—original draft, CS/USM.; writing review and editing, H.F.H.; all authors equally contributed to this work. All authors have read and agreed to the published version of the manuscript.

Funding: Fundamental Research Grant Scheme (FRGS) of grant no. FRGS/1/2019/ICT03/USM/02/1. Postgraduate funding scheme (PFS) of grant no. PFS/9/2019/PhD 19166/01/Art College/ Aliraqia University.

Informed Consent Statement: Not applicable.

Acknowledgments: The authors sincerely acknowledge and dedicate this paper to the support of the Ministry of Higher Education, Malaysia, and Universiti Sains Malaysia (USM). We would also like to express our deepest gratitude to the college of Art, Aliraqia University, Baghdad, Iraq, for their highly appreciated sponsorship.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. List of acronyms.

Abbreviation	Description
CH	Chameleon Hash Function
PKE	Public Key Encryption
ABE	Attribute Base Encryption
TCH	Threshold Chameleon Hash
DS	Digital Signature
ZR	Zero Knowledge Proof
TTCH	Threshold Trapdoor Chameleon Hash
MA_ABE	Multi Authority Attribute Base Encryption
TEE	Trusted Execution Environment
DGS	Digital Group Signature
TUCH	Time Update Chameleon Hash
CHCT	Chameleon Hash Changeable Trapdoor
MTRA	Meta Transaction Base Redactable Approach
CVRA	Consensus Voting Base Redaction Approach
PRA	Pruning Base redaction Approach
P	Permissioned Blockchain (Privat Blockchain)
PL	Permissionless Blockchain (Public Blockchain)
CP	Consortium Blockchain
BL	Block Level
TL	Transaction Level
TFL	Transaction Field Level
DLP	Discrete Logarithm Problem
CDHP	Computational Diffie Hillman Problem
SXDH	Standard Symmetric External Diffie Hillman
SIS	Small Integer Solution
DLIN	Decision Linear Assumption
DPDH	Decisional Bilinear Diffie Hillman

References

1. Ma, Z.; Jiang, M.; Gao, H.; Wang, Z. Blockchain for Digital Rights Management. *Future Gener. Comput. Syst.* **2018**, *89*, 746–764. [CrossRef]
2. Aitzhan, N.Z.; Svetinovic, D. Security and Privacy in Decentralized Energy Trading Through Multi-Signatures, Blockchain and Anonymous Messaging Streams. *IEEE Trans. Dependable Secur. Comput.* **2018**, *15*, 840–852. [CrossRef]
3. Khan, K.M.; Arshad, J.; Khan, M.M. Investigating Performance Constraints for Blockchain Based Secure E-Voting System. *Future Gener. Comput. Syst.* **2020**, *105*, 13–26. [CrossRef]
4. Khan, N.; Aljoaey, H.; Tabassum, M.; Farzamnia, A.; Sharma, T.; Tung, Y.H. Proposed Model for Secured Data Storage in Decentralized Cloud by Blockchain Ethereum. *Electronics* **2022**, *11*, 3686. [CrossRef]
5. Wang, B.; Li, Z. Healthchain: A Privacy Protection System for Medical Data Based on Blockchain. *Future Internet* **2021**, *13*, 247. [CrossRef]
6. Yiu, N.C.K. Toward Blockchain-Enabled Supply Chain Anti-Counterfeiting and Traceability. *Future Internet* **2021**, *13*, 86. [CrossRef]
7. Abidi, M.H.; Alkhalefah, H.; Umer, U.; Mohammed, M.K. Blockchain-Based Secure Information Sharing for Supply Chain Management: Optimization Assisted Data Sanitization Process. *Int. J. Intell. Syst.* **2021**, *36*, 260–290. [CrossRef]
8. Kapassa, E.; Themistocleous, M.; Christodoulou, K.; Iosif, E. Blockchain Application in Internet of Vehicles: Challenges, Contributions and Current Limitations. *Future Internet* **2021**, *13*, 313. [CrossRef]

9. Petroc Taylor. Worldwide Spending on Blockchain Solutions from 2017 to 2024. Available online: https://www.statista.com/statistics/800426/w.orldwide-blockchain-solutions-spending (accessed on 23 May 2022).
10. Hyla, T.; Pejaś, J. EHealth Integrity Model Based on Permissioned Blockchain. *Future Internet* **2019**, *11*, 76. [CrossRef]
11. Sanka, A.I.; Irfan, M.; Huang, I.; Cheung, R.C.C. A Survey of Breakthrough in Blockchain Technology: Adoptions, Applications, Challenges and Future Research. *Comput. Commun.* **2021**, *169*, 179–201. [CrossRef]
12. Narayanan, A.; Bonneau, J.; Felten, E.; Miller, A.; Goldfeder, S.; Clark, J. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*; Princeton University Press: Princeton, NJ, USA, 2016; ISBN 978-0-691-17169-2.
13. Voulgaris, S.; Fotiou, N.; Siris, V.A.; Polyzos, G.C.; Jaatinen, M.; Oikonomidis, Y. Blockchain Technology for Intelligent Environments. *Future Internet* **2019**, *11*, 213. [CrossRef]
14. Przytarski, D.; Stach, C.; Gritti, C.; Mitschang, B. Query Processing in Blockchain Systems: Current State and Future Challenges. *Future Internet* **2022**, *14*, 1. [CrossRef]
15. Al-Abdullah, M.; Alsmadi, I.; AlAbdullah, R.; Farkas, B. Designing Privacy-Friendly Data Repositories: A Framework for a Blockchain That Follows the GDPR. *Digit. Policy Regul. Gov.* **2020**, *22*, 389–411. [CrossRef]
16. Zhang, D.; Le, J.; Mu, N.; Liao, X. An Anonymous Off-Blockchain Micropayments Scheme for Cryptocurrencies in the Real World. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 32–42. [CrossRef]
17. Zheng, X.; Zhu, Y.; Si, X. A Survey on Challenges and Progresses in Blockchain Technologies: A Performance and Security Perspective. *Appl. Sci.* **2019**, *9*, 4731. [CrossRef]
18. El Ioini, N.; Pahl, C. A Review of Distributed Ledger Technologies. In Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Valletta, Malta, 22–26 October 2018; Volume 11230.
19. Casino, F.; Politou, E.; Alepis, E.; Patsakis, C. Immutability and Decentralized Storage: An Analysis of Emerging Threats. *IEEE Access* **2020**, *8*, 4737–4744. [CrossRef]
20. Matzutt, R.; Hiller, J.; Henze, M.; Ziegeldorf, J.H.; Müllmann, D.; Hohlfeld, O.; Wehrle, K. A Quantitative Analysis of the Impact of Arbitrary Blockchain Content on Bitcoin. In Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Nieuwpoort, The Netherlands, 26 February–2 March 2018; Volume 10957.
21. Jordan Pearson The Bitcoin Blockchain Could Be Used to Spread Malware, INTERPOL Says: (27 March 2015). Available online: https://www.vice.com/en/article/ezv8jn/the-bitcoin-blockchain-could-be-used-to-spread-malware-interpol-says (accessed on 23 May 2022).
22. Tziakouris, G. Cryptocurrencies—A Forensic Challenge or Opportunity for Law Enforcement? An INTERPOL Perspective. *IEEE Secur. Priv.* **2018**, *16*, 92–94. [CrossRef]
23. Schellinger, B.; Völter, F.; Urbach, N.; Sedlmeir, J. Yes, I Do: Marrying Blockchain Applications with GDPR. In Proceedings of the 55th Hawaii International Conference on System Sciences, Maui, HI, USA, 4–7 January 2022.
24. Bai, P.; Kumar, S.; Kumar, K.; Kaiwartya, O.; Mahmud, M.; Lloret, J. GDPR Compliant Data Storage and Sharing in Smart Healthcare System: A Blockchain-Based Solution. *Electronics* **2022**, *11*, 3311. [CrossRef]
25. Campanile, L.; Iacono, M.; Marulli, F.; Mastroianni, M. Designing a GDPR Compliant Blockchain-Based IoV Distributed Information Tracking System. *Inf. Process. Manag.* **2021**, *58*, 102511. [CrossRef]
26. Bigini, G.; Freschi, V.; Lattanzi, E. A Review on Blockchain for the Internet of Medical Things: Definitions, Challenges, Applications, and Vision. *Future Internet* **2020**, *12*, 208. [CrossRef]
27. Gatteschi, V.; Lamberti, F.; Demartini, C.; Pranteda, C.; Santamaría, V. Blockchain and Smart Contracts for Insurance: Is the Technology Mature Enough? *Future Internet* **2018**, *10*, 20. [CrossRef]
28. Zheng, J.; Dike, C.; Pancari, S.; Wang, Y.; Giakos, G.C.; Elmannai, W.; Wei, B. An In-Depth Review on Blockchain Simulators for IoT Environments. *Future Internet* **2022**, *14*, 182. [CrossRef]
29. Politou, E.; Alepis, E.; Patsakis, C. Forgetting Personal Data and Revoking Consent under the GDPR: Challenges and Proposed Solutions. *J. Cybersecur.* **2018**, *4*, tyy001. [CrossRef]
30. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System, October 2008. *Cited* 2008. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 23 May 2022).
31. Swan, M. *Blockchain: Blueprint for a New Economy*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2015.
32. Koblitz, N.; Menezes, A.J. Cryptocash, Cryptocurrencies, and Cryptocontracts. *Des. Codes Cryptogr.* **2016**, *78*, 87–102. [CrossRef]
33. Workie, H. Distributed Ledger Technology: Implications of Blockchain for the Securities Industry. *J. Secur. Oper. Custody* **2017**, *9*, 347–355.
34. Chaudhary, K.; Fehnker, A.; Van De Pol, J.; Stoelinga, M. Modeling and Verification of the Bitcoin Protocol. In Proceedings of the Electronic Proceedings in Theoretical Computer Science, EPTCS, Suva, Fiji, 23 November 2015; Volume 196.
35. Tschorsch, F.; Scheuermann, B. Bitcoin and beyond: A Technical Survey on Decentralized Digital Currencies. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2084–2123. [CrossRef]
36. Christodoulou, K.; Iosif, E.; Inglezakis, A.; Themistocleous, M. Consensus Crash Testing: Exploring Ripple's Decentralization Degree in Adversarial Environments. *Future Internet* **2020**, *12*, 53. [CrossRef]
37. Chen, Y.; Guo, J.; Li, C.; Ren, W. FaDe: A Blockchain-Based Fair Data Exchange Scheme for Big Data Sharing. *Future Internet* **2019**, *11*, 225. [CrossRef]

38. Conte de Leon, D.; Stalick, A.Q.; Jillepalli, A.A.; Haney, M.A.; Sheldon, F.T. Blockchain: Properties and Misconceptions. *Asia Pac. J. Innov. Entrep.* **2017**, *11*, 286–300. [CrossRef]
39. Khanal, Y.P.; Alsadoon, A.; Shahzad, K.; Al-Khalil, A.B.; Prasad, P.W.C.; Rehman, S.U.; Islam, R. Utilizing Blockchain for IoT Privacy through Enhanced ECIES with Secure Hash Function. *Future Internet* **2022**, *14*, 77. [CrossRef]
40. Buccafurri, F.; De Angelis, V.; Lazzaro, S. A Blockchain-Based Framework to Enhance Anonymous Services with Accountability Guarantees. *Future Internet* **2022**, *14*, 243. [CrossRef]
41. Chen, Y.-C.; Chou, Y.-P.; Chou, Y.-C. An Image Authentication Scheme Using Merkle Tree Mechanisms. *Future Internet* **2019**, *11*, 149. [CrossRef]
42. Feng, H.; Wang, J.; Li, Y. An Efficient Blockchain Transaction Retrieval System. *Future Internet* **2022**, *14*, 267. [CrossRef]
43. Eyal, I.; Gencer, A.E.; Sirer, E.G.; Van Renesse, R. Bitcoin-NG: A Scalable Blockchain Protocol. In Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2016, Santa Clara, CA, USA, 16–18 March 2016.
44. Qu, Q.; Xu, R.; Chen, Y.; Blasch, E.; Aved, A. Enable Fair Proof-of-Work (PoW) Consensus for Blockchains in IoT by Miner Twins (MinT). *Future Internet* **2021**, *13*, 291. [CrossRef]
45. Bentov, I.; Gabizon, A.; Mizrahi, A. Cryptocurrencies without Proof of Work. In Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Christ Church, Barbados, 26 February 2016; Volume 9604.
46. Gazi, P.; Kiayias, A.; Zindros, D. Proof-of-Stake Sidechains. In Proceedings of the IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 19–23 May 2019; Volume 2019.
47. Bentov, I.; Lee, C.; Mizrahi, A.; Rosenfeld, M. Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake. *Cryptol. Eprint Arch.* **2014**, *452*, 34–37.
48. Duan, S.; Reiter, M.K.; Zhang, H. BEAT: Asynchronous BFT Made Practical. In Proceedings of the ACM Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018.
49. Xiao, Y.; Zhang, N.; Lou, W.; Hou, Y.T. A Survey of Distributed Consensus Protocols for Blockchain Networks. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1432–1465. [CrossRef]
50. Antonopoulo, A.M. *Mastering Bitcoin Unlocking Digital Cryptocurrencies*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2014; Volume 9.
51. Habib, G.; Sharma, S.; Ibrahim, S.; Ahmad, I.; Qureshi, S.; Ishfaq, M. Blockchain Technology: Benefits, Challenges, Applications, and Integration of Blockchain Technology with Cloud Computing. *Future Internet* **2022**, *14*, 341. [CrossRef]
52. Makridakis, S.; Christodoulou, K. Blockchain: Current Challenges and Future Prospects/Applications. *Future Internet* **2019**, *11*, 258. [CrossRef]
53. Gong, Y.; van Engelenburg, S.; Janssen, M. A Reference Architecture for Blockchain-Based Crowdsourcing Platforms. *J. Theor. Appl. Electron. Commer. Res.* **2021**, *16*, 937–958. [CrossRef]
54. Yu, H.; Yang, Z.; Sinnott, R.O. Decentralized Big Data Auditing for Smart City Environments Leveraging Blockchain Technology. *IEEE Access* **2019**, *7*, 6288–6296. [CrossRef]
55. Cai, W.; Wang, Z.; Ernst, J.B.; Hong, Z.; Feng, C.; Leung, V.C.M. Decentralized Applications: The Blockchain-Empowered Software System. *IEEE Access* **2018**, *6*, 53019–53033. [CrossRef]
56. Xie, J.; Tang, H.; Huang, T.; Yu, F.R.; Xie, R.; Liu, J.; Liu, Y. A Survey of Blockchain Technology Applied to Smart Cities: Research Issues and Challenges. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2794–2830. [CrossRef]
57. Bodziony, N.; Jemioło, P.; Kluza, K.; Ogiela, M.R. Blockchain-Based Address Alias System. *J. Theor. Appl. Electron. Commer. Res.* **2021**, *16*, 1280–1296. [CrossRef]
58. Garay, J.; Kiayias, A.; Leonardos, N. The Bitcoin Backbone Protocol: Analysis and Applications. In Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Sofia, Bulgaria, 26–30 April 2015; Volume 9057.
59. Kiayias, A.; Russell, A.; David, B.; Oliynykov, R. Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol. In Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Santa Barbara, CA, USA, 20–24 August 2017; Volume 10401.
60. Krawczyk, H.; Rabin, T. Chameleon Hashing and Signatures. *IACR Cryptol. Eprint Arch.* **1998**, *1998*. Available online: https://eprint.iacr.org/1998/010 (accessed on 23 May 2022).
61. Ateniese, G.; Magri, B.; Venturi, D.; Andrade, E.R. Redactable Blockchain—Or—Rewriting History in Bitcoin and Friends. In Proceedings of the 2nd IEEE European Symposium on Security and Privacy, EuroS and P 2017, Paris, France, 26–28 April 2017.
62. Khalili, M.; Dakhilalian, M.; Susilo, W. Efficient Chameleon Hash Functions in the Enhanced Collision Resistant Model. *Inf. Sci.* **2020**, *510*, 155–164. [CrossRef]
63. Derler, D.; Samelin, K.; Slamanig, D. Bringing Order to Chaos: The Case of Collision-Resistant Chameleon-Hashes. In Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Edinburgh, UK, 4–7 May 2020; Volume 12110.
64. Derler, D.; Samelin, K.; Slamanig, D.; Striecks, C. Fine-Grained and Controlled Rewriting in Blockchains: Chameleon-Hashing Gone Attribute-Based. IACR Crypto ePrint Archive. 2019. 406p. Available online: https://eprint.iacr.org/2019/406 (accessed on 23 May 2022).
65. Tian, Y.; Li, N.; Li, Y.; Szalachowski, P.; Zhou, J. Policy-Based Chameleon Hash for Blockchain Rewriting with Black-Box Accountability. In Proceedings of the ACM International Conference Proceeding Series, Austin, TX, USA, 7–11 December 2020.

66. Xu, S.; Ning, J.; Ma, J.; Huang, X.; Deng, R.H. K-Time Modifiable and Epoch-Based Redactable Blockchain. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 4507–4520. [CrossRef]
67. Chen, X.; Gao, Y. CDEdit: A Highly Applicable Redactable Blockchain with Controllable Editing Privilege and Diversified Editing Types. *arXiv* **2022**, arXiv:2205.07054.
68. Panwar, G.; Vishwanathan, R.; Misra, S. ReTRACe: Revocable and Traceable Blockchain Rewrites Using Attribute-Based Cryptosystems. In Proceedings of the ACM Symposium on Access Control Models and Technologies, SACMAT, Virtual, 16–18 June 2021.
69. Jia, Y.; Sun, S.F.; Zhang, Y.; Liu, Z.; Gu, D. Redactable Blockchain Supporting Supervision and Self-Management. In Proceedings of the ASIA CCS 2021—Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security, Hong Kong, China, 7–11 June 2021.
70. Xu, S.; Ning, J.; Ma, J.; Xu, G.; Yuan, J.; Deng, R.H. Revocable Policy-Based Chameleon Hash. In Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Darmstadt, Germany, 4–8 October 2021; Volume 12972.
71. Zhang, Z.; Li, T.; Wang, Z.; Liu, J. Redactable Transactions in Consortium Blockchain: Controlled by Multi-Authority CP-ABE. In Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Virtual Event, 1–3 December 2021; Volume 13083.
72. Ma, J.; Xu, S.; Ning, J.; Huang, X.; Deng, R.H. Redactable Blockchain in Decentralized Setting. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 1227–1242. [CrossRef]
73. Guo, L.; Wang, Q.; Yau, W.C. Online/Offline Rewritable Blockchain with Auditable Outsourced Computation. *IEEE Trans. Cloud Comput.* **2021**, *14*, 1–16. [CrossRef]
74. Hou, H.; Hao, S.; Yuan, J.; Xu, S.; Zhao, Y. Fine-Grained and Controllably Redactable Blockchain with Harmful Data Forced Removal. *Secur. Commun. Netw.* **2021**, *2021*, 3680359. [CrossRef]
75. Liu, L.; Tan, L.; Liu, J.; Xiao, J.; Yin, H.; Tan, S. Redactable Blockchain Technology Based on Distributed Key Management and Trusted Execution Environment. In Proceedings of the Communications in Computer and Information Science, Guangzhou, China, 5–6 August 2021; Volume 1490.
76. Wu, C.; Ke, L.; Du, Y. Quantum Resistant Key-Exposure Free Chameleon Hash and Applications in Redactable Blockchain. *Inf. Sci.* **2021**, *548*, 438–449. [CrossRef]
77. Matzutt, R.; Ahlrichs, V.; Pennekamp, J.; Karwacik, R.; Wehrle, K. A Moderation Framework for the Swift and Transparent Removal of Illicit Blockchain Content. In Proceedings of the IEEE International Conference on Blockchain and Cryptocurrency, ICBC 2022, Shanghai, China, 2–5 May 2022; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2022.
78. Tian, Y.; Liu, B.; Li, Y.; Szalachowski, P.; Zhou, J. Accountable Fine-Grained Blockchain Rewriting in the Permissionless Setting. *arXiv* **2021**, arXiv:2104.13543.
79. Huang, K.; Zhang, X.; Mu, Y.; Wang, X.; Yang, G.; Du, X.; Rezaeibagha, F.; Xia, Q.; Guizani, M. Building Redactable Consortium Blockchain for Industrial Internet-of-Things. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3670–3679. [CrossRef]
80. Huang, K.; Zhang, X.; Mu, Y.; Rezaeibagha, F.; Du, X.; Guizani, N. Achieving Intelligent Trust-Layer for Internet-of-Things via Self-Redactable Blockchain. *IEEE Trans. Ind. Inform.* **2020**, *16*, 2677–2686. [CrossRef]
81. Zhang, J.; Lu, Y.; Liu, Y.; Yang, X.; Qi, Y.; Dong, X.; Wang, H. Serving at the Edge: A Redactable Blockchain with Fixed Storage. In Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Guangzhou, China, 23–25 September 2020; Volume 12432.
82. Lv, W.; Wei, S.; Li, S.; Yu, M. Verifiable Blockchain Redacting Method for a Trusted Consortium with Distributed Chameleon Hash Authority. In Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Dallas, TX, USA, 11–13 December 2020; Volume 12575.
83. Huang, K.; Zhang, X.; Mu, Y.; Rezaeibagha, F.; Du, X. Scalable and Redactable Blockchain with Update and Anonymity. *Inf. Sci.* **2021**, *546*, 25–41. [CrossRef]
84. Gao, W.; Chen, L.; Rong, C.; Liang, K.; Zheng, X.; Yu, J. Security Analysis and Improvement of a Redactable Consortium Blockchain for Industrial Internet-of-Things. *Comput. J.* **2022**, *65*, 2430–2438. [CrossRef]
85. Zhang, C.; Ni, Z.; Xu, Y.; Luo, E.; Chen, L.; Zhang, Y. A Trustworthy Industrial Data Management Scheme Based on Redactable Blockchain. *J. Parallel Distrib. Comput.* **2021**, *152*, 167–176. [CrossRef]
86. Wei, J.; Zhu, Q.; Li, Q.; Nie, L.; Shen, Z.; Choo, K.K.R.; Yu, K. A Redactable Blockchain Framework for Secure Federated Learning in Industrial Internet of Things. *IEEE Internet Things J.* **2022**, *9*, 17901–17911. [CrossRef]
87. Camenisch, J.; Derler, D.; Krenn, S.; Pöhls, H.C.; Samelin, K.; Slamanig, D. Chameleon-Hashes with Ephemeral Trapdoors and Applications to Invisible Sanitizable Signatures. In Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Amsterdam, The Netherlands, 28–31 March 2017; Volume 10175.
88. Goyal, V.; Pandey, O.; Sahai, A.; Waters, B. Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data. In Proceedings of the ACM Conference on Computer and Communications Security, Taipei, Taiwan, 21–24 March 2006.
89. Rivest, R.L.; Shamir, A.; Adleman, L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]

90. Boldyreva, A.; Goyal, V.; Kumart, V. Identity-Based Encryption with Efficient Revocation. In Proceedings of the ACM Conference on Computer and Communications Security, Alexandria, VA, USA, 27–31 October 2008.
91. Agrawal, S.; Chase, M. FAME: Fast Attribute-Based Message Encryption. In Proceedings of the ACM Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017.
92. Liu, J.K.; Wei, V.K.; Wong, D.S. Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups (Extended Abstract). *Lect. Notes Comput. Sci. (Incl. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinform.)* **2004**, *3108*, 325–335. [CrossRef]
93. Sabt, M.; Achemlal, M.; Bouabdallah, A. Trusted Execution Environment: What It Is, and What It Is Not. In Proceedings of the 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2015, Helsinki, Finland, 20–22 August 2015; Volume 1.
94. Ateniese, G.; Chou, D.H.; Medeiros, B.D.; Tsudik, G. Sanitizable Signatures. In Proceedings of the n European Symposium on Research in Computer Security, Milan, Italy, 12–14 September 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 159–177.
95. Deuber, D.; Magri, B.; Thyagarajan, S.A.K. Redactable Blockchain in the Permissionless Setting. In Proceedings of the Proceedings—IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 19–23 May 2019.
96. Thyagarajan, S.A.K.; Bhat, A.; Magri, B.; Tschudi, D.; Kate, A. Reparo: Publicly Verifiable Layer to Repair Blockchains. In Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Virtual Event, 1–5 March 2021; Volume 12675.
97. Puddu, I.; Zurich, E.; Dmitrienko, A.; Capkun, S. Mchain: How to Forget without Hard Forks. *arXiv* **2017**. Available online: https://eprint.iacr.org/2017/106.pdf (accessed on 23 May 2022).
98. Dorri, A.; Kanhere, S.S.; Jurdak, R. MOF-BC: A Memory Optimized and Flexible Blockchain for Large Scale Networks. *Future Gener. Comput. Syst.* **2019**, *92*, 357–373. [CrossRef]
99. Florian, M.; Henningsen, S.; Beaucamp, S.; Scheuermann, B. Erasing Data from Blockchain Nodes. In Proceedings of the 4th IEEE European Symposium on Security and Privacy Workshops, EUROS and PW 2019, Stockholm, Sweden, 17–19 June 2019.
100. Matzutt, R.; Kalde, B.; Pennekamp, J.; Drichel, A.; Henze, M.; Wehrle, K. How to Securely Prune Bitcoin's Blockchain. In Proceedings of the IFIP Networking 2020 Conference and Workshops, Networking 2020, Paris, France, 22–26 June 2020.
101. Yiu, N.C.K. An Overview of Forks and Coordination in Blockchain Development. *arXiv* **2021**, arXiv:2102.10006.
102. Webb, N. A Fork in the Blockchain: Income Tax and the Bitcoin/Bitcoin Cash Hard Fork. *North Carol. J. Law Technol.* **2018**, *19*, 283.
103. Marsalek, A.; Zefferer, T. A Correctable Public Blockchain. In Proceedings of the 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering, TrustCom/BigDataSE 2019, Rotorua, New Zealand, 5–8 August 2019.
104. Ostrovsky, R.; Yung, M. How to Withstand Mobile Virus Attacks (Extended Abstract). In Proceedings of the Tenth Annual ACM Symposium on Principles of Distributed Computing—PODC '91, Montreal, QC, Canada, 19–21 August 1991.
105. Pyoung, C.K.; Baek, S.J. Blockchain of Finite-Lifetime Blocks with Applications to Edge-Based IoT. *IEEE Internet Things J.* **2020**, *7*, 2102–2116. [CrossRef]
106. Xu, X.; Zhu, D.; Yang, X.; Wang, S.; Qi, L.; Dou, W. Concurrent Practical Byzantine Fault Tolerance for Integration of Blockchain and Supply Chain. *ACM Trans. Internet Technol.* **2021**, *21*, 1–17. [CrossRef]