Journal of
Cybersecurity and Privacy

MDPI

Review

Secure and Privacy-Aware Blockchain Design: Requirements, Challenges and Solutions

Sidra Aslam [1,2], Aleksandar Tošić [1,2] and Michael Mrissa [1,2,*]

[1] InnoRenew CoE, Izola 6310, Slovenia; sidra.aslam@innorenew.eu (S.A.); aleksandar.tosic@innorenew.eu (A.T.)

[2] Faculty of Mathematics, Natural Sciences and Information Technologies, University of Primorska, Koper 6000, Slovenia

* Correspondence: michael.mrissa@innorenew.eu

Received: 28 December 2020; Accepted: 9 March 2021; Published: 14 March 2021

check for updates

Abstract: During the last decade, distributed ledger solutions such as blockchain have gained significant attention due to their decentralized, immutable, and verifiable features. However, the public availability of data stored on the blockchain and its link to users may raise privacy and security issues. In some cases, addressing these issues requires blockchain data to be secured with mechanisms that allow on-demand (as opposed to full) disclosure. In this paper, we give a comprehensive overview of blockchain privacy and security requirements, and detail how existing mechanisms answer them. We provide a taxonomy of current attacks together with related countermeasures. We present a thorough comparative analysis based on various parameters of state-of-the-art privacy and security mechanisms, we provide recommendations to design secure and privacy-aware blockchain, and we suggest guidelines for future research.

Keywords: blockchain; privacy; security

1. Introduction

Usually, people rely on a Trusted Third Party (TTP), such as an online bank operator, to transfer assets over the internet. A TTP is responsible for guaranteeing a secure exchange and intervenes in case of failure or security breach. However, the centralized design of TTPs makes them vulnerable to the single point of failure problem. Nowadays, blockchain [1] has been widely adopted as an immutable (if some constraints are respected, such as, the majority of the nodes are honest) Distributed Ledger (a distributed database that shares transaction records between all the participants) technology (DLT) allowing transaction information to be securely replicated over a set of nodes, thus eliminating the need for TTP and avoiding the single point of failure problem. However, transaction data on the blockchain is publicly accessible, and at the same time, depending on the requirements of their application, users may need, in some cases, on-demand (as opposed to full) data disclosure, leading to privacy concerns that should be handled in the design of the blockchain for this particular solution. On-demand data disclosure requires fine-grained access control to check who is the recipient of the data before applying required security (encryption strategies) and privacy (data protection) measures.

We define privacy-sensitive data as any data that is subject to disclosure constraints. Such data is highly dependent on the application domain (medical care, production chain, etc.). Diverse solutions are available to overcome the privacy challenge (such as k-anonymity, l-diversity, t-closeness, etc.). In this context, privacy solutions are combined with security measures (such as mix network [2], Zero-Knowledge Proofs (ZKPs) [3], ring signature [4], etc.) that provide required services to protect data from unwanted disclosure, control data access, and prevent identity theft. Although several studies [5–10] focus on security or privacy, a comprehensive and structured analysis of blockchain requirements, the latest advances and their relevance in this context is still missing as discussed below.

1.1. Paper Position

In the following, we highlight the relationship between the current work presented in this paper and existing work on blockchain attacks, countermeasures, privacy and security threats, and defense mechanisms. Authors in [11], review Ethereum security, attacks, and defense mechanisms without considering privacy attacks and their defense mechanisms. They identified layers of attacks for Ethereum, but do not address other blockchain implementations. More specifically, their Ethereum application layer, data layer, consensus layer, environment layer correspond to our blockchain attack categories and their network layer correspond to our network attack category, as shown in Section 4. We took into consideration this classification of Ethereum attacks to build our own attack classification that approaches blockchain from an inclusive high-level perspective, without binding to any specific implementation.

Therefore, our attack classification presents the following advantages. First, it applies to all types of DLT implementations, including Ethereum but not limited to it. Second, we also include attacks related to user and transaction data privacy and security. Third, for each attack, we provide a description of the malicious user's goal, a detailed discussion on privacy and security countermeasures, and we discuss countermeasure advantages and limitations.

Other survey papers [5,8] focus only on blockchain privacy threats and mechanisms, without addressing security aspects. In our work, we provide a comprehensive overview of privacy and security challenges, attacks, requirements, and solutions for blockchain. We analyze the limitations of existing solutions with respect to complexity, trust, computation time, transaction time, cost, latency, and involvement of a trusted third party.

In [9], privacy and security mechanisms for blockchain transactions are studied, however, only a few blockchain attacks are addressed (double-spending attack, majority attack, and DDoS attack) which is only 3 of the 21 network and blockchain attacks discussed in this paper. We also provide a comparative analysis of existing mechanisms based on different parameters (e.g., scalability, complexity, size, computation, and communication overhead, etc.).

In [6,7], authors mainly focused on privacy and security for bitcoin only without discussing users' identity management mechanisms and blockchain challenges as we do in this paper. Another survey [12], focuses on blockchain privacy and security issues without discussing countermeasures to overcome them.

1.2. Contribution

In this paper, we structure our contribution as follows:

- We provide detailed background knowledge of blockchain design, and discuss the development of blockchain implementations.
- We identify blockchain requirements in terms of privacy and security, and we present several mechanisms to meet them.
- We propose a classification of current attacks and matching countermeasures.
- We provide a critical comparative analysis of the state of the art privacy and security defense mechanisms to highlight their corresponding advantages and drawbacks.
- We present step-by-step recommendations to provide understanding of the mechanisms and we then discuss current trends based on our study and suggest future work to improve privacy and security management for blockchain.

The rest of this paper is organized as follows. Section 2 overviews background knowledge about blockchain technology and blockchain generations to facilitate understanding of this paper. It then describes blockchain requirements for privacy and security and summarizes existing mechanisms to enforce them. Section 4 proposes a classification of existing attacks and describes related countermeasures.

Section 5 provides a comparative analysis, detailed in multiple tables, of existing technological solutions. Based on this analysis, it covers directions for future research towards improved privacy and security management of blockchain. Section 6 summarizes the main outcome of our study and our guidelines for future work.

2. Blockchain: Background Knowledge

In this section, we provide the necessary background knowledge about blockchain design and operation for a good understanding of this paper. After that, we discuss blockchain structure and its main components. Then, we present the global view of blockchain generations and classification of blockchain types.

2.1. Overview of Blockchain Technology

A blockchain is a transparent and decentralized database, secured with cryptography techniques to enforce the integrity of its contents, stored under the form of transactions grouped in blocks linked to each other [13]. A block is based on two main parts which are the block header and the transactions. The block header comprises of different fields such as block version, the Merkle tree Root Hash, timestamp, nonce, and parent block hash. Transactions include information such as transaction time, transaction amount, recipient address, etc. and their contents can be verified by any user on the network. Transactions are broadcast through the network of nodes and are eventually added to a block after a process called mining.

Concretely, mining is the process of adding a block to the blockchain using consensus mechanisms (e.g., proof of work, proof of stake, etc.) to validate the transactions in a block [14]. It helps to verify transactions from this block and secure the blockchain against a double-spending attack. Miners receive rewards to validate the block (e.g., new coin and transaction fees).

In order to add each block in the chain, the proof of work mechanism requires miners to solve complex mathematical calculations that must be accepted by all the miners. Once the miners validate the transactions, a block is added to the blockchain network. For this solution to work, it must be difficult for an attacker to compromise more than half of the hashing power on the network. It is easy and fast to verify the proof and its correctness. However, proof of work is inefficient due to high computational power to solve complex mathematical calculations.

To cope with this issue, other consensus mechanisms such as proof of stake [15] allow miners to validate block transactions according to the amount of digital currency (stake) a miner owns [1]. As compared to proof of work it saves more energy because it eliminates the high amount of computing power required from the consensus mechanism. Unfortunately, it may lead to unwanted centralization since rewards are distributed based on the amount staked. Wealthy node operators will earn the most rewards, making them even more wealthy and in time, enable them to operate the majority of nodes. Different methods exist, such as the coin age selection approach that is used to avoid the wealthiest node in the network [16]. The coin age selection method selects nodes based on how long their tokens have been staked for. In this approach, older and larger sets of coins have a higher chance of mining the next block. Once a user has mined a block, their coin age is reset to zero and then they must wait a certain period of time to mine another block.

2.2. Structure of Blockchain

- **Block:** In blockchain, verified transactions are stored in a block. Any node in a blockchain system can initiate a transaction and send a copy to other available nodes on the network. When network nodes verify that all stored transactions in the block are valid, the next step is to add a block to the blockchain. Each block contains certain information such as transaction time, number of occurred transactions, etc. Each block is linked to the hash of the previous block via a hash function (Hash functions such as MD5 and SHA-256 transform an input message into a fixed size value called a hash, making it impossible to go back to the original text, but possible to check that a copy is exact by comparing generated hashes [17]) to ensure immutability. A unique hash code is then assigned to each block which is generated by the hash function and differentiates a current block from other blocks.

- **Merkle tree:** It is defined as a binary search tree in which tree nodes are linked to each other using hash pointers. The blockchain transactions are arranged in a Merkle tree structure. The hashes of all nodes are combined to create the Merkle tree [18], in which each two child nodes are combined together into one node called parent node as shown in Figure 1. This process is repeated from down to top to reach the root of the tree. The root of the Merkle tree verifies all the transactions in the block. An advantage of Merkle tree is that it allows us to prove both the integrity and validity of data. If an adversary attempts to change the transaction then it is needed to change all the subsequent block hashes.
- **Digital signature:** It ensures the data validity by using a cryptographic algorithm. The digital signature scheme is comprised of three components. The first component is a key generation algorithm, which generates a key pair known as a private key and public key. The private key is kept secret and is used to sign a message, whereas the public key is available to the public and is used to verify the message whether the message has been signed with the corresponding private key. Second component is a signing algorithm, which uses the given private key to creates a signature on the input message. The third component is a verification algorithm, that takes three parameters as input, such as signature, a message, public key and validates the message signature by using the public key. The advantage of using a digital signature is to prevent non-repudiation so participants on the blockchain network cannot deny their own activities.

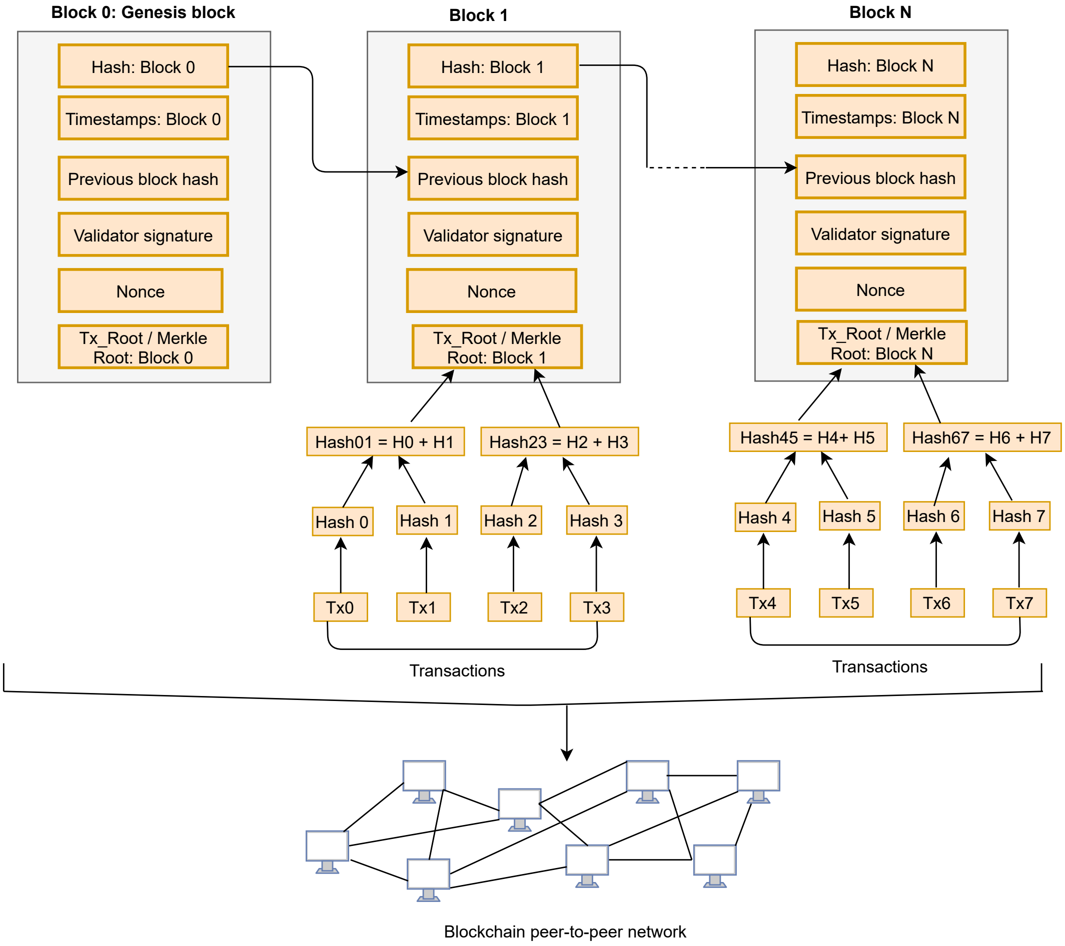

Figure 1. Merkle tree structure on blockchain.

2.3. Development of Blockchain Implementations

Over time, it has been observed that the development of blockchain can be organized into 3 generations [19–21], as follows.

- **First generation: digital currency.** The first generation of blockchain is a decentralized digital currency (e.g., bitcoin or digital coin) that allows participants to make transactions directly without the involvement of a centralized party [19]. The first blockchain generation solved major issues to create decentralized currency, this generation mainly relied on proof of work consensus algorithm. The main advantages of this generation are to provide decentralized storage, to enable nodes to share data directly, and to ensure transparency during transaction processing [20]. The main drawbacks are related to the energy consumption of consensus algorithms and the fact that proof-of-work gives rewards to the participants who already have the most computation power, which can be a security issue.
- **Second generation: smart contracts.** The concept of smart contract was proposed to support the self-execution of programs that execute when specific terms are met [22]. The smart contract code is stored on a public ledger and cannot be changed. The transactions that occur in a smart contract are run by the blockchain without a third party. A contract is comprised of three main parts: a unique contract address to identify them on the blockchain, private storage and amount of balance (for example, Ethereum the first implementation to provide smart contracts uses the Ether cryptocurrency). A smart contract can be written in high-level languages, for example, Ethereum uses the Solidity programming language, which can compile into low-level bytecode for the Ethereum virtual machine (EVM) code [23]. Smart contracts guarantee tamper-proof fraud prevention and reduce verification costs. It is important to highlight that some blockchain implementations do not fully, or at all, implement a smart contract. However, this second-generation still remains limited in terms of performance and scalability, as witnesses, for example, the low number to process transactions per second [21].
- **Third generation: scalability.** The third generation of blockchain focuses on improving the scalability identified in previous generations. Most limitations relate to mining delay, energy consumption, low number of transactions, mostly related to the use of proof of work type of consensus and application of smart contracts. We have also observed a growth in the number and variety of applications, such as e-Health [24] and supply chain management systems [25]. For example, the third generation of blockchain platforms includes Dfinity [26] and NEO [27], that aim at supporting different programming languages and the development of mobile-based applications [20].

Generally, blockchain has different types depending on data availability and on what actions are allowed to perform on data by the user [28]. Thus, the following types of blockchain are available nowadays: public, private and consortium.

- **Public blockchain:** is a permissionless ledger that is available to everyone on the network and anyone can view, read and write data on the blockchain [29]. Examples of public blockchain include Ethereum and bitcoin [30].
- **Private blockchain:** is a permissioned blockchain, which allows only specific people to verify and add transaction blocks to the blockchain [31]. Monax and multichain are private blockchain [30].
- **Consortium blockchain:** is known as federated or public permission blockchain, which allows only a group of organizations to verify and add data to the blockchain. It can be an open ledger or restricted to a particular group. R3 and Corda are consortium blockchain [32].

Although blockchain brings several advantages such as trust, traceability, transparency, and secure distributed storage, it also has specific privacy and security requirements, as described below.

3. Privacy and Security Requirements for Blockchain

In this section, we provide a detailed analysis of privacy and security requirements for blockchain, before showing how current technological advances answer them and highlighting their shortcomings, as summarized in Table 1.

Blockchain requirements with respect to privacy and security differ from usual ones due to its decentralized operation. Indeed, blockchain provides privacy and security to sensitive data as a user can make transactions with public and private keys without revealing their real identity. However, public blockchain does not guarantee the privacy of transactions since the content of transactions (e.g., amount) is publicly visible [33,34]. This issue causes leakage of privacy-sensitive information. Furthermore, users' bitcoin addresses can be linked to identify users' real identity [35]. Therefore, we suggest that to provide privacy and security on blockchain, mechanisms should be selected based on well-defined requirements. We identify and detail the most significant requirements below.

- **Transaction data protection:** The transaction contents (payload) recorded on the distributed ledger (such as healthcare or business data) and the transaction itself (e.g., transaction time and amount, etc.) may need to be protected from unauthorized disclosure. Indeed, such information may be sensitive, especially for companies with public records that can be matched and that may not want to disclose their transactions on the blockchain.
- **User identity protection:** As mentioned above, the structure of transactions establishes a direct link to the involved users through asymmetric cryptography (if one finds out who a public key belongs to). Typically, blockchain users may want to protect the link between their identity and the transactions they perform, thus making identity protection a major issue for blockchain.
- **Security properties:** Typical data security properties are required: confidentiality to protect from unauthorized data access, integrity to make sure the contents are the original ones, availability as data should be accessible upon request and non-repudiation as recorded data cannot be denied. By definition, blockchain implementations must provide such properties, therefore, the challenge is to make sure that additional measures dealing with other requirements do not break any of those properties.
- **Avoiding TTP:** The centralized ledgers of current transaction systems (e.g., banks) are subject to the single-point of failure vulnerability, so they can be more easily compromised than decentralized solutions. As Blockchain helps to eliminate TTPs, additional solutions that answer privacy and security problems should avoid relying on TTPs as well. Unfortunately, it is still common practice to rely on TTPs [36].
- **Blockchain support:** While our research approaches blockchain from a theoretical perspective, it is important to look at existing blockchain implementations that show the feasibility of those theoretical advances. Therefore, we show in Table 1 how the studied privacy and security mechanisms are supported with blockchain implementations.

Table 1. Overview of blockchain requirements and existing mechanisms.

Mechanism	Requirements				
	Identity Protection	**Transaction Data Protection**	**Security Properties**	**Avoiding TTP**	**Blockchain Support**
Ring Signatures	Yes [4,37,38]	Yes [39]	Confidentiality [4,37,38]	No [37] Yes [4,38]	Monero BC [39]
Zero-Knowledge proof	Yes [3,5]	Yes [5]	Integrity [40]; Authentication [3]	No [3,40]	Bitcoin and Ethereum [5]
Mix Network	No	Yes [41]	Integrity; Authentication [1,41]	Yes [1]	Bitcoin [1]
Onion and Tor Routing	Yes [42–44]	Yes [43,44]	Integrity [42,44]	No	Loki Blockchain [42]
K-anonymity	Yes [45]	No [46]	Integrity [47]	Yes [45]	N/A
L-diversity	Yes [48]	No [49]	Confidentiality [50,51]	Yes [48]	N/A
T-closeness	Yes [52]	Yes [52]	N/A	Yes [52]	N/A
Identity-based Encryption	Yes [53–55]	Yes [53,54]	Integrity [55]	Yes [53,55]	Bitcoin and Namecoin [55]
Attribute-based Encryption	No	Yes [56,57]	Confidentiality [56]	Yes Multiple parties to generate public/private keys [56]	Private [56]
Symmetric key Cryptography (DES, AES, Blowfish, RC5 etc)	Yes [58]	Yes [58,59]	Integrity [58] Confidentiality Non-repudiation [59]	No	Bitcoin [58,59]
Asymmetric key Cryptography (Elgamal, RSA, DSA, Elliptic curve, etc.)	Yes [60]	Yes [60,61]	Authentication [60,61]; Confidentiality [61]	Server to manage user identity and password [61]	Permissioned BC [61]
Homomorphic Encryption	No	Yes [62,63]	Confidentiality [62,63]	Yes [63]	Bitcoin [62]
Hash Function (MD5, SHA-1)	Yes [64]	Yes [64]	Integrity; Confidentiality [64]	No	Bitcoin [64]

According to those requirements, we are now able to evaluate the main privacy-preserving and security mechanisms that support transactions anonymity, enhance data anonymization and user identity protection as shown in Table 1, and describe how they apply to blockchain.

Ring signature provides data confidentiality and user identity privacy. A ring signature is a type of cryptographic digital signature that relies on a group of users (called a ring) provided with asymmetric keys to sign messages. Once a message has been signed, it can be decrypted with the ring signature only, and we cannot identify the actual signer of the message within the group. As an implementation example, the Ring CryptoNote protocol hides transaction details (e.g., amount, origin, destination) in the decentralized cryptocurrency Monero [39]. However, ring signature involves a TTP to manage user identities and the cost of both generation and verification of ring signature increases due to digital certificates [37]. Identity-based ring signature overcomes these issues and enhances users' privacy. It also prevents against full key exposure attack [65].

Zero knowledge proof (ZKP) allows one entity (the prover) to prove to another entity (the verifier), that a given value is true, without disclosing any information apart from that the proof itself is correct. Zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge) is a ZKP that allows proving correct computation of information without any interaction between both parties (prover and verifier). ZKP-based solutions are particularly interesting for data integrity and authentication without revealing anything as they provide proof of a statement without revealing that particular statement. They also preserve anonymity for sensitive data and do not rely on TTP. However, they are computationally quite intensive as compared to other solutions to validate and generate proofs [5].

Mix networks take input data from multiple users and forward them to the next destination according to a different time frame. They basically randomize the order of the transaction data, so that it is harder for an attacker to trace end-to-end communication. However, the mix network becomes a TTP and has latency issues as compared to other mechanisms. One example of blockchain implementation is coinJoin [66]. On the other hand, onion routing and tor routing enable anonymous communication to provide protection on user's data without any TTPs.

K-anonymity concept was introduced in 1998 to prevent identity disclosure [67]. It preserves the privacy of sensitive information by hiding a user's identity in a given data set [68]. It enforces integrity on data without any TTPs. However, it is insufficient to protect attribute privacy in transactional data due to lack of diversity in the data.

L-diversity complements k-anonymity. It augments the diversity of users' sensitive information in the data set to achieve higher privacy. It effectively protects the user's identity and guarantees the confidentiality of the data [50,51].

T-closeness ensures better protection of identity privacy and transaction data privacy than existing data anonymization techniques [52]. It enforces privacy without the involvement of TTPs [52]. The principle of t-closeness is simple and easy to understand as it enforces a maximum distance between sensitive attributes in a data set to ensure data protection.

Identity-based encryption is a technique of public-key encryption that allows a user to generate a public key from its unique identity (e.g., IP address or email address) [69]. This scheme may involve a TTP such as a private key generation algorithm to derive corresponding private keys. It protects user's identity and guarantees the integrity of the message. It entails a longer public key, message size, and memory overhead than attribute-based encryption.

Attribute-based encryption derives from identity-based encryption [70] in which only a user can decrypt the ciphertext if the user attributes (e.g., hometown) satisfy the access rules defined by the owner of the system. It enforces confidentiality on data and fulfills transaction data requirements.

Symmetric key cryptography is efficient to enforce integrity, confidentiality, and non-repudiation on data without TTPs. Symmetric key cryptography refers to use the same key for encryption and decryption of messages to guarantee that only authorized users do so. Key should be shared through a key exchange protocol to protect further communications. There are several symmetric key algorithms such as DES, Blowfish, and AES [71]. In our context, the main drawbacks of symmetric cryptography are communication overhead and the required key exchange protocol [58,59].

Asymmetric key cryptography, also called public-key cryptography (e.g., RSA [72]) is efficient for both user identity and transaction data anonymity due to its security properties e.g., confidentiality and authentication on data [61]. It is based on a couple of keys (called public and private) to encrypt and decrypt messages. Only the public key can be shared. It relies on digital signatures to verify that messages come from a particular sender (if encrypted with the sender's private key) or to guarantee that only a specific receiver can read it (if encrypted with the receiver's public key). However, it is computationally more intensive compared to symmetric key cryptography, which makes it less suitable for encrypting large amounts of data. In practice, it typically requires a TTPs to manage user's identities, although key management can be realized individually.

Homomorphic encryption [73] ensures data confidentiality and provides confidentiality on transaction data, and therefore user anonymity. It encrypts data and allows anyone to perform actions on the encrypted version without accessing the encryption key. However, it depends on TTPs and has a high computation overhead.

Hash Functions are most often utilized to confirm data integrity over a copy of original data and a signature by comparing with the one generated from the copy without any intermediate party. A hash function is a one-way function that transforms input data into fixed-length data. It has a low computation overhead as compared to other available solutions.

4. Attacks and Countermeasures

After giving an overview of blockchain requirements and existing technology, it is appropriate to look at the attacks and how they are handled. Therefore, in the following, we overview the main existing attack vectors on blockchain, including remote attacks and attacks related to data disclosure. We provide a classification of existing attacks and match associated countermeasures as follows:

- **Network attacks**, and therefore network security, have been historically existing since the beginning of distributed computing. Network security is typically defined as a subcategory of security [74], and as such network security attacks and countermeasures apply to blockchain implementations as well. Typical countermeasures include designing secure protocols, applying symmetric/asymmetric encryption, filtering and monitoring networks.
- **Blockchain attacks** specifically exploit weaknesses in the blockchain operation to be successful. Typical countermeasures involve integrating secure protocols and algorithms into the blockchain operation and extending the original blockchain design.
- **Privacy attacks** exploit available privacy-sensitive data to disclose additional information. Typical countermeasures rely on generating fake transactions (e.g., mixins in Monero) and use statistical models to disable data exploitation.

Figure 2 gives a detailed overview of the different attacks described in the following, and the associated countermeasures, including references to the related literature.

4.1. Network Attacks and Countermeasures

Network attacks leverage network design and properties to be successful. They can be categorized as active, where a malicious user alters or deletes information, and passive, where a malicious user listens to the data passing on the network. In the following, we detail the countermeasures that apply to the network attacks. Most of them rely on secure protocols such as TLS based on asymmetric cryptography (Public Key Infrastructure–PKI) or symmetric cryptography, whereas others are implemented at the network level, for example with network filtering, monitoring, or routing. We categorized **active attacks** as follows:

- **IP spoofing attack:** In which an attacker uses (or spoofs) someone else's identity and communicates with this identity to get benefits from it [75].
- **Countermeasures:** Typically, it is possible to prevent such attacks by filtering network traffic in order to stop the delivery of spoofing packets. In [76], routing-based mechanisms are discussed to deal with IP spoofing attacks. These solutions are implemented by routers or end-hosts.
- **Message modification attack:** A malicious user modifies the contents of a message being transmitted [75].
- **Countermeasures:** To deal with message modification attack, using a hash function is a typical solution [77].
- **Message forgery attack:** The goal of this attack is to create false information on the network. Sending a spam email by using the identity (e.g., name or email address) of another entity is an example of this attack [75].
- **Countermeasures:** Cryptographic mechanisms such as asymmetric key cryptography can be used to mitigate a message forgery attack. Sender and receiver have their own public and private key pair to encrypt and decrypt the message [78]. It ensures that the receiver receives the original message and proves the identity of the sender.
- **(Distributed) Denial-of-Service (DoS) attack:** The adversary sends enough traffic so that the target is unable to process it and remains unavailable to legit users. Distributed denial-of-service (DDoS) is a distributed version of the DoS attack, where multiple nodes are orchestrated to attack the target simultaneously [6].
- **Countermeasures:** In [79], the authors discuss some prevention techniques against DDoS attack. It includes an intrusion prevention mechanism to disable unused network services which are not needed, install security patches to the host machine to prevent vulnerabilities in the system, use firewalls, and dynamically switch the IP address of the active server within a pool of other servers.
- **Man-in-the-middle (MITM) attack:** An attacker intercepts communication and acts on behalf of one end to collect valuable data [80].
- **Countermeasures:** One proposed solution to avoid man-in-the-middle attacks is to monitor the network using an intrusion detection system. It observes the behavior of an attacker and gives an alert if an attacker tries to take control of network traffic [81].
- **Time delay attack:** The goal of this attack is to slow down the network traffic. An attacker tries to control the network traffic, which leads to the delay in the transmission of message [82]. Delay attack can be categorized in the following two attacks: BGP hijacking and Liveness attack.
- **Countermeasures:** To prevent time delay attack, Arman et al. [83], proposed a method to track and estimate the time delay during communication on the network. If the measured time delay is more than the allowed time delay, then the system issue an alarm signal to stable the system. It is a simple and inexpensive way to secure the system from time delay attacks.

  - **Border gateway protocol hijacking (BGP) attack:** BGP is a routing protocol of the Internet. It helps to forward IP packets to their destination. The attacker conducts BGP hijacking to control the network traffic on the blockchain, which leads to delay of network messages [84].
  - **Countermeasures:** Xing et al. [85], proposed BGPcoin based on Ethereum blockchain, a BGP framework that is generated by a smart contract in the network Ethereum. It provides a transparent allocation of sources (e.g., IP prefixes). It also checks source assignments using blockchain to enhance system security.
  - **Liveness attack:** This attack [86] causes the delay of target transaction time on blockchain. It works in three phases, such as (1) attack preparation phase, in which the attacker creates a private chain longer than the public chain, (2) in-transaction denial phase, where the attacker tries to privately keep the block that contains target transaction, (3) in-blockchain phase, where the attacker will integrate their private blocks into the public chain to slow down the growth rate of the public chain.
  - **Countermeasures:** To resolve the liveness attack Chenxing et al. [87] proposed the conflux consensus protocol that allows fast confirmation and ensures the consensus progress. It is a secure, scalable, and decentralized blockchain platform.

On the contrary, with passive attacks, an attacker monitors the network and attempts to gain sensitive information about a target. We identified the most well-known passive attacks as follows:

- **Traffic analysis attack:** An adversary tries to observe the communication route between the sender and receiver. The goal of this attack is to analyze the network communication and from this analysis gather useful information to execute an attack [88].
- **Countermeasures:** Onion routing provides a solution to traffic analysis as it allows to transfer data from a source through multiple onion routers before the data reaches its destination, thus guaranteeing anonymity. Each message has various layers of encryption, where each layer links to one onion router. It prevents the attacker to identify the endpoints of communication [89].
- **Eavesdropping attack:** The aim of this attack is to listen to private information over the network [90].
- **Countermeasures:** Cryptographic techniques such as symmetric key encryption can be used to protect confidential communication against eavesdropping attack. It limits data access only to the authorized user [91].

As a summary, we observe that most cryptographic mechanisms that are used to ensure security on sensitive data rely on key management solutions. Symmetric cryptography is computationally efficient, but is sensitive to attacks during key agreement or key distribution. For instance, the well-known Diffie-Hellman (DH) scheme is vulnerable to the man-in-the-middle attack. Efficient key agreement and distribution are still being researched intensively, though improvements like Station-to-station (STS) addressed the issues by authenticating both parties of the exchange through public-key cryptography. Although DDoS attack remains dangerous, it is generally prevented by implementing firewalls on routers, where networks are interconnected.

4.2. Blockchain Attacks and Countermeasures

Blockchain has gained massive attention for both industry and academia, however, it is subject to specific attacks. In the following, we classify them and present possible countermeasures.

- **Malleability attack:** It consists of changing the unique ID of a transaction before confirmation on the network, so an attacker may pretend that a transaction did not occur [92]. This attack was successfully executed on many blockchains due to the nature of the ECDSA digital signature scheme. Suppose Alice sends a payment transaction to Bob with a $txID$. Before the transaction is accepted into a block, the $txID$ is changed to $txID'$. Bob receives the payment regardless but now Alice not knowing about the change, cannot know if Bob received the funds. This way Bob can ask Alice to resend the funds until eventually, Alice notices. The vulnerability has gained a lot of attention once it was cited by Mt Gox exchange as a key reason for suspending withdrawals.
- **Countermeasures:** Malleability can rarely be exploited to the attacker's benefit. At present, it is more of a theoretical problem than a real one as the attacker stands very little to gain. The threat is addressed on the application layer with wallet software not trusting zero-confirmation transactions by either displaying them as pending or not at all. Additionally, changing a transaction ID is not trivial especially now that users' digital signatures are properly checked by clients in most protocols [93]. However, with the influx of new blockchain implementation it is important to be aware of the potential threat.
- **Nothing at Stake Problem:** Proof of stake-based blockchains require staked nodes to attest, validate, and verify blocks. Attestations are used to decide, which of the two conflicting blocks is accepted. Proof of stake nodes can collude to vote on conflicting chains. Once their private chain gains more attestations for a series of blocks, they can attempt to convince the rest of the network to fork, and accept their version of the chain. In a proof of stake setting, this is referred to as voting on conflicting blocks. These issues stem from the so-called Nothing-at-Stake problem [15]. A node that receives two conflicting blocks must vote and decide which one to accept. It is always in the node's best interest to vote for both blocks to earn the reward for contributing to the consensus, whichever block is accepted.
- **Countermeasures:** To mitigate this problem, proof of stake-based consensus mechanisms introduce game theoretical concepts that penalize any nodes that are discovered to vote on conflicting blocks. The state of the art protocols such as Casper [94], and GHOST [95] or recently a combination of both [96] propose modified fork-choice rules and block finalization protocols. Casper, for example, requires validators to be bonded i.e. to commit Ether to a smart contract. Validator sets continuously get randomly shuffled. Withdrawing their bond would take a long time (months). It creates a long time window within which other votes will be collected. This allows for a long enough period where the validators bond can be destroyed (called slashing in Casper) by another validator set once all votes are collected.
- **Majority attack** A malicious actor tries to obtain a majority representation in the network to compromise the blockchain either on the consensus level, or the network level. On the consensus level, this can lead to a double-spending attack [97] (described below). The severity of this attack largely depends on the network. In proof-of-work-based blockchain, this can lead to a double-spend using a selfish mining attack. The malicious party submits a large payment transaction, and in return receives goods or services. The transaction is confirmed on the public chain. Once the goods/services are received, the malicious party reconnects their network to the public chain, and propagates their version of the blockchain. Their version of the chain will be longer due to higher hash-rate, forcing the public chain to revert a few blocks, reverting also, the payment transaction. The extent of the thread largely depends on the underlying protocol, its consensus algorithm, and its fault tolerance.
- **Countermeasures:** In [97], a merged mining technique is developed to protect against majority attack, in which, two different crypto-coins are combined to be mined at the same time. This technique enhances security, because miners contribute to the hash-rate of both crypto-currencies and miners can mine multiple blocks simultaneously. However, this process is complex and merged crypto-currencies must rely on the same hashing algorithm.

- **Selfish mining attack:** Also known as block withholding attack, occurs when attacker keeps track of their own private chain, separate from the public chain [98]. The attacker mines on the private chain and keeps private blocks, then use the majority attack. This attack is mostly applicable to Proof of Work blockchains if a malicious actor is able to sustain majority mining power for more than one block. Upon reconnecting with the public network, they would have mined more blocks, forcing the rest of the network to accept the longer chain. Transactions on the public network would hence be reversed, potentially opening the door to a double-spend attack. In proof of the stake-based consensus algorithm, this idea applies. However, the condition is different as the malicious party needs to control the majority of validator nodes. They then vote for both the blocks on the public network and the private network. However, proof of stake chains can mitigate this by finalizing blocks, and not simply following the "longest chain wins" rule of the Nakamoto consensus.
- **Countermeasures:** To resolve selfish mining attacks Siamak et al. [99] proposed Zeroblock mechanism, in which, if a block does not generate within fixed interval time, then this block will be rejected by all other miners. However, maintaining time synchronization is not trivial. The problem becomes much harder when nodes display byzantine behavior. Blockchains with PoS consensus mechanisms rely on different fork choice rules, and block finality protocols to prevent large forks such as Casper FFG [94].
- **Eclipse attack:** In which an adversary tries to control the incoming and outgoing connection of a node and mining power in the network. It allows an attacker to manipulate the information, and hence nodes operation and launch an attack [100]. The attacker runs a modified version of the protocol on multiple instances in an attempt to convince one or multiple nodes to connect to his malicious nodes. Once the attacked nodes only maintain connections to nodes operated by the attacker, the nodes are eclipsed from the rest of the network. The attacker can arbitrarily filter the message passing possibly even censoring transactions from eclipsed nodes. The most recent successful attack was on the Monero network [101], where the attackers ran multiple nodes by renting cloud instances. They exploited the network code that favored outgoing connections to nodes running behind the same IP address. This allowed the attacker to gain network share by eclipsing honest nodes with malicious ones. The attack ended after a few weeks with a fix of the P2P code in Monero.
- **Countermeasures:** In paper [100], proposed solutions to mitigate eclipse attack is to use a white-list (e.g., known miners) and disable incoming connections in the network. It prevents new nodes to join the network. Another paper [102], proposed a mechanism against eclipse attack. The basic idea behind this mechanism is to bind the incoming and outgoing connections of the overlay network and allow a node to choose neighbors based on performance metrics. This countermeasure is effective because it is difficult for an attacker to control the nodes in the network.
- **Sybil attack:** the attacker tries to control the peer network by participating with multiple identities. It tries to gain the majority of influence on the network [8]. In the context of blockchains, this attack is applicable in a proof-of-stake setting. All proof of work-based consensus protocols relies on voting schemes to decide the fork choice rule. With the majority of the consensus nodes, an attacker can win the vote for invalid blocks. However, due to staking requirements, the attack is rarely feasible. On the one hand, should such an attack be successful, all trust in the chain would be lost, and the value of the attacker's staked coins would drop drastically. On the other hand, the investment needed to acquire such a large influence is usually too high.
- **Countermeasures:** To prevent Sybil attack George et al. [103], proposed Xim, a two-party mixing protocol that is compatible with bitcoin and virtual crypto-currencies. It is a decentralized system that allows participants to find partners anonymously to hide coin

exchange by mixing. It does not rely on TTP to choose partners for mixing, and enhances security because the malicious user cannot identify the evidence of participants that mix up.

- **Double spending attack:** It consists of spending digital currency more than once. This is what blockchain primarily solves, however an attacker may replicate a digital coin and use it in another transaction [92].
- **Countermeasures:** To prevent the double-spending attack Hyunjae et al. [104] proposed a recipient-oriented transaction method based on a private blockchain. The proposed mechanism ensures the privacy of recipients and mixes the incoming transactions to protect them from attackers. However, in public permission-less networks, double-spends are only possible by exploiting the consensus algorithm. It allows the transaction's receiver to verify the validity of transactions before adding them to the block. The actual attack vector varies depending on the type of consensus mechanism.
- **Malicious smart contract:** It facilitates, for instance, leakage of privacy-sensitive information (e.g., email address into a smart contract is publicly visible) and password theft (e.g., a fair exchange between two parties can only be done after receiving a valid password) [105]. In [106], the authors explain the example of password theft. Dealing with malicious smart contracts remains an open research problem.
- **Countermeasures:** Slither is a fast and reliable security mechanism to detect bugs in a smart contract. The main objective of this mechanism is to understand code and detect vulnerability automatically [107]. In [108], the authors proposed the SmartCheck vulnerability analysis framework to protect the code of solidity smart contracts. It helps to identify the vulnerabilities in smart contracts and the reason of these vulnerabilities with recommendations. Ahmed et al. [34], proposed the Hawk framework, based on zero-knowledge proofs to enforce transactional privacy in a smart contract. It allows a user to write a private smart contract and does not store transactions publicly on blockchain.

  - **Distributed autonomous organizations (DAO) attack:** It occurs with the help of a malicious smart contract on Ethereum. In this way, the attacker injects some malicious functions in the smart contract, such as withdraw function, call send to Ether and steal all the Ethers from DAO [109].
  - **Countermeasures:** As DAO attack exploits calling a reentrant function in the smart contract execution, its solution consists in making sure that the code of the smart contract is not callable in such a way [110].
  - **Wallet theft attack:** Usually, the user's private keys are generated and maintained by the user with the help of wallets. In paper [111], authors claim that ECDSA (Elliptic Curve Digital Signature Algorithm) scheme does not maintain user's privacy because it is unable to create enough randomness during the signature process, through which an attacker can steal the user's private key.
  - **Countermeasures:** To deal with wallet theft attack Weiqi et al. [112] proposed to use the Trust zone mechanism present on ARM processors that creates a secure execution environment (SEE) to guarantee a safe and reliable environment for the execution of sensitive processes. This is realized with non-maskable interrupts (NMI) and guarantees complete isolation from the operating system, using a secure memory zone and storage to protect user's private key, wallet's address, and transaction verification process.

- **Blockchain Poisoning:** It is an attempt to store illegal files (e.g., malware, malicious content, and privacy information, etc.) in the free space for smart contracts of blockchain. A malicious user can force blockchain nodes to download malicious files which leads to other attacks in the blockchain such as DoS (see Section 4.1). Another kind of poisoning consists of spamming many transactions and indexing the outputs so when those outputs are used as decoys on other transactions, one can consider them decoys and increase the odds of tracing the real outputs, leading to the statistical modeling attack [113].
- **Countermeasures:** blockchain poisoning attack can be prevented by introducing a minimal fee determined by its resemblance with existing contracts in which the minimal fee reduces if the number of similar contracts increases. The idea is to calculate the fee of all contracts which are used on blockchain and it becomes higher if there is no similar contract. In this approach, the cost to send a transaction relies on the number of similar contracts. Because the number of malicious smart contracts related to falsified transactions is small in the blockchain, then, a malicious user would need to pay a higher fee, which reduces the likelihood of such an attack to occur [113].

In summary, due to its public nature, blockchain is vulnerable to several attacks. In order to tackle these attacks and improve blockchain security and performance, we have discussed possible countermeasures. Among these, smart contract attack is a growing concern due to bugs in code, which can lead to loss of cryptocurrency and privacy leakage. Writing secure smart contracts can be difficult due to various rules, as well as platform vulnerabilities and limitations. Research towards securing the execution of smart contracts on blockchain-based systems is still an open question.

4.3. *Privacy Attacks and Countermeasures*

As a complement to previous categories, we identified the most relevant privacy attacks and their countermeasures, that focus on exploiting available data to gain knowledge of privacy-sensitive information.

- **Homogeneity attack and background knowledge attack:** Homogeneity attack discloses privacy-sensitive information and breaks privacy when all values of a quasi-identifier attribute (e.g., zip code, date of birth, and gender) are similar in a given table [114]. Background knowledge attack occurs when an adversary has a background knowledge of a quasi-identifier to reveal the user's sensitive attributes [67].
- **Countermeasures:** K-anonymity is based on quasi-identifier attributes to hide user's identity in a dataset [45] by providing at least k identical records in the data set for each quasi-identifier attribute. Thus, even if an attacker accesses the data set, he could not identify the real identity of a person due to a similar record in the data set. However, background knowledge attacks can still link outside information to disclose sensitive information.

  L-diversity extends k-anonymity to prevent homogeneity attack and background knowledge attack as it normalizes data distribution. It improves the diversity of user's sensitive information in the data set to achieve higher privacy. It ensures data privacy without disclosure of sensitive attributes. However, it is vulnerable to skewness and similarity attacks [67].
- **Skewness attack and similarity attack:** Skewness attack occurs when the values in the table have a non-uniform distribution, a malicious user obtains a sensitive value based on its higher frequency distribution over a subset of the data [115]. A similarity attack happens when the values of sensitive attributes are distinct but semantically similar, and the attacker uses the meaning of data to infer missing information [49].
- **Countermeasures:** The t-closeness approach [49] improves privacy by integrating k-anonymity and l-diversity. It provides protection against sensitive attribute disclosure. This approach works properly if the distance between two distributions in the table should not more than the given threshold value. It does not work properly if the distribution of sensitive values in a class is not close to the distribution of sensitive values in the overall dataset.

- **Statistical modeling:** Such attacks are specific to blockchain implementations that use decoys to prevent transaction identification, such as Monero. Statistical modeling, coupled with a heuristic approach for decoy elimination is believed to be the attack vector CipherTrace is using to attempt to trace Monero transactions at the time of writing [116]. The accuracy of such models can be increased significantly when coupled with other privacy attacks such as blockchain poisoning.
- **Countermeasures:** Statistical modeling can be prevented with binned mixin sampling, which changes the current mixin sampling process. The idea is to combine outputs into groups of some fixed size in the Monero blockchain, called bins. Each input transaction is referencing an output transaction in a bin, either as a mixin or spend. It estimates the real spend-time distribution, and then sample mixins according to this distribution. It helps to prevent the traceability of transaction input and disclosure of mixin sampling distribution [117].

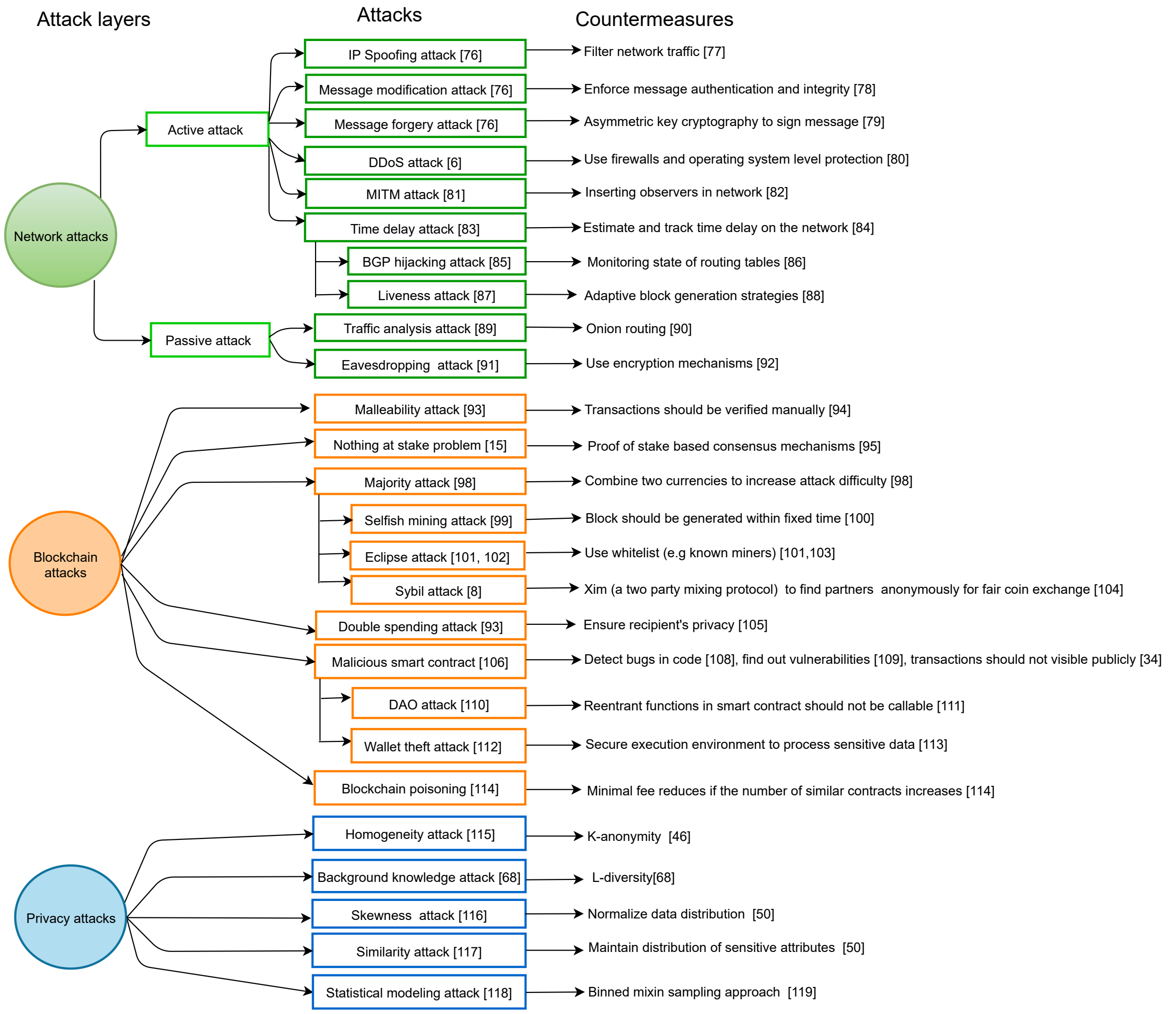
Attack layers
Attacks
Countermeasures
Network attacks
Active attack
Passive attack
IP Spoofing attack [76]
Filter network traffic [77]
Message modification attack [76]
Enforce message authentication and integrity [78]
Message forgery attack [76]
Asymmetric key cryptography to sign message [79]
DDoS attack [6]
Use firewalls and operating system level protection [80]
MITM attack [81]
Inserting observers in network [82]
Time delay attack [83]
Estimate and track time delay on the network [84]
BGP hijacking attack [85]
Monitoring state of routing tables [86]
Liveness attack [87]
Adaptive block generation strategies [88]
Traffic analysis attack [89]
Onion routing [90]
Eavesdropping attack [91]
Use encryption mechanisms [92]
Blockchain attacks
Malleability attack [93]
Transactions should be verified manually [94]
Nothing at stake problem [15]
Proof of stake based consensus mechanisms [95]
Majority attack [98]
Combine two currencies to increase attack difficulty [98]
Selfish mining attack [99]
Block should be generated within fixed time [100]
Eclipse attack [101, 102]
Use whitelist (e.g known miners) [101,103]
Sybil attack [8]
Xim (a two party mixing protocol) to find partners anonymously for fair coin exchange [104]
Double spending attack [93]
Ensure recipient's privacy [105]
Malicious smart contract [106]
Detect bugs in code [108], find out vulnerabilities [109], transactions should not visible publicly [34]
DAO attack [110]
Reentrant functions in smart contract should not be callable [111]
Wallet theft attack [112]
Secure execution environment to process sensitive data [113]
Blockchain poisoning [114]
Minimal fee reduces if the number of similar contracts increases [114]
Privacy attacks
Homogeneity attack [115]
K-anonymity [46]
Background knowledge attack [68]
L-diversity[68]
Skewness attack [116]
Normalize data distribution [50]
Similarity attack [117]
Maintain distribution of sensitive attributes [50]
Statistical modeling attack [118]
Binned mixin sampling approach [119]

Figure 2. Classification of attacks and countermeasures.

As a summary, the main highlight is the diversity of attacks that blockchain-based solutions face at the network, blockchain and privacy levels. Therefore, today's solutions must combine existing techniques to guarantee reasonable protection from this diversity of potential breaches. In particular, data confidentiality, privacy, and smart contract execution are vital challenges for blockchain implementations. We discuss this further in the following section.

5. Evaluation and Discussion

This section starts with a comparative analysis of privacy and security mechanisms and follows with a set of recommendations for design choices that should be considered when designing blockchain-based solutions and research directions based on the identified open challenges.

5.1. Comparative Analysis

In order to compare existing work, we deem it appropriate to examine the following criteria: complexity, scalability, size overhead, memory cost, efficiency, computation overhead, communication overhead, and latency. Table 2 gives a global overview of existing technologies with respect to these criteria. We analyze in detail each privacy and security mechanism of Table 2, including their main advantages and limitations, in the following. We mentioned "N/A" in the table if we did not find detailed literature for specific criteria.

Complexity is a general indicator that helps us assess how costly a solution will be when adopted, based on its mathematical complexity. Scalability gives an insight into how the compared solutions are suitable to large-scale contexts. Size overhead reflects the cost to apply the solution in terms of storage space. Memory cost shows an amount of memory that is required to apply the solution, which helps us to measure solution performance. Efficiency determines the speed and average execution time taken by a mechanism to complete the given task, which affects the solution performance. Computation overhead shows the cost of privacy and security mechanism operations, which facilitate us to measure the number of steps compulsory to solve the problem. Communication overhead is measured by the number of bytes in every message sent during communication over the network and latency refers to how much time it takes to transfer information to its destination. We follow with Table 3 that provides a detailed discussion on the advantages and drawbacks of the previously mentioned privacy and security techniques to guide further research.

The tables show that ring signature [37], is relatively efficient and works with very low latency. Security properties (e.g., confidentiality and authentication) and data transaction protection [4,39] are also provided without reducing signature size overhead. However, key management and long signature size are drawbacks as the number of key couples is linear to the number of ring participants.

Zero-knowledge proof [118], is a promising solution to validate transactions or to ensure privacy of the data in the blockchain. It is good to provide anonymous authentication and is secure against man-in-middle attack [119]. However, zero-knowledge proof is not efficient for large-scale systems because it requires high computation time and memory resources to create and validate proofs [5].

Mix networks, as proposed in [1,41], achieve both efficiency and scalability. It has a low size overhead because the size of input data is linear to the number of mix servers. The proposed scheme is unable to reduce complexity, latency, memory, and computation overhead due to the high cost of generating an input cipher text and cost of encryption keys used by mix servers.

In [43,44], the Tor mechanism is presented to improve the identified limitations of mix network and achieves low memory cost, data size, computation, and communication overhead. Their motivation was to reduce latency in the anonymity system. However, Tor is not scalable because it cannot handle a large number of nodes or servers. Using Tor entails reducing network speed as compared to normal internet and it does not ensure data security outside of the network. Along with this, onion routing protocol [42], enables sending end-to-end encrypted messages with minimal disclosure of user metadata. The focus is put on scalability and efficiency in communication. However, memory cost and communication overhead are high.

From the data privacy perspective, k-anonymity technique is scalable and works without any latency [120]. It is easy to implement and risk of re-identification is reduced when the diverse values are high. However, it is not efficient because of the long processing time [121,122]. It is prone to the background knowledge attack and homogeneity attack.

In [123], the authors proposed the l-diversity mechanism that has a low size overhead. Computation overhead is also low due to the same frequency of sensitive attributes in a given dataset. However, it is not efficient due to the chance of attribute disclosure. It also fails to prevent the skewness attack and similarity attack.

The authors in [68], presented the t-closeness mechanism which is not scalable to handle large datasets. T-closeness suffers from higher time complexity for larger datasets because running time increases in proportion to the number of inputs. It is therefore a complex computational process to enforce t-closeness.

In [53], authors proposed an identity-based encryption mechanism that is simple and efficient because it allows verifying the validity of ciphertexts publicly and reduces the ciphertext size and decryption time as compared to existing approaches (e.g., adaptive chosen ciphertext (CCA2) attacks). The solution relies on an encapsulation algorithm to enhance security against selective-identity attacks and to reduce the public and private key sizes as compared to typical encryption systems. However, key management is a limitation of the identity-based encryption technique because of the private-key generator which is used to generate user's private key. A secure channel between a user and private-key generator is needed to transmit the private key. One main limitation of Identity-based encryption is a single point of failure, because if a private-key generator is compromised then it means that the entire system is compromised.

Attribute-based encryption [124], allows a user to encrypt and decrypt data based on the user's attributes. It is scalable and efficient for large size encryption and enforces confidentiality on data [125]. The identified limitations of attribute-based encryption are high time complexity and computational cost in terms of key generation, encryption, and decryption [57]. Another drawback of the attribute-based encryption mechanism is high communication overhead due to a larger ciphertext size. In the future, attribute-based encryption with fixed ciphertext size might be a possible solution to overcome identified limitations.

In [58], the authors propose a blockchain-based credit network that provides privacy and security without TTP using symmetric key cryptography. They present security and scalability analysis to demonstrate advantages in terms of transaction privacy, achieve low communication overhead, maintain concurrent transactions, and identify attackers in transactions. The major disadvantage of the proposed approach is public transactions and higher time complexity.

Asymmetric key cryptography ensures secure communication between sender and receiver by using public and private keys [126].It has high computational time overhead because of key generation time and encryption/decryption time, which makes encryption/decryption more complex for large size of data [127]. Asymmetric key cryptography is memory consuming and computationally intensive, leading to long execution times [128]. Lastly, key management remains an issue.

Table 2. Comparative analysis of privacy and security approaches.

Mechanism	**Complexity**	**Scalable**	**Size Overhead**	**High Memory Cost**	**Efficiency**	**Computation Overhead**	**Communication Overhead**	**High Latency**
Ring Signature	Low [4,37,38]	No	O(n) ring signature grows linear with group size [4,37–39]	No [4,37,38]	Yes [4,37,38]	- High [4] - O (n + 1) n is group size of ring signature [129,130] - O(n) [38]	N/A	No
Zero Knowledge Proof	High [118]	Yes [119]	Low [5]	Yes [5]	No [5]	Expensive [5]	N/A	Yes [5]
Mix Network	O(snN) s is length of plain text parallel to input [41]	Yes [1]	O(n) Input entities is linear to mix servers [41]	Yes [1]	yes [41]	O(N + n) All inputs [41]	N/A	Yes [1]
Tor Routing	Low [44]	No	Low [44]	No [44]	Yes [44]	Low [44]	N/A	Low [44]
Onion Routing	Low [42,43]	Yes [42]	High [42]	Yes [42]	Yes [42]	High [42]	Yes [42]	High [43]
K-anonymity	Very low [68]	Yes [120]	Low [121]	No [121]	No [122]	Low [121]	N/A	No [121]
L-diversity	Low [68]	Yes [123]	Low [123]	Low [122]	No [122]	Low [123]	N/A	No
T-closeness	Very high [68]	No [68]	Low [52]	Low [52]	Yes [52]	Low [52]	N/A	No [131]
Identity-based Encryption (session key, and encapsulation algorithm)	Low [53]	No	- O(n) public-key size and encryption time grows linearly [53] - n-bits messages [54]	Yes [55]	Yes [53,54]	weak [53,54]	N/A	No
Attribute-based Encryption (bilinear Diffie-Hellman)	High [57]	Yes [125]	Low [57]	No [57,125]	Yes [124,125]	High [125]	High [125]	No

Table 2. *Cont.*

Mechanism	Complexity	Scalable	Size Overhead	High Memory Cost	Efficiency	Computation Overhead	Communication Overhead	High Latency
Symmetric key Cryptography (DES, AES, RC5 etc)	Low [58]	Yes [59]	Low [72]	Yes [58]	Yes [58]	Low [132]	Yes [58,59]	Yes [58]
Asymmetric key Cryptography (Elgamal, RSA, DSA, etc.)	High [127]	Yes [61]	High [133]	High [61,128]	Yes [60,128]	High [127]	Yes [134]	No [134]
Homomorphic Encryption	Low [62]	No	Low [62]	No [62]	Yes [62,63]	Low [62]	Low [62]	No
Hash Function (MD5, SHA-1)	O(N), N is a size of data [135]	No	High [136]	Yes [136]	Yes [135]	Low [136]	N/A	No

Homomorphic encryption allows mathematical operations on encrypted data by untrusted third party and ensures the privacy of sensitive data [137]. In [62], the authors present a transaction-privacy bitcoin framework by using homomorphic encryption that is efficient and less complex because it has small unit of time in terms of key generation, encryption, verification, and decryption. The proposed framework is secure and prevents the active attack, double spending attack, and passive attack but it does not ensure identity privacy. This technique has low computation overhead in terms of modular operations on data. The communication cost between the user and the third-party server is low.

In [135], the authors discussed the Hash function that is simple and efficient. The complexity of hash function is O(N), where N is the size of the input data. It is computationally intensive due to large hash size for SHA-256 and SHA-512 algorithms [136].

In addition to privacy challenges on blockchain, the reader is referred to additional survey papers [1,5], where authors summarize the limitations of consensus algorithms (e.g., proof of work, proof of stake, and delegated proof of stake etc.) and privacy challenges (e.g., transaction linkability, private key management, and malicious smart contract etc.) as well as their corresponding countermeasures. However, as it is proved in literature, full anonymity is not ensured in the bitcoin blockchain. Related to cryptographic aspects, another identified limitation is to deal with the high computational cost of ZK-SNARKS on the blockchain.

In [60], a simulation has been conducted to test the computation cost of an authentication scheme using Elliptic Curve Cryptography to achieve efficiency, key security, and user anonymity. The experimental results show that it is secure against replay attacks and man-in-the-middle attacks. The main disadvantage of this approach is its complexity because of Elliptic Curve Cryptography.

In [61], the authors propose a blockchain-based system to send data anonymously and securely. The proposed system achieves scalability and reduces communication overhead without latency. Security properties (e.g., data confidentiality and authentication) and user's privacy are also provided but the proposed system is limited to private permission blockchain. However, the proposed system presents several drawbacks related to system complexity, computational cost, and high storage overhead.

Authors in [64] propose a system that ensures identity privacy, unlinkability during a fair exchange, and security on data. The proposed system is based on decentralized blockchain and smart contracts are used to enforce fair transaction exchange between parties. It reduces memory cost. However, it is not scalable and requires high communication overhead. Analysis of complexity and attack risks are also not provided. Table 3 summarizes our findings that are detailed below.

Table 3. Advantages-limitations of privacy and security mechanisms.

Privacy and Security Mechanisms	Advantages	Limitations	Attack Risks
Ring Signature	Flexible, Efficient; Does not depend on TTP.	Computational infeasible; Could not identify actual signer of transaction.	Attribution attack.
Zero Knowledge Proof	Protect identity and transaction data.	Efficiency; Need high computational time; Size of proofs.	One user pretends to be another.
Mix Network	Efficient; Difficult for an attacker to trace communication.	Require TTP as mixing server.	Active attack; Sleeper attack.
Onion Routing	Enforce security on data.	High communication overhead.	Replay attack.
Tor Routing	Efficient; Protect user's identity online.	Does not ensure data security; Slow network speed.	Attack on client through internet browser and external software e.g., Flash etc
K-Anonymity	Ensure identity privacy; Easy to understand.	It does not prevent attribute disclosure; Attacks are possible; Re-identification is possible.	Homogeneity attack; Background knowledge attack.

Table 3. *Cont.*

Privacy and Security Mechanisms	Advantages	Limitations	Attack Risks
l-Diversity	Ensure privacy of user attributes.	Does not prevent data against the attacks; In-efficient; Attribute disclosure can be possible.	Skewness attack; Similarity attack.
T-closeness	Protect data privacy	Difficult to identify the closeness between the knowledge gained and t-value by using Earth Mover's Distance(EMD)	Linkability risk is possible to link external data to the equivalence areas
Symmetric Key Cryptography	Re-identification of message is not possible; Authorized user has access to data; Fast; Easy to use.	Difficult to manage multiple keys; Require secure platform to share secret keys.	Re-identification of key from plain text; Recover plain text from cipher text.
Asymmetric Key Cryptography	Ensures data confidentiality.	Slow; Does not provide data integrity.	Possible to crack private keys; Brute force attack.
Hash Function	Flexible; Easy to use.	Not effective on small scale dataset; High computation cost; In-efficient.	Collision attack; Brute force attack.
Homomorphic Encryption	No one can read or modify data; Prevention of active and passive attacks.	Slow.	Heuristic attack.

5.2. Current Recommendations and Research Directions

In this section, we first describe our recommendation for the selection of privacy and security mechanisms on the basis of identified requirements to protect privacy and security-sensitive information. Then, we detail the most prominent privacy and security challenges for blockchain as an outcome of our analysis.

5.2.1. Current Recommendations

Some features are the top priority when designing new data privacy and security mechanisms for blockchain. Based on our previous analysis, these features are unlinkability, low cost, scalability, reliability, and efficiency, low overhead and low computational cost. In the following, we present the most appropriate privacy and security mechanisms and explain their relevance for blockchain.

1. **Ring signature** we consider ring signature to be efficient for identity protection, because of its lower complexity than zero-knowledge proof. However, it is not as scalable as other mechanisms. While scalability can be moderated by creating several rings with a manageable number of participants, key management remains a central problem for ring signature.
2. **Zero-knowledge proof** has higher memory cost and complexity as compared to ring signature. It is mostly considered as a solution to identity management where identity attributes need to be proven without revealing the actual identity. In such a case, it is considered relevant to prevent identity/transaction data disclosure. Despite their higher cost, zero-knowledge-proof constructions are currently the most suitable to address both privacy, and scalability. We conclude that ZKPs can be used to provide anonymous transactions such as Monero [39]. However, they can also be used as a scaling mechanism, where transactions can be batched off-chain, and verified on-chain in blockchain platforms with Turing-complete smart contracts. Presently, the use of ZKPs is limited by the time complexity of constructing proofs, and the lack of protocols that enable trust-less setup [138] with the exception of STARKs [40].
3. **Mix network** is efficient to randomize the order of input data and it is scalable as compared to other available solutions. However, mixed networks require TTPs to shuffle transaction data [1] which is a major drawback in our context.

4. **Identity-based encryption** is better to prevent users' identity disclosure because of its low complexity and computation overhead as compared to attribute-based encryption. However, it is not scalable and has a higher memory overhead than other available solutions.
5. **Attribute-based encryption mechanism** is good to enforce confidentiality on data and fulfills transaction data protection requirements without latency. It works efficiently because it has lower data size overhead and memory cost than identity-based encryption and onion routing mechanisms.
6. **Symmetric key cryptography** is relatively fast for large amounts of data because of its low computation overhead as compared to asymmetric key cryptography. In symmetric key cryptography, keys can be developed and exchanged between two parties through cryptographic algorithms (e.g Diffie-hellman key exchange).
7. **Asymmetric key cryptography** is easy to use due to separate public and private keys, however, it has high computation overhead as compared to symmetric key cryptography.
8. **Homomorphic encryption** is considered less complex and efficient due to its low memory cost as compared to ZKPs, onion routing, and symmetric key cryptography [62]. Advances in cryptography are consistently being integrated to possibly overcome security and privacy issues. Homomorphic encryption is well-positioned to address most privacy concerns. However, the lack of practical, scalable, and audited implementations is hindering integration.
9. **Hash function** is good to fulfill identity and transaction data protection requirements without an intermediate party, because it has lower computation overhead without latency. It generates a unique signature of fixed size from variable-size data input and it is useful to assess the integrity of data during data exchange. It is very widely used for data integrity verification.

5.2.2. Research Directions

As an emerging technology, blockchain is facing some challenging issues. In the following, we summarize privacy and security directions for blockchain environments.

- **Key management**: As it is the case with ring signature, most solutions today rely on asymmetric cryptography. Therefore, a pair of public/private is required for each involved actor. Key management is an issue that needs to be mitigated as a large-scale adoption of developed solutions would need to handle very large numbers of key pairs, including the important constraint that the private key must not be disclosed.
- **Energy consumption of consensus protocols**: Blockchain consensus protocols consume a lot of computing resources, especially the ones based on proof-of-work, which leads to low-system throughput and high system latency. Designing a better consensus mechanism to improve system throughput is challenging due to the algorithmic complexity involved.
- **Security and consensus protocols**: As detailed in Section 4 a good part of security issues related to consensus protocols, such as proof of work, proof of stake, and delegated proof of stake. While consensus mechanisms also have their own challenges in terms of computational cost, they are also central from a security perspective. One possible option in this direction is to constrain the longest chain rule to alleviate majority attacks, which remains an open problem [97].
- **Privacy**: On most public blockchains, transaction data is verified and stored on every node due to its decentralized nature and security concerns. It increases the chances of misuses of the user identities and transaction data. As stated in [34], blockchain does not guarantee transaction and user's identity privacy since all data on the blockchain is publicly available. A lot of work as been ongoing to integrate solutions for privacy protection into the blockchain, a good example of implementation being Monero. Further work is required to protect anonymity at the level of transaction contents, blockchain, history of added blocks, and even blockchain access, so that data read and write accesses becomes totally protected from monitoring attacks. It is noticeable that this problem spreads across the OSI model layers, as identifiers also relate to the network and data link layers (IP and MAC addresses), therefore a well-designed solution should make sure

that all layers are covered. The ability to perform computation on encrypted data will arguably have enormous implications on privacy, security, and scalability. Therefore, future research on homomorphic encryption should be aimed not only at mathematical constructions but also audited implementations. The ability to verify, and process transactions without revealing the data will be the key enabler technology for the future development of DLT.

- **Storage cost**: Due to the decentralized nature of blockchain, data is copied on every node in the network. Therefore, it imposes a huge cost due to the ever-increasing size of the chain. Building storage-wise lightweight blockchain solutions still remains an open challenge. One direction to explore in this domain is the study of multiple subsets with data intersections, where replication would occur on relevant data subsets while maintaining the security properties that make the blockchain interesting. Local network structures could serve as filters in such setup, guaranteeing that majority attacks could not occur within one particular network by preventing external participants to contribute. At the higher level, a decentralized marketplace of micro-currencies would guarantee global communication, maybe using a central blockchain as we know them now to store the history of the different subsets.
- **Blockchain vulnerabilities**: Consensus mechanisms are arguably the biggest security risk. Different consensus mechanisms are available, each with different properties and trade-offs between scalability, security, and decentralization. Secure mechanisms like proof of work have issues with scalability, whereas scalable consensus mechanisms like proof of stake, have issues with security and decentralization [139]. Efficient mechanisms yet require guaranteeing the privacy of transactions and protect against attacks such as double-spending.

6. Conclusions

In this paper, we look at privacy and security concerns for distributed ledger technology. We identify requirements of data privacy and security for blockchain and provide a detailed overview of the limitations of existing technologies. We classify attacks into different categories that provide a comprehensive global overview of the topic, and provide the matching privacy and security countermeasures to mitigate these attacks.

Then, we provide a thorough comparative analysis of state-of-the-art privacy and security-preserving mechanisms before giving recommendations for the different problems blockchain face nowadays. Indeed, while each individual mechanism has advantages and limitations and should be selected according to the constraints of the application domain, some aspects such as scalability, privacy, overhead, memory cost and consensus mechanism remain cross-domain research concerns.

We follow with an overview of the work direction that we identified from our study. We detail open research challenges and propose research directions. We have identified consensus protocols and key management as central elements to improve blockchain developments to come. As well, integration of existing privacy and security protection mechanisms will be a central concern for the development of secure, safer and trusted distributed ledger technology. In this respect, the compromise between complexity and performance will be a key point during this process.

Author Contributions: Conceptualization, S.A., A.T. and M.M.; methodology, S.A., A.T. and M.M.; investigation, S.A., A.T. and M.M.; writing—original draft preparation, S.A. and M.M.; writing—review and editing, S.A., A.T. and M.M.; supervision, M.M.; project administration, M.M.; funding acquisition, M.M. All authors have read and agreed to the published version of the manuscript.

Funding: The authors gratefully acknowledge the European Commission for funding the InnoRenew project (Grant Agreement #739574) under the Horizon2020 Widespread-Teaming program and the Republic of Slovenia (Investment funding of the Republic of Slovenia and the European Regional Development Fund). They also acknowledge the Slovenian Research Agency ARRS for funding the project J2-2504.

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

References

1. Zheng, Z.; Xie, S.; Dai, H.; Chen, X.; Wang, H. An overview of blockchain technology: Architecture, consensus, and future trends. In Proceedings of the 2017 IEEE International Congress on Big Data (BigData Congress), Honolulu, HI, USA, 25–30 June 2017; pp. 557–564.
2. Bonneau, J.; Narayanan, A.; Miller, A.; Clark, J.; Kroll, J.A.; Felten, E.W. Mixcoin: Anonymity for bitcoin with accountable mixes. In Proceedings of the International Conference on Financial Cryptography and Data Security, Christ Church, Barbados, 3–7 March 2014; pp. 486–504.
3. Lu, L.; Han, J.; Hu, L.; Huai, J.; Liu, Y.; Ni, L.M. Pseudo trust: Zero-knowledge based authentication in anonymous peer-to-peer protocols. In Proceedings of the 2007 IEEE International Parallel and Distributed Processing Symposium, Long Beach, CA, USA, 26–30 March 2007; pp. 1–10.
4. Rivest, R.L.; Shamir, A.; Tauman, Y. How to leak a secret. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, 9–13 December 2001; pp. 552–565.
5. Bernabe, J.B.; Canovas, J.L.; Hernandez-Ramos, J.L.; Moreno, R.T.; Skarmeta, A. Privacy-preserving solutions for Blockchain: Review and challenges. *IEEE Access* **2019**, *7*, 164908–164940. [CrossRef]
6. Conti, M.; Kumar, E.S.; Lal, C.; Ruj, S. A survey on security and privacy issues of bitcoin. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 3416–3452. [CrossRef]
7. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2019. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 1 March 2021).
8. Feng, Q.; He, D.; Zeadally, S.; Khan, M.K.; Kumar, N. A survey on privacy protection in blockchain system. *J. Netw. Comput. Appl.* **2019**, *126*, 45–58. [CrossRef]
9. Zhang, R.; Xue, R.; Liu, L. Security and privacy on blockchain. *ACM Comput. Surv. (CSUR)* **2019**, *52*, 1–34. [CrossRef]
10. Halpin, H.; Piekarska, M. Introduction to Security and Privacy on the Blockchain. In Proceedings of the 2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Paris, France, 26–28 April 2017; pp. 1–3.
11. Chen, H.; Pendleton, M.; Njilla, L.; Xu, S. A survey on ethereum systems security: Vulnerabilities, attacks and defenses. *arXiv* **2019**, arXiv:1908.04507.
12. Joshi, A.P.; Han, M.; Wang, Y. A survey on security and privacy issues of blockchain technology. *Math. Found. Comput.* **2018**, *1*, 121–147. [CrossRef]
13. Narayanan, A.; Bonneau, J.; Felten, E.; Miller, A.; Goldfeder, S. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*; Princeton University Press: Princeton, NJ, USA, 2016.
14. Zheng, Z.; Xie, S.; Dai, H.N.; Chen, X.; Wang, H. Blockchain challenges and opportunities: A survey. *Int. J. Web Grid Serv.* **2018**, *14*, 352–375. [CrossRef]
15. Li, W.; Andreina, S.; Bohli, J.M.; Karame, G. Securing proof-of-stake blockchain protocols. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*; Springer: Cham, Switzerland, 2017; pp. 297–315.
16. King, S.; Nadal, S. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. *Self-Publ. Pap.* **2012**, *19*, 1.
17. Sobti, R.; Geetha, G. Cryptographic hash functions: A review. *Int. J. Comput. Sci. Issues* **2012**, *9*, 461.
18. Merkle, R.C. Protocols for public key cryptosystems. In Proceedings of the 1980 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 14–16 April 1980; p. 122.
19. Burgess, K.; Colangelo, J. *The Promise of Bitcoin and the Blockchain*; Consumers' Research: Oxford, UK, 2015.
20. Yang, W.; Garg, S.; Raza, A.; Herbert, D.; Kang, B. Blockchain: Trends and future. In *Pacific Rim Knowledge Acquisition Workshop*; Springer: Cham, Switzerland, 2018; pp. 201–210.
21. Crosby, M.; Pattanayak, P.; Verma, S.; Kalyanaraman, V. Blockchain technology: Beyond bitcoin. *Appl. Innov.* **2016**, *2*, 71.
22. Szabo, N. *The Idea of Smart Contracts*; Satoshi Nakamoto Institute: Nashville, TN, USA, 1997; Volume 6.
23. Brent, L.; Jurisevic, A.; Kong, M.; Liu, E.; Gauthier, F.; Gramoli, V.; Holz, R.; Scholz, B. Vandal: A scalable security analysis framework for smart contracts. *arXiv* **2018**, arXiv:1809.03981.
24. Ekblaw, A.; Azaria, A.; Halamka, J.D.; Lippman, A. A Case Study for Blockchain in Healthcare: "MedRec" prototype for electronic health records and medical research data. In Proceedings of the IEEE Open & Big Data Conference, Washington, DC, USA, 5–8 December 2016; Volume 13, p. 13.
25. Underwood, S. Blockchain beyond bitcoin. *Commun. ACM* **2016**, *59*, 15–17. [CrossRef]

26. Hanke, T.; Movahedi, M.; Williams, D. Dfinity technology overview series, consensus system. *arXiv* **2018**, arXiv:1805.04548.
27. Eisses, J.; Verspeek, L.; Dawe, C.; Dijkstra, S. Effect Network: Decentralized Network for Artificial Intelligence. 2018. Available online: http://effect.ai/download/effectwhitepaper.pdf (accessed on 9 March 2021).
28. Fernández-Caramés, T.M.; Fraga-Lamas, P. A Review on the Use of Blockchain for the Internet of Things. *IEEE Access* **2018**, *6*, 32979–33001. [CrossRef]
29. Xu, L.; Shah, N.; Chen, L.; Diallo, N.; Gao, Z.; Lu, Y.; Shi, W. Enabling the sharing economy: Privacy respecting contract based on public blockchain. In Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts, Abu Dhabi, United Arab Emirates, 2–6 April 2017; pp. 15–21.
30. Kumari, K.A.; Padmashani, R.; Varsha, R.; Upadhayay, V. Securing Internet of Medical Things (IoMT) using private blockchain network. In *Principles of Internet of Things (IoT) Ecosystem: Insight Paradigm*; Springer: Cham, Switzerland, 2020; pp. 305–326.
31. Dinh, T.T.A.; Wang, J.; Chen, G.; Liu, R.; Ooi, B.C.; Tan, K.L. Blockbench: A framework for analyzing private blockchains. In Proceedings of the 2017 ACM International Conference on Management of Data, Chicago, IL, USA, 14–19 May 2017; pp. 1085–1100.
32. Polge, J.; Robert, J.; Le Traon, Y. Permissioned blockchain frameworks in the industry: A comparison. *ICT Express* **2020**. [CrossRef]
33. Henry, R.; Herzberg, A.; Kate, A. Blockchain access privacy: Challenges and directions. *IEEE Secur. Priv.* **2018**, *16*, 38–45. [CrossRef]
34. Kosba, A.; Miller, A.; Shi, E.; Wen, Z.; Papamanthou, C. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2016; pp. 839–858.
35. Barcelo, J. User Privacy in the Public Bitcoin Blockchain. Available online: http://www.dtic.upf.edu/jbarcelo/papers/20140704User.2014PrivacyinthePublicBitcoinBlockchain/paper.pdf (accessed on 9 May 2016).
36. Ammous, S. Blockchain Technology: What Is It Good for? 2016. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2832751 (accessed on 9 March 2021).
37. Chow, S.S.; Yiu, S.M.; Hui, L.C. Efficient identity based ring signature. In *International Conference on Applied Cryptography and Network Security*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 499–512.
38. Tsang, P.P.; Au, M.H.; Liu, J.K.; Susilo, W.; Wong, D.S. A suite of non-pairing id-based threshold ring signature schemes with different levels of anonymity. In *International Conference on Provable Security*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 166–183.
39. Noether, S. Ring SIgnature Confidential Transactions for Monero. *IACR Cryptol. ePrint Arch.* **2015**, *2015*, 1098.
40. Ben-Sasson, E.; Bentov, I.; Horesh, Y.; Riabzev, M. Scalable, transparent, and post-quantum secure computational integrity. *IACR Cryptol. ePrint Arch.* **2018**, *2018*, 46.
41. Jakobsson, M.; Juels, A. An optimally robust hybrid mix network. In Proceedings of the Twentieth Annual ACM Symposium on Principles of Distributed Computing, Newport, RI, USA, 26–29 August 2001; pp. 284–292.
42. Jefferys, K.; Shishmarev, M.; Harman, S. Session: A Model for End-To-End Encrypted Conversations with Minimal Metadata Leakage. *arXiv* **2020**, arXiv:2002.04609.
43. Swan, K. Onion Routing and Tor. *Geo. Law Tech. Rev.* **2016**, *1*, 110–118.
44. Dingledine, R.; Mathewson, N.; Syverson, P. *Tor: The Second-Generation Onion Router*; Technical Report; Naval Research Lab.: Washington, DC, USA, 2004.
45. Sharma, K.; Jayashankar, A.; Banu, K.S.; Tripathy, B. Data Anonymization Through Slicing Based on Graph-Based Vertical Partitioning. In *Proceedings of the 3rd International Conference on Advanced Computing, Networking and Informatics*; Springer: New Delhi, India, 2016; pp. 569–576.
46. Domingo-Ferrer, J.; Torra, V. A critique of k-anonymity and some of its enhancements. In Proceedings of the 2008 Third International Conference on Availability, Reliability and Security, Barcelona, Spain, 4–7 March 2008; pp. 990–993.
47. Aggarwal, G.; Feder, T.; Kenthapadi, K.; Motwani, R.; Panigrahy, R.; Thomas, D.; Zhu, A. Approximation algorithms for k-anonymity. In Proceedings of the International Conference on Database Theory (ICDT 2005), Edinburgh, UK, 5–7 January 2005.

48. Yao, L.; Chen, Z.; Hu, H.; Wu, G.; Wu, B. Sensitive attribute privacy preservation of trajectory data publishing based on l-diversity. *Distrib. Parallel Databases* **2020**, 1–27. [CrossRef]
49. Li, N.; Li, T.; Venkatasubramanian, S. t-closeness: Privacy beyond k-anonymity and l-diversity. In Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering, Istanbul, Turkey, 15–20 April 2007; pp. 106–115.
50. Machanavajjhala, A.; Kifer, D.; Gehrke, J.; Venkitasubramaniam, M. l-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data* **2007**, *1*, 3-es. [CrossRef]
51. Kern, M. Anonymity: A formalization of privacy-l-diversity. In Proceedings of the Zum Seminar Future Internet (FI), Innovative Internet Technologien und Mobilkommunikation (IITM) und Autonomous Communication Networks (ACN), Munich, Germany, 13 April 2013; Volume 49.
52. Wang, M.; Jiang, Z.; Zhang, Y.; Yang, H. T-closeness slicing: A new privacy-preserving approach for transactional data publishing. *INFORMS J. Comput.* **2018**, *30*, 438–453. [CrossRef]
53. Boyen, X.; Mei, Q.; Waters, B. Direct chosen ciphertext security from identity-based techniques. In Proceedings of the 12th ACM Conference on Computer and Communications Security, Alexandria, VA, USA, 7–11 November 2005; pp. 320–329.
54. Waters, B. Efficient identity-based encryption without random oracles. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 114–127.
55. Fotiou, N.; Polyzos, G.C. Decentralized name-based security for content distribution using blockchains. In Proceedings of the 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), San Francisco, CA, USA, 10–14 April 2016; pp. 415–420.
56. Guo, R.; Shi, H.; Zhao, Q.; Zheng, D. Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems. *IEEE Access* **2018**, *6*, 11676–11686. [CrossRef]
57. Rahulamathavan, Y.; Phan, R.C.W.; Rajarajan, M.; Misra, S.; Kondoz, A. Privacy-preserving blockchain based IoT ecosystem using attribute-based encryption. In Proceedings of the 2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Bhubaneswar, India, 17–20 December 2017; pp. 1–6.
58. Panwar, G.; Misra, S.; Vishwanathan, R. Blanc: Blockchain-based anonymous and decentralized credit networks. In Proceedings of the Ninth ACM Conference on Data and Application Security and Privacy, Richardson, TX, USA, 25–27 March 2019; pp. 339–350.
59. Vasantha, R.; Prasad, R.S.; Guntur, A. Secured email data based on blowfish with blockchain technology. *Sci. Technol. Dev.* **2019**, *8*, 456–464.
60. Zhang, Z.; Qi, Q.; Kumar, N.; Chilamkurti, N.; Jeong, H.Y. A secure authentication scheme with anonymity for session initiation protocol using elliptic curve cryptography. *Multimed. Tools Appl.* **2015**, *74*, 3477–3488. [CrossRef]
61. Shah, S. *Use of Blockchain as a Software Component to Send Messages Anonymously for a Data Trading Platform*; Archemy: New Egypt, NJ, USA, 2017.
62. Wang, Q.; Qin, B.; Hu, J.; Xiao, F. Preserving transaction privacy in bitcoin. *Future Gener. Comput. Syst.* **2017**, *107*, 793–804 [CrossRef]
63. Fun, T.S.; Samsudin, A. A survey of homomorphic encryption for outsourced big data computation. *KSII Trans. Internet Inf. Syst.* **2016**, *10*, 3826–3851.
64. AlTawy, R.; ElSheikh, M.; Youssef, A.M.; Gong, G. Lelantos: A blockchain-based anonymous physical delivery system. In Proceedings of the 2017 15th Annual Conference on Privacy, Security and Trust (PST), Calgary, AB, Canada, 28–30 August 2017; pp. 15–1509.
65. Gamage, C.; Gras, B.; Crispo, B.; Tanenbaum, A.S. An identity-based ring signature scheme with enhanced privacy. In Proceedings of the 2006 Securecomm and Workshops, Baltimore, MD, USA, 28 August–1 September 2006; pp. 1–5.
66. Maxwell, G. CoinJoin: Bitcoin Privacy for the Real World. Bitcoin Forum. 2013. Available online: https://bitcointalk.org/index (accessed on 9 March 2021).
67. Kiran, P.; Kavya, N. A survey on methods, attacks and metric for privacy preserving data publishing. *Int. J. Comput. Appl.* **2012**, *53*, 20–28.
68. Sriramoju, S.B. Analysis and Comparison of Anonymous Techniques for Privacy Preserving in Big Data. *Int. J. Adv. Res. Comput. Commun. Eng.* **2017**, *6*, 64–67.

69. Boneh, D.; Franklin, M. Identity-based encryption from the Weil pairing. In *Annual International Cryptology Conference*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 213–229.
70. Shamir, A. Identity-based cryptosystems and signature schemes. In Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques, Paris, France, 9–11 April 1984; pp. 47–53.
71. Nie, T.; Song, C.; Zhi, X. Performance evaluation of DES and Blowfish algorithms. In Proceedings of the 2010 International Conference on Biomedical Engineering and Computer Science, Wuhan, China, 23–25 April 2010; pp. 1–4.
72. Jeeva, A.; Palanisamy, D.V.; Kanagaram, K. Comparative analysis of performance efficiency and security measures of some encryption algorithms. *Int. J. Eng. Res. Appl.* **2012**, *2*, 3033–3037.
73. Yakoubov, S.; Gadepally, V.; Schear, N.; Shen, E.; Yerukhimovich, A. A survey of cryptographic approaches to securing big-data analytics in the cloud. In Proceedings of the 2014 IEEE High Performance Extreme Computing Conference (HPEC), Waltham, MA, USA, 9–11 September 2014; pp. 1–6.
74. Marin, G.A. Network security basics. *IEEE Secur. Priv.* **2005**, *3*, 68–72. [CrossRef]
75. Jawandhiya, P.M.; Ghonge, D.; Ali, M.; Deshpande, J. A survey of mobile ad hoc network attacks. *Int. J. Eng. Sci. Technol.* **2010**, *2*, 4063–4071. [CrossRef]
76. Ehrenkranz, T.; Li, J. On the state of IP spoofing defense. *ACM Trans. Internet Technol.* **2009**, *9*, 1–29. [CrossRef]
77. Tsudik, G. Message Authentication with One-Way Hash Functions. *SIGCOMM Comput. Commun. Rev.* **1992**, *22*, 29–38. [CrossRef]
78. Chandra, S.; Paira, S.; Alam, S.S.; Sanyal, G. A comparative survey of symmetric and asymmetric key cryptography. In Proceedings of the 2014 International Conference on Electronics, Communication and Computational Engineering (ICECCE), Hosur, India, 17–18 November 2014; pp. 83–93.
79. Yarımtepe, O.; Dalkılıç, G.; Özcanhan, M.H. Distributed Denial of Service Prevention Techniques. In Proceedings of the 3rd International Symposium on Digital Forensics and Security, Ankara, Turkey, 11–12 May 2015; pp. 24–28.
80. Ekparinya, P.; Gramoli, V.; Jourjon, G. Impact of man-in-the-middle attacks on ethereum. In Proceedings of the 2018 IEEE 37th Symposium on Reliable Distributed Systems (SRDS), Salvador, Brazil, 2–5 October 2018; pp. 11–20.
81. Aliyu, F.; Sheltami, T.; Shakshuki, E.M. A detection and prevention technique for man in the middle attack in fog computing. *Procedia Comput. Sci.* **2018**, *141*, 24–31. [CrossRef]
82. Lou, X.; Huu, C.T.; Tan, R.; Yau, D.K.; Kalbarczyk, Z.T. Assessing and Mitigating Impact of Time Delay Attack against Cyber-Physical Systems. Available online: http://publish.illinois.edu/cps-security/files/2018/05/delay.pdf (accessed on 14 September 2020).
83. Sargolzaei, A.; Yen, K.K.; Abdelghani, M.N.; Mehbodniya, A.; Sargolzaei, S. A novel technique for detection of time delay switch attack on load frequency control. *Intell. Control. Autom.* **2015**, *6*, 205. [CrossRef]
84. Apostolaki, M.; Zohar, A.; Vanbever, L. Hijacking bitcoin: Routing attacks on cryptocurrencies. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; pp. 375–392.
85. Xing, Q.; Wang, B.; Wang, X. Bgpcoin: Blockchain-based internet number resource authority and bgp security solution. *Symmetry* **2018**, *10*, 408. [CrossRef]
86. Kiayias, A.; Panagiotakos, G. On trees, chains and fast transactions in the blockchain. In *International Conference on Cryptology and Information Security in Latin America*; Springer: Cham, Switzerland, 2017; pp. 327–351.
87. Li, C.; Li, P.; Zhou, D.; Yang, Z.; Wu, M.; Yang, G.; Xu, W.; Long, F.; Yao, A.C.C. A Decentralized Blockchain with High Throughput and Fast Confirmation. In Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC 20), Boston, MA, USA, 15–17 July 2020; pp. 515–528.
88. Serjantov, A.; Sewell, P. Passive attack analysis for connection-based anonymity systems. In *European Symposium on Research in Computer Security*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 116–131.
89. Hiller, J.; Pennekamp, J.; Dahlmanns, M.; Henze, M.; Panchenko, A.; Wehrle, K. Tailoring onion routing to the Internet of Things: Security and privacy in untrusted environments. In Proceedings of the 2019 IEEE 27th International Conference on Network Protocols (ICNP), Chicago, IL, USA, 8–10 October 2019; pp. 1–12.
90. Pawar, M.V.; Anuradha, J. Network security and types of attacks in network. *Procedia Comput. Sci.* **2015**, *48*, 503–506. [CrossRef]

91. Sinha, P.; Jha, V.; Rai, A.K.; Bhushan, B. Security vulnerabilities, attacks and countermeasures in wireless sensor networks at various layers of OSI reference model: A survey. In Proceedings of the 2017 International Conference on Signal Processing and Communication (ICSPC), Coimbatore, India, 28–29 July 2017; pp. 288–293.
92. Decker, C.; Wattenhofer, R. Bitcoin transaction malleability and MtGox. In *European Symposium on Research in Computer Security*; Springer: Cham, Switzerland, 2014; pp. 313–326.
93. Rajput, U.; Abbas, F.; Oh, H. A solution towards eliminating transaction malleability in bitcoin. *J. Inf. Process. Syst.* **2018**, *14*, 837–850.
94. Buterin, V.; Griffith, V. Casper the friendly finality gadget. *arXiv* **2017**, arXiv:1710.09437.
95. Sompolinsky, Y.; Zohar, A. Secure high-rate transaction processing in bitcoin. In *Conference on Financial Cryptography and Data Security*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 507–527.
96. Buterin, V.; Hernandez, D.; Kamphefner, T.; Pham, K.; Qiao, Z.; Ryan, D.; Sin, J.; Wang, Y.; Zhang, Y.X. Combining GHOST and Casper. *arXiv* **2020**, arXiv:2003.03052.
97. Sayeed, S.; Marco-Gisbert, H. Assessing blockchain consensus and security mechanisms against the 51% attack. *Appl. Sci.* **2019**, *9*, 1788. [CrossRef]
98. Eyal, I.; Sirer, E.G. Majority is not enough: Bitcoin mining is vulnerable. In *Conference on Financial Cryptography and Data Security*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 436–454.
99. Solat, S.; Potop-Butucaru, M. Brief announcement: Zeroblock: Timestamp-free prevention of block-withholding attack in bitcoin. In *International Symposium on Stabilization, Safety, and Security of Distributed Systems*; Springer: Cham, Switerland, 2017; pp. 356–360.
100. Heilman, E.; Kendler, A.; Zohar, A.; Goldberg, S. Eclipse attacks on bitcoin's peer-to-peer network. In Proceedings of the 24th {USENIX} Security Symposium ({USENIX} Security 15), Washington, DC, USA, 12–14 August 2015; pp. 129–144.
101. Simmons, S. A Brief Breakdown of Monero's Ongoing Network Attacks. 2021. Available online: https://sethsimmons.me/posts/moneros-ongoing-network-attack/ (accessed on 26 February 2021).
102. Singh, A.; Castro, M.; Druschel, P.; Rowstron, A. Defending against eclipse attacks on overlay networks. In Proceedings of the 11th Workshop on ACM SIGOPS European Workshop, Leuven, Belgium, 19–22 September 2004; p. 21-es.
103. Bissias, G.; Ozisik, A.P.; Levine, B.N.; Liberatore, M. Sybil-resistant mixing for bitcoin. In Proceedings of the 13th Workshop on Privacy in the Electronic Society, Scottsdale, AZ, USA, 3 November 2014; pp. 149–158.
104. Lee, H.; Shin, M.; Kim, K.S.; Kang, Y.; Kim, J. Recipient-oriented transaction for preventing double spending attacks in private blockchain. In Proceedings of the 2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Hong Kong, China, 11–13 June 2018; pp. 1–2.
105. Li, X.; Jiang, P.; Chen, T.; Luo, X.; Wen, Q. A survey on the security of blockchain systems. *Future Gener. Comput. Syst.* **2020**, *107*, 841–853. [CrossRef]
106. Juels, A.; Kosba, A.; Shi, E. The ring of gyges: Investigating the future of criminal smart contracts. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 283–295.
107. Feist, J.; Grieco, G.; Groce, A. Slither: A static analysis framework for smart contracts. In Proceedings of the 2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB), Montreal, QC, Canada, 27 May 2019; pp. 8–15.
108. Tikhomirov, S.; Voskresenskaya, E.; Ivanitskiy, I.; Takhaviev, R.; Marchenko, E.; Alexandrov, Y. Smartcheck: Static analysis of ethereum smart contracts. In Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain, Gothenburg, Sweden, 27 May–3 June 2018; pp. 9–16.
109. Atzei, N.; Bartoletti, M.; Cimoli, T. A survey of attacks on ethereum smart contracts (sok). In *International Conference on Principles of Security and Trust*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 164–186.
110. Sayeed, S.; Marco-Gisbert, H.; Caira, T. Smart contract: Attacks and protections. *IEEE Access* **2020**, *8*, 24416–24427. [CrossRef]
111. Mayer, H. ECDSA security in bitcoin and ethereum: A research survey. *CoinFaabrik* **2016**, *28*, 126.
112. Dai, W.; Deng, J.; Wang, Q.; Cui, C.; Zou, D.; Jin, H. SBLWT: A secure blockchain lightweight wallet based on trustzone. *IEEE Access* **2018**, *6*, 40638–40648. [CrossRef]
113. Laurent, M.; Giannetsos, T. (Eds.) *Information Security Theory and Practice*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; Volume 12024. [CrossRef]

114. Xiao, Z.; Xu, J.; Meng, X. p-sensitivity: A semantic privacy-protection model for location-based services. In Proceedings of the 2008 Ninth International Conference on Mobile Data Management Workshops (MDMW), Beijing, China, 27–30 April 2008; pp. 47–54.
115. Sowmyarani, C.; Srinivasan, G.; Sukanya, K. A New Privacy Preserving Measure: p-Sensitive, t-Closeness. In *International Conference on Advances in Computing*; Springer: New Delhi, India, 2013; pp. 57–62.
116. Kumar, A.; Fischer, C.; Tople, S.; Saxena, P. A traceability analysis of monero's blockchain. In Proceedings of the European Symposium on Research in Computer Security, Oslo, Norway, 11–15 September 2017; pp. 153–173.
117. Möser, M.; Soska, K.; Heilman, E.; Lee, K.; Heffan, H.; Srivastava, S.; Hogan, K.; Hennessey, J.; Miller, A.; Narayanan, A.; et al. An empirical analysis of traceability in the monero blockchain. *Proc. Priv. Enhanc. Technol.* **2018**, *2018*, 143–163. [CrossRef]
118. Prabhakaran, M.; Sahai, A. Concurrent Zero Knowledge Proofs with Logarithmic Roundcomplexity. 2004. Available online: http://eprint.iacr.org/2002/055.pdf (accessed on 9 March 2021).
119. Lu, L.; Han, J.; Liu, Y.; Hu, L.; Huai, J.P.; Ni, L.; Ma, J. Pseudo trust: Zero-knowledge authentication in anonymous P2Ps. *IEEE Trans. Parallel Distrib. Syst.* **2008**, *19*, 1325–1337. [CrossRef]
120. Byun, J.W.; Kamra, A.; Bertino, E.; Li, N. Efficient k-anonymization using clustering techniques. In *International Conference on Database Systems for Advanced Applications*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 188–200.
121. Meyerson, A.; Williams, R. On the complexity of optimal k-anonymity. In Proceedings of the Twenty-Third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, Paris, France, 14–16 June 2004; pp. 223–228.
122. Ghinita, G.; Karras, P.; Kalnis, P.; Mamoulis, N. Fast data anonymization with low information loss. In Proceedings of the 33rd International Conference on Very Large Data Bases, Vienna, Austria, 23–27 September 2007; pp. 758–769.
123. Mehta, B.B.; Rao, U.P. Improved l-Diversity: Scalable Anonymization Approach for Privacy Preserving Big Data Publishing. *J. King Saud Univ. Comput. Inf. Sci.* **2019**. [CrossRef]
124. Chase, M.; Chow, S.S. Improving privacy and security in multi-authority attribute-based encryption. In Proceedings of the 16th ACM conference on Computer and Communications Security, Chicago, IL, USA, 9–13 November 2009; pp. 121–130.
125. Edemacu, K.; Park, H.K.; Jang, B.; Kim, J.W. Privacy Provision in Collaborative Ehealth With Attribute-Based Encryption: Survey, Challenges and Future Directions. *IEEE Access* **2019**, *7*, 89614–89636. [CrossRef]
126. Farah, S.; Javed, Y.; Shamim, A.; Nawaz, T. An experimental study on performance evaluation of asymmetric encryption algorithms. In Proceedings of the 3rd European Conference of Computer Science on Recent Advances in Information Science (EECS-12), Paris, France, 2–4 December 2012; pp. 121–124.
127. Levi, A.; Savas, E. Performance evaluation of public-key cryptosystem operations in WTLS protocol. In Proceedings of the Eighth IEEE Symposium on Computers and Communications (ISCC 2003), Kemer-Antalya, Turkey, 3 July 2003; pp. 1245–1250.
128. Maqsood, F.; Ahmed, M.; Ali, M.M.; Shah, M.A. Cryptography: A comparative analysis for modern techniques. *Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*, 442–448. [CrossRef]
129. Awasthi, A.K.; Lal, S. ID-based ring signature and proxy ring signature schemes from bilinear pairings. *arXiv* **2005**, arXiv:cs/0504097.
130. Boneh, D.; Gentry, C.; Lynn, B.; Shacham, H. Aggregate and verifiably encrypted signatures from bilinear maps. In *International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 416–432.
131. Jurczyk, P.; Xiong, L. Distributed anonymization: Achieving privacy for both data subjects and data providers. In *IFIP Annual Conference on Data and Applications Security and Privacy*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 191–207.
132. Kawle, P.; Hiwase, A.; Bagde, G.; Tekam, E.; Kalbande, R. Modified advanced encryption standard. *Int. J. Soft Comput. Eng.* **2014**, *4*, 21–23.
133. Padmavathi, B.; Kumari, S.R. A survey on performance analysis of DES, AES and RSA algorithm along with LSB substitution. *Int. J. Sci. Res.* **2013**, 170–174.
134. Mansour, A.; Malik, K.M.; Kaso, N. AMOUN: Asymmetric lightweight cryptographic scheme for wireless group communication. *arXiv* **2020**, arXiv:2002.06713.

135. Amin, M.; Faragallah, O.S.; Abd El-Latif, A.A. Chaos-based hash function (CBHF) for cryptographic applications. *Chaos Solitons Fractals* **2009**, *42*, 767–772. [CrossRef]
136. Nakajima, J.; Matsui, M. Performance analysis and parallel implementation of dedicated hash functions. In *Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 165–180.
137. Gahi, Y.; Guennoun, M.; El-Khatib, K. A secure database system using homomorphic encryption schemes. *arXiv* **2015**, arXiv:1512.03498.
138. Gennaro, R.; Micciancio, D.; Rabin, T. An efficient non-interactive statistical zero-knowledge proof system for quasi-safe prime products. In Proceedings of the 5th ACM Conference on Computer and Communication Security (CCS'98), San Francisco, CA, USA, 3–5 November 1998; ACM Press: San Francisco, CA, USA, 1998; pp. 67–72.
139. Brown-Cohen, J.; Narayanan, A.; Psomas, A.; Weinberg, S.M. Formal barriers to longest-chain proof-of-stake protocols. In Proceedings of the 2019 ACM Conference on Economics and Computation, Phoenix, AZ, USA, 24–28 June 2019; pp. 459–473.

Short Biography of Authors

Sidra Aslam received her Master degree in Computer Science from the COMSATS University Islamabad, Pakistan in January 2017. During this period, she worked on research projects and published research papers in scholarly journals and conference proceedings. In 2015, she received the best research paper award at the National Software Engineering Conference (NSEC), IEEE in Pakistan. She is currently an Assistant Researcher at the InnoRenew CoE, a teaching assistant and a Ph.D. student at the Faculty of Mathematics, Natural Sciences and Information Technologies, University of Primorska, Slovenia. Her current research interests include information security and privacy, blockchain, and semantic web technologies.

Aleksandar Tošić received his Masters in Computer Science from the University of Primorska in 2016. He is currently a teaching assistant and Phd student at University of Primorska, and a research assistant at InnoRenew CoE. His main areas of research are related to distributed, and decentralised network protocols, and distributed ledger technology.

Dr. Michael Mrissa received his PhD in computer science from the University of Lyon, France, in 2007. His main areas of research are related to distributed systems and include service-oriented computing, semantic web, privacy, web of things. He has authored 80+ peer-reviewed publications in international conferences and journals, and he has been involved in several European, French and Slovenian research projects. He is currently researcher in the ICT research group of the InnoRenew CoE. He also holds a full professor position at the Faculty of Mathematics, Natural sciences and Information Technologies of the University of Primorska.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.