computers

MDPI

Article

Privacy-Preserving Passive DNS

Pavlos Papadopoulos [1,*], Nikolaos Pitropakis [1,*], William J. Buchanan [1] and Owen Lo [1] and Sokratis Katsikas [2,3]

1 School of Computing Edinburgh Napier University, Edinburgh EH10 5DT, UK; B.Buchanan@napier.ac.uk (W.J.B.); O.Lo@napier.ac.uk (O.L.)
2 Department of Information Security and Communication Technology, Norwegian University of Science and Technology, 2815 Gjøvik, Norway; sokratis.katsikas@ntnu.no
3 Faculty of Pure and Applied Sciences, Open University of Cyprus, 2220 Latsia, Cyprus
* Correspondence: pavlos.papadopoulos@napier.ac.uk (P.P.); N.Pitropakis@napier.ac.uk (N.P.)

Received: 14 July 2020; Accepted: 9 August 2020; Published: 12 August 2020

Abstract: The Domain Name System (DNS) was created to resolve the IP addresses of web servers to easily remembered names. When it was initially created, security was not a major concern; nowadays, this lack of inherent security and trust has exposed the global DNS infrastructure to malicious actors. The passive DNS data collection process creates a database containing various DNS data elements, some of which are personal and need to be protected to preserve the privacy of the end users. To this end, we propose the use of distributed ledger technology. We use Hyperledger Fabric to create a permissioned blockchain, which only authorized entities can access. The proposed solution supports queries for storing and retrieving data from the blockchain ledger, allowing the use of the passive DNS database for further analysis, e.g., for the identification of malicious domain names. Additionally, it effectively protects the DNS personal data from unauthorized entities, including the administrators that can act as potential malicious insiders, and allows only the data owners to perform queries over these data. We evaluated our proposed solution by creating a proof-of-concept experimental setup that passively collects DNS data from a network and then uses the distributed ledger technology to store the data in an immutable ledger, thus providing a full historical overview of all the records.

Keywords: passive DNS (Domain Name System); privacy-preserving; distributed ledger; blockchain; hyperledger fabric; private data collection

1. Introduction

The Domain Name System (DNS) translates human-readable domain names to machine-readable IP addresses [1]. An end-user can thus access a website through a web browser using a combination of a name, e.g., "example" and a TLD, e.g., ".com", ".uk", ".us". Figure 1 illustrates the steps involved in the DNS query resolution process. DNS provides a foundation element of the trustworthiness of the Internet, but its simplicity and general lack of trust has led to a range of security issues. Botnets [2], parking domains [3] and domain squatting [4] are examples of types of malicious DNS use. To identify DNS abuse, the DNS queries and responses often have to be collected for further analysis.

Florian Weimer is the creator of the Passive DNS systems [5]; he used recursive name servers to log responses received from different name servers and then copied this logged data onto a central database. The almost instantaneous recording of the majority of passive DNS data before the recursive name server means that Passive DNS is composed of responses and referrals from online authoritative name servers. These logged data are deduped, compressed, time-stamped, and then copied onto a central database where they are analysed and archived.

Passive DNS collects the DNS queries along with the IP address of the host that is making the queries. In situations when the passive DNS collector is placed within the ISP (Internet Service Provider)

or at a TLD (Top-Level Domain) server, each query contains the IPs of the end-users and can be linked back to them. Both the GDPR and NIST consider IP addresses as personal data when a correlation of the queries and the identity of the end-user can be made [6,7]. When it comes to public DNS servers, end-users can benefit from better (than the DNS servers of their ISP) stability, availability and protection against certain DNS attacks, but they expose their personal data to companies such as Google [8], Cloudflare [9], and OpenDNS [10] that could profit from commercially exploiting these data [11]. It follows that there is a clear need to appropriately protect data collected, stored, and processed to identify malicious domain names through the use of passive DNS data analysis methods. The majority of existing solutions for passive DNS data analysis provide APIs for queries of the related data. However, the collection of passive DNS data is being questioned [12], since the privacy of the end-users that contributed their passive DNS collections may be compromised [13].

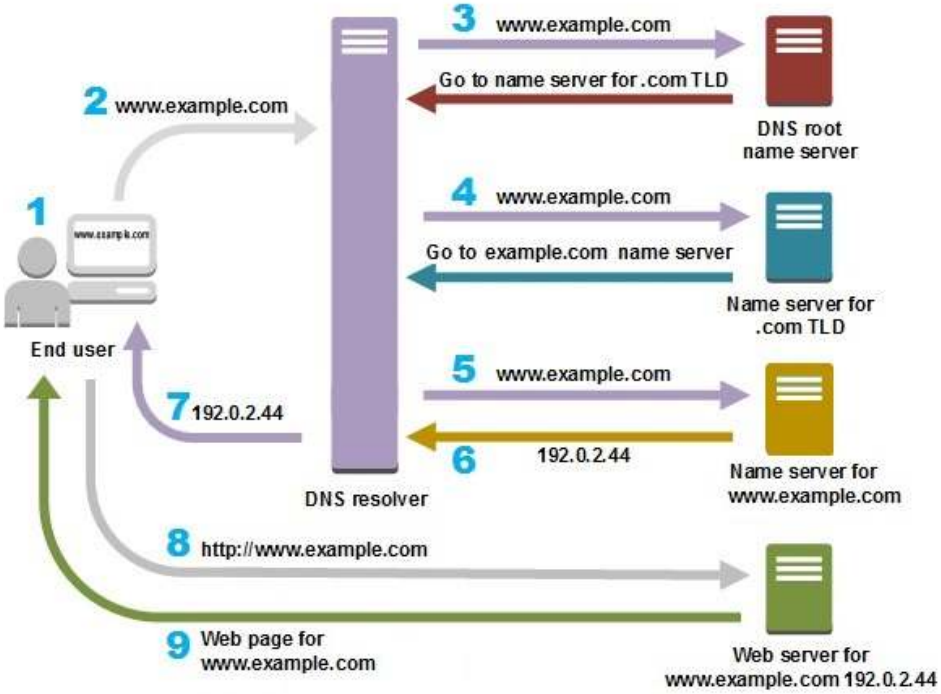


Figure 1. Overview of the Domain Name System (DNS).

This paper proposes a solution to the aforementioned problem and outlines Privacy Preserving Passive (PRESERVE DNS), a system that collects passive DNS data for further analysis, whilst preserving the privacy of the end-users, by virtue of storing the data in an immutable distributed ledger. This immutable ledger can be used to identify abuses and malicious DNS usage, such as for domain squatting and botnets. Blockchain technology can also provide the complete history of the data transactions, without the need for a central authority to control it. The proposed solution is not connected with any passive DNS database provider, such as Farsight (DNSDB) [14], and VirusTotal [15]. Furthermore, it provides transparency of the stored data, thus establishing trust with the users, while at the same time allowing authorized entities to query only non-personal data.

Not all data collected are personal or in need of the same level of privacy protection. Selectively protecting personal data, such as source IP addresses, is possible in a permissioned distributed ledger. In a permissioned-protected field of an immutable ledger, even the administrators do not have access to the private data. This choice allows only the data owners to query the data. In this paper, Hyperledger Fabric is the selected permissioned blockchain platform. This platform has the advantage of resolving potential scalability issues since consensus of participating peers and their respective permissions can be configured. Additionally, with the encryption and anonymisation

functionality provided within Hyperledger Fabric, participating entities can securely provide passive DNS collections to the system.

The contributions of our work can be summarised as follows:

- We have developed PRESERVE DNS, a privacy-preserving passive DNS data solution, by leveraging distributed ledger technology. The implementation of this solution does not require any changes to the server side of the current DNS infrastructure.
- We have evaluated the robustness and security of PRESERVE DNS.
- We have comparatively evaluated the performance of PRESERVE DNS against a traditional database with column level encryption, and against an existing alternative solution.

The remaining of the paper is organised as follows: Section 2 provides the background on blockchain technology that is necessary to ensure the self-sustainability of the paper. The relevant literature is discussed in Section 3. In Section 4 the proof-of-concept implementation of PRESERVE DNS, which also serves as the evaluation testbed, are discussed. Section 5 presents the process and the results of the evaluation of PRESERVE DNS. Finally, Section 6 summarizes our conclusions and suggests some future work.

2. Background

2.1. DNS Privacy Concerns

The DNS infrastructure is outdated and has been created without considering security or privacy. This leads to DNS being targeted by numerous malicious actors that try to exploit its vulnerabilities to get profit from unaware end-users. Malevolent individuals are able to profit directly from their victims, or by selling batches of their private information [16–18]. A number of current privacy issues cannot be fully resolved without a complete redesign of DNS [19]. Fortunately, new solutions are being proposed and developed to further enhance the security of DNS and to preserve the privacy of its end-users. Blockchain DNS solutions promise to resolve existing DNS privacy issues [20–22]. Our work combines part of the old DNS infrastructure with the new blockchain technology.

2.2. Blockchain

Blockchain is a distributed ledger technology [23] that became popular as the foundational block of the bitcoin cryptocurrency [24,25]. Over the past few years it has seen rapid growth, both in terms of research and commercial usage. A blockchain is a cryptographically linked list of records that maintains a publicly verifiable ledger without the need for a central authority; as such, it is a new paradigm of trust between entities in various application domains. The technology behind blockchains originated in cryptocurrency applications but its advancements over existing architectures motivated researchers to apply it to a broad spectrum of application domains [26–30]. The main benefits of this technology are the decentralized nature, immutability, inherent anonymity, resilience, trust, security, autonomy, integrity and scalability.

Blockchains can be categorized according to who can access them as public, private or consortium. Anyone can join a public blockchain and act as a simple node or as miner/validator; no approval by any third party is needed. In the case of a private or consortium blockchains, the owner restricts network access, to herself or to a group of participants. Blockchains can also be categorized according to whether each entity requires authorization to perform an action or not as permissioned or permissionless respectively [31].

2.3. Hyperledger Fabric

Hyperledger Fabric is an open source enterprise-grade permissioned distributed ledger technology platform [32]. The main participants in a Hyperledger Fabric blockchain are the peers, the organizations,

the orderers, the Certificate Authority, and the Membership Service Provider (MSP). Participants perform actions on the ledger by using chaincode [30].

The chaincode is a blockchain program that runs autonomously and performs a set of actions defined by the developer. It is written in common general-purpose programming languages such as Javascript, Java or Go. The chaincode is installed and instantiated in the peers and the orderer, and contains all the blockchain's logic, security mechanisms and capabilities.

Peers are the most crucial entities on the blockchain network. They install the chaincode, and they host the ledger. Peers can host more than one ledgers and chaincodes, enabling private communications with other entities.

Organizations are groups of peers whose actions can be defined by policy [33]. The network is formed by different organizations, that all together form a consortium [34]. It is possible that different ledgers can be present at the same time and only authorized organisations and entities have access to them.

Ordering Service or orderer is the entity that receives the transactions from the peers' applications and updates the ledger according to the defined consensus mechanism. The consensus mechanism can be defined during the time of creation, and complex fault-tolerant algorithms can be used for the validation of each transaction [35].

The Certificate Authority is the entity that assigns an identity to each peer. The Membership Service Provider (MSP) is the entity that validates the identity of each participant in the blockchain network. It manages and examines all the cryptographic mechanisms and certificates that the peers use to perform actions in the ledger [32].

3. Related Work

Despite the success of Weimer's concept, the issue of the impact of the collection of passive DNS data to end-user privacy was soon raised, as users could be clueless if a passive DNS collector is placed in their DNS resolver [7]. Consequently, the security research community focused on this issue and several approaches addressing the issue were published.

One of the first approaches to mitigate the impact of the issue was to use tools that could eliminate confidential information from collected network packets [36]. Another approach argued that a Cryptography-based Prefix preserving Anonymization algorithm [37] or other encryption techniques that would secure the IP prefix [38] should be employed. At the other end of the spectrum, an entirely different solution was proposed: the collection of active DNS data [39]. This was made possible by creating a system called Thales which can systematically query and collect large volumes of active DNS data using as input an aggregation of publicly accessible sources of domain names and URLs that have been collected for several years by the research team. These sources include but are not limited to Public Blacklists, the Alexa ranking, the Common Crawl project, and various Top Level Domain (TLD) zone files. This system's output is a refined dataset that can be easily used by the security community.

Liang et al. [40] proposed a system that combines two technologies, namely blockchain and cloud computing to effectively and efficiently create a decentralised DNS records database. To ensure the security of the stored data, they employed a hashed version of the sensitive data as a proof-of-identity, and allowed only the administrator of the system to correlate each identity to the hashed data. This is a successful countermeasure against outsider malicious actors, but sensitive data may still be compromised when attacks are launched by insiders. In contrast, the data in PRESERVE DNS are stored by the users themselves and are available in a separate ledger only to them; the remaining networks only have access to a hash of the actual data.

Liu et al. [22] proposed a decentralized, blockchain-based DNS (DecDNS) system which maintains a stored database of DNS records and performs the resolution using the nodes of the blockchain. The advantages and default security mechanisms of blockchain, such as the tamper-proof state of the data and the resilience against Distributed Denial of Service (DDoS) attacks are important features of the system. Moreover, the solution does not require significant changes to the existing DNS

infrastructure. However, the privacy of the users is not preserved; the sacalability of the solution is questionable; and performance is a challenge, as all the data are in a hashed form, and they should be decrypted in every DNS query. In contrast, in PRESERVE DNS public data needed for further analysis are in plain text, and only authorized entities are allowed to query them.

Rather than attempting to secure the existing DNS infrastructure, another line of research proposes the development of a more secure, easily audited, transparent domain names organization. Examples are systems such as Namecoin [41] and Blockstack [42], that created a substitute of the Internet Corporation for Assigned Names and Numbers (ICANN), where each user does not need to buy a domain name from a third party. The proposed system is built on a blockchain network, bitcoin in this case, where users can "mine" cryptocurrency. Then users are able to use this cryptocurrency to buy domain names with new ".bit", ".id" TLDs that did not exist before [42]. The privacy of the users can be ensured since their identity is protected by bitcoin's identity management mechanism. The downside of these systems is that the users need specific extensions to be able to query blockchain registered domain names.

Along similar lines, Kalodner et al. [41] and Ali et al. [42] proposed solutions that can address common DNS issues and attacks, by changing the existing infrastructure to a more secure, resilient version with far more opportunities, security mechanisms and defences. However attractive these proposals may seem, the requirement to make changes to the existing DNS global infrastructure is rather unrealistic. In contrast, PRESERVE DNS is able to secure the existing DNS functionality without the need for major changes at the server, thus offering the opportunity for a faster transition without downtime or enormous expenses.

DNS Trusted Sharing Model (DNSTSM) is a recently published approach, which also uses the Hyperledger Fabric platform [43]. DNSTSM is a system resilient to various DNS attacks, that achieves a high performance of DNS resolutions, and does not require changes in the current global DNS infrastructure. However, DNSTSM is built on the older v1.1 version of Hyperledger Fabric, which does not have the private data collection feature as the v1.4 version that PRESERVE DNS is built upon does. This means that DNSTSM cannot preserve end-user privacy without effectively re-designing its architecture so as to be able to exploit features available in newer versions of Hyperledger Fabric.

PRESERVE DNS differentiates itself from previous works in various ways. By leveraging the private data collection feature provided by Hyperledger Fabric, in PRESERVE DNS two separate ledgers are created, one for the public DNS information used for further analysis, and one for the sensitive data. The latter is stored only on the peer nodes of the owners of the data, and only they can query it. Furthermore, PRESERVE DNS stores the passive DNS data collection through storing API requests directly from the users, making use of each user's identity. This method enhances the privacy of the users as only themselves have access to their personal data and are able to query them. Additionally, PRESERVE DNS is efficient, since the rest of the data are in plain text, and only trusted validated users are allowed to query them. Data are available through query API requests to authorized entities, and further analysis of passive DNS data towards, e.g., malicious domain name identification and domain squatting is possible.

Finally, as will be discussed in Section 5, PRESERVE DNS is able to thwart various DNS attacks such as DDoS, DNS fast-flux, DNS amplification attacks. In situations where the PRESERVE DNS distributed DNS records database is being used for the DNS resolution, the DNS cache poisoning attack, one of the most difficult attacks to defend against, can be thwarted as well.

A summary of salient characteristics of PRESERVE DNS and of the approaches discussed above is depicted in Table 1.

Table 1. Comparison of methods.

Method	Attack Thwarting	User Privacy	Existing DNS Infrastructure
DecDNS Liu et al. [44]	✓	X	✓
Liang et al. [40]	✓	X	✓
Namecoin Kalodner et al. [41]	✓	✓	X
Blockstack Ali et al. [42]	✓	✓	X
DNSTSM Yu et al. [43]	✓	X	✓
PRESERVE DNS	✓	✓	✓

4. Proof-of-Concept Implementation

4.1. Architecture

In order to demonstrate the workings of PRESERVE DNS, and to evaluate its operation and performance, we developed a proof-of-concept implementation, whose architecture is depicted in Figure 2. This proof-of-concept implementation comprises a private data collection that contains the Passive DNS data and is controlled by two authorized Organizations with regards to reproducing a Passive DNS infrastructure. Each Organization contains two Peers that own the blockchain ledger. Each Passive DNS data record consists of ten fields, namely the associated blockchain and record IDs; the domain name; its IP address; the Time-To-Live (TTL); two timestamp fields (in seconds and milliseconds); the number of times that the user visited the domain; the IP address of the end-user; and the server that performed the resolution. The last fields contain personal data and form the private data collection, which is accessible only by the Peers of Organization 1; others may access only the non-personal data. Note that read and write query times of the Peers are measured as part of the evaluation of the proposed infrastructure, reported in the next section.

As shown in Figure 2, the implementation involves a network of a number of computers that run various operating systems (Microsoft Windows 10, Apple MacOS, Kali Linux and Ubuntu Linux) and use a distributed infrastructure as a local DNS resolver. Specifically, a Kubernetes pod [45] has been configured locally as the DNS resolver of the network. The Kubernetes pod acts as a host machine, using its own IP address. The remaining machines use the DNS resolver as their DNS server, and the local DNS resolver can resolve the DNS queries itself, it can use the ISP's DNS servers, or one of the public DNS servers provided by companies such as Google [8], Cloudflare [9] or OpenDNS [10]. In our case, the specified DNS resolver uses Google's public DNS servers for the resolution of the DNS queries, eliminating the chance of a "bad" ISP DNS resolver for each DNS query [46]. A passive DNS data collector is needed to collect the DNS queries and responses to store the data in a database for further analysis. This configuration is able to capture queries such as A, AAAA, MX records and the translations between their servers with the domain names. The stored data involves the IPs of the machines that performed the DNS queries and the server that performed the DNS resolution. The DNS resolver passively collects and stores this information in a database, using passivedns from gamelinux [47], and stores the data in JSON format. The system built for the resolution of the DNS queries uses Berkeley Internet Name Domain (BIND) version 9. BIND is the most common implementation of a DNS resolver, and it depends solely on the nameserver it queries [44]. The technical specifications of the computers that performed the queries meet the minimum docker container needs. The technical specifications of the distributed infrastructure that hosts the blockchain system are as follows: 6th Generation 2.0 GHZ dual-core Intel Core i5 CPU, with 8 GB RAM running at 1866 MHz and 256 GB PCIe-based flash storage.

This proof-of-concept configuration gives rise to privacy concerns, since IPs can be correlated to the identities of the end-users that visited the sites. However, we emphasize that this configuration is being used only for creating the Passive DNS database; as such its privacy is outside the scope of this paper. The privacy of Passive DNS data collector systems is ensured by PRESERVE DNS that employs a blockchain solution to store the DNS data in an immutable ledger.

The blockchain solution should adhere to the following specifications:

- Participants of the network should be able to easily query the data stored in the blockchain.
- The queried data should be available only to authorized entities in order to be analysed further for its maliciousness or even to be used as a distributed DNS database protected from various attacks and misuses.
- Consequently, specific data segments on the ledger, e.g., IPs of the end-users should be available only to themselves and remain private to all other entities.
- To achieve consensus for storing the data, peers should approve the transaction only for authorized entities who call the corresponding storing procedure. Additionally, for each transaction related to data storage processes, new blocks should be created and added to the ledger, and all the participants should update their "local" ledgers to include these new blocks.

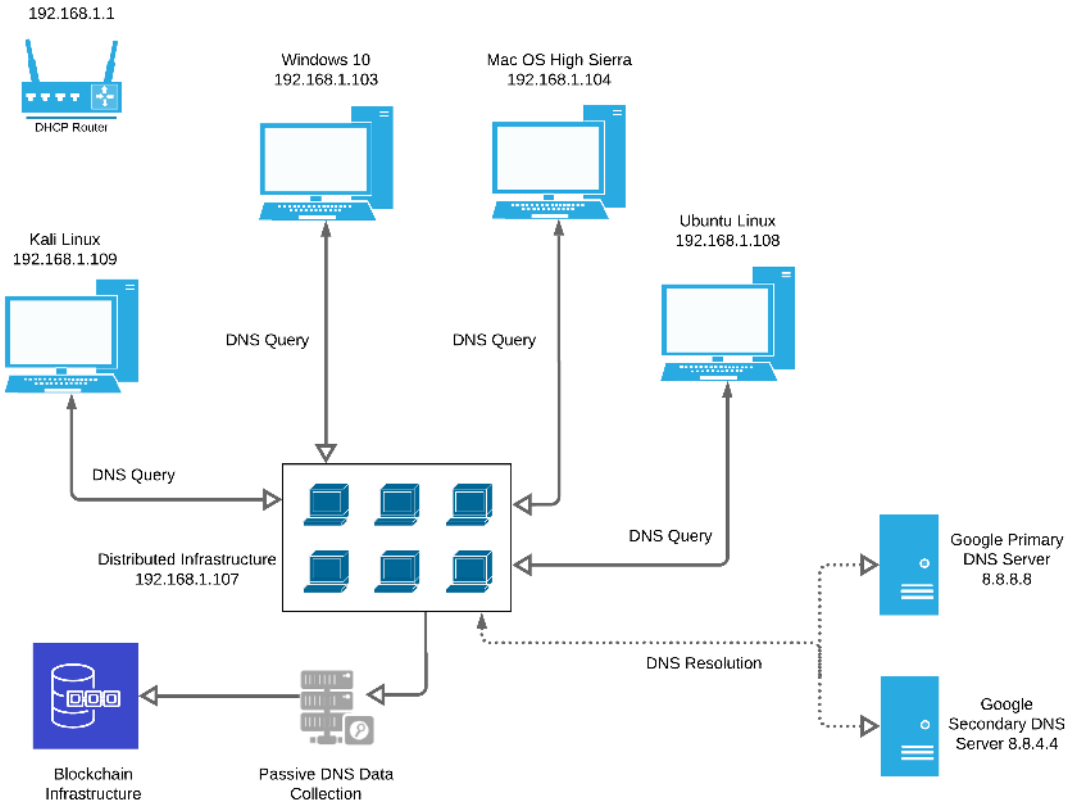


Figure 2. Privacy Preserving Passive (PRESERVE DNS) proof-of-concept implementation architecture for the test data collection.

4.2. The Blockchain

The Hyperledger Fabric platform is used for implementing the blockchain. As illustrated in Figure 3, the blockchain infrastructure is composed of two organisations of two peers each; a certificate authority; and an orderer. All the entities of the blockchain are docker containers, authorized for performing their respective purpose according to their identity. The identities issued are X.509 digital certificates signed by the Certificate Authority and checked by the MSP [48]. During the creation of the blockchain network, the state database that is going be used is defined [32]; we used *CouchDB*, a complete database available in Hyperledger Fabric that stores data in key-value pairs and also offers rich queries functionality [49]. The chaincode is written in the Go programming language.

The first organisation acts as an end-user that stores its passive DNS data collection in the blockchain. Since this data includes personal information (i.e., the IP of the client's machine and the IP

of the DNS resolver), they should remain inaccessible by all the other participants of the blockchain, including the second ISP organisation. A private data collection has been used to allow only the first organisation to query for protected data. All the other recipients are able to query only data that they are entitled to access. If an unauthorized peer attempts to query protected data, the query is rejected and an error message is returned.

Hyperledger Fabric provides chaincode APIs to use with command-line (CLI) tools. These chaincode APIs extend the functionalities of the peers and are distinguished in the Init API, Invoke API and Query API classes. Init API is used when initialization or upgrade of the chaincode is executed. Invoke API and Query API are used when storing or reading transactions to the ledger are performed [50].

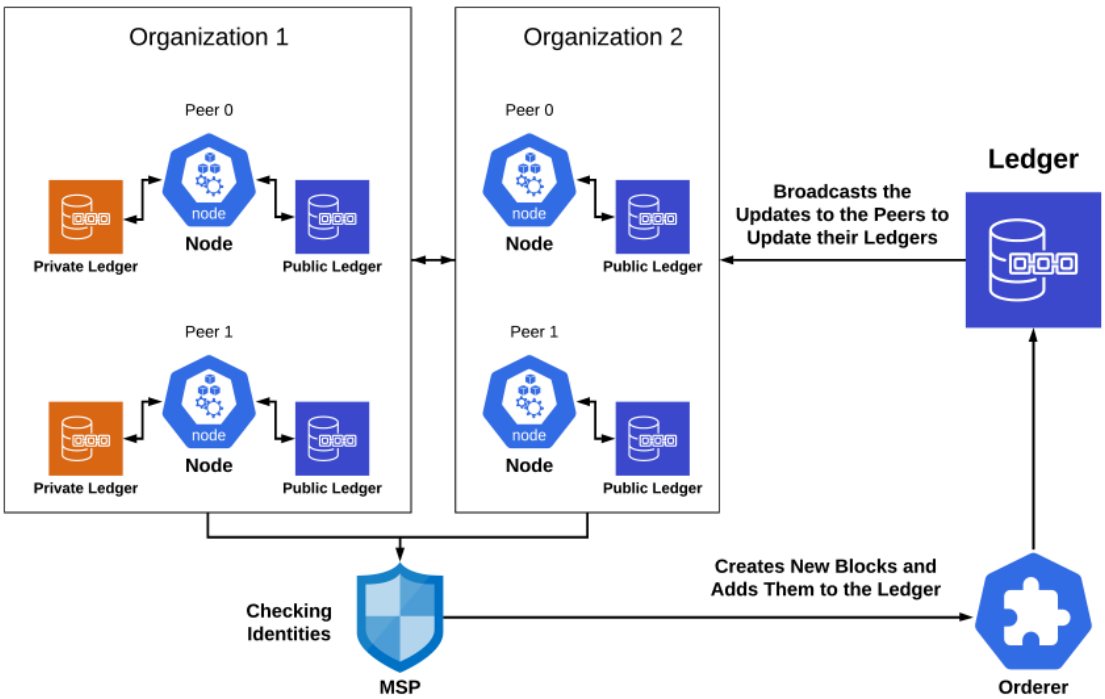


Figure 3. Hyperledger fabric infrastructure.

Peers of the blockchain can store data on the ledger using Hyperledger Fabric's Invoke API. First they have to declare their identity and then use the Invoke API with the corresponding storing function and the arguments in JSON format to send each transaction to the orderer. The orderer receives the data and performs the requested storing function. In case of success, this procedure will create a new block on the ledger and will send an update signal to each of the peers to update their ledgers.

To receive data from the ledger, peers use the Query API. They declare their identity and then use the corresponding query function with the arguments in JSON format to send the transaction to the orderer. The orderer receives the transaction and subsequently displays only those data to the recipients that they are allowed to access according to the defined query function and the private data collection configuration. The specified identity functionality enables a peer to query only specific blocks. Some data, such as the IPs of each end-user, must remain private and available only to them.

Some fundamental differences exist between the "traditional" public and the "permissioned" blockchain infrastructures as discussed in Sections 2.2 and 2.3. One of these is the consensus mechanism, which in Hyperledger Fabric can be configured. The consensus mechanism used in the proof-of-concept implementation of PRESERVE DNS requires at least one peer from any organization to accept a transaction, in order for the transaction to be considered valid. Another major difference is the time taken for transaction processing [51]. Bitcoin requires approximately ten minutes per new block that contains a few transactions. Compared to that, Hyperledger Fabric can process a few thousands transactions per second [32], while maintaining the promoted privacy and security. A similar feature

in-context in "traditional" blockchains with Hyperledger Fabric is the Peers entity that could be compared to "miners" or full nodes [52].

5. Evaluation

5.1. Security Evaluation—DNS Attacks

Common DNS attacks are DNS DDoS [53], DNS fast-flux [54] and DNS cache poisoning [55]; PRESERVE DNS can thwart these attacks as follows:

- The proof-of-concept implementation of PRESERVE DNS described in the previous section contains only one orderer. A potential DDoS attack against the orderer container may result in particular writes to the ledger to be blocked. However, in a production environment, this attack can be prevented using more ordering services under the same Kafka cluster. When one orderer fails, the Kafka cluster assigns another orderer to complete the transaction.
- In a fast-flux DNS attack a malicious actor uses short-timed time-to-live (TTL) records to change legitimate to malicious servers under the same hostname. PRESERVE DNS thwarts Fast-flux DNS attacks, since the administrator of the blockchain configures the TTL of the ledger's blocks.
- PRESERVE DNS is able to thwart DNS cache poisoning attacks, but only if it is being used as the local DNS database in the system, as in the case of our proof-of-concept implementation. This means that the local DNS resolver should query PRESERVE DNS for every DNS query instead of using the local DNS cache first. A potential solution to this issue in a production environment is to continually update the local DNS cache with the data from the blockchain using a scheduled job.

5.2. Security Evaluation—Blockchain Attacks

Blockchain is an immutable ledger, and the data stored cannot be manipulated by malicious actors. Each transaction needs to be authorized by the policy, and unauthorized requests are rejected automatically. All the functions and security mechanisms of the blockchain are included in the chaincode that is installed in each participant. A collection configuration is developed to advise the orderer about the state of the stored data, the time of their availability until they purge, and all entities that have access to them.

Each peer is obliged to prove its identity to the orderer before being allowed to perform a transaction. According to the configured policy, the store and query transactions are restricted to peers which are not included in the policy, thus preserving the privacy of the stored data. These fundamental principles eliminate the possibility of an unauthorized, malicious actor to store arbitrary data to the ledger. Furthermore, a malicious actor is overall unable to query data. The personal data can be queried only by specified entities and the remaining data are available only to participants.

DDoS attacks against the blockchain are thwarted as well. Hyperledger Fabric uses docker containers to act as peers of the blockchain. Each peer has the whole ledger stored, including the history of each transaction. PRESERVE DNS is composed of Docker containers acting as the participants of the network. This means that when a Docker container fails the query is passed on to another active peer. When the failed peer recovers, it initiates the gossip protocol to update its ledger with missing entries, from the rest of the peers. For a successful DDoS attack to take place against PRESERVE DNS, all the peers should be successfully attacked at the same time; this is highly improbable and practically impossible. Another solution to this problem is to develop the docker containers inside a Kubernetes cluster. In case of a container failure, Kubernetes is able to restart it or create an identical twin to operate as the failed one.

Attacks such as DNS amplification attacks and some zero-day DNS attacks against the blockchain infrastructure are not possible, because to launch them data in the ledger would need to be altered. PRESERVE DNS is operationally resilient, since it is composed of distributed peer nodes and organisations that have identical ledgers.

Further, infrastructure continuity in PRESERVE DNS is ensured as long as at least one peer node is operating normally [56]. Accordingly, PRESERVE DNS has no single point of failure.

A significant issue for permissioned blockchains such as the Hyperledger Fabric is the human factor, considered to be the weakest link in any system's security chain. Key management is a crucial part of the blockchain infrastructure's security. Potential theft of a legitimate user's identity by a malicious actor can lead to unauthorized queries and arbitrary storage of data in the ledger. Additionally, the chaincode is compiled and can run completely autonomously, but it is coded by humans. Thus, it is possible that the code contains bugs that are discovered late in the infrastructure lifecycle [32].

Despite the enhanced security that the blockchain technology enjoys, it is still young, and new kinds of attacks are most likely to be discovered in the future. One potential source of threats is the advancement of quantum computing, against which most of the existing cryptographic techniques are defenseless. Thus, whenever possible, it is recommended to use a quantum-robust algorithm if its overhead on the performance of the system is tolerable [57].

5.3. Performance Evaluation

The last part of the evaluation compares the performance of PRESERVE DNS against an alternative blockchain solution, namely Blockstack, and against a traditional database that offers column level privacy, such as PostgreSQL. Blockstack uses by default the Gaia decentralised storage system [58]. Even if it does not offer the enhanced privacy of PRESERVE DNS, it is a promising alternative to the current DNS infrastructure. On the other hand, the PostgreSQL database is the most popular option for production environments because it is a database server that offers extra functionalities, such as remote connections [59]. Additionally, the PostgreSQL database server can run isolated in a Docker container [60].

As already discussed in Section 3, the DNSTSM system [43] is very similar to ours. However, being based on an older version of Hyperledger Fabric, it is unable to create a complete privacy-preserving infrastructure. Hence, we did not compare its performance to that of PRESERVE DNS, and we limited ourselves to a comparison of security levels, presented in Table 1.

The time taken to perform a (a) Read data and (b) Write Data transaction in each of the three alternatives (PRESERVE DNS, PostgreSQL database and Blockstack's Gaia decentralised storage) for various numbers of DNS entries (10, 1000, 10,000, 100,000, 1,000,000) is used as the performance metric. The results are depicted in Table 2 and in Figure 4. As illustrated in Figure 4 the PostgreSQL database query time (or query overhead) is less for a small number of DNS Entries but raises linearly with each additional DNS entry. Data stored in Blockstack's Gaia decentralised storage is in the form of key-value pairs and is stored off-chain. On the other hand, the query time in PRESERVE DNS is unchanged, since it queries indexed items stored in a distributed ledger. The benefits of using a distributed solution, such as PRESERVE DNS, are even greater in production environments because the passive DNS data analysis consists of millions of DNS entries and scaling is necessary. All comparisons were done over the same architecture, illustrated in Section 4; they were executed independently and in isolation, to eliminate bias and ensure the accuracy of the results.

Furthermore, we benchmarked the CPU and Memory performance of our proposed solution. Figure 5 shows that the CPU usage of the blockchain nodes during Read queries workflow is low (5–10%) over a varying number of DNS entries (1000, 10,000, 100,000). Additionally, the Write queries workflow follows a similar CPU usage pattern (<20%). The blockchain nodes in the form of Docker containers are distinguished in the first and second Peers (Peer0, Peer1) of Organization 1 and Organization 2 (Org1, Org2). Moreover, the last *CLI* container is the container that is being used by Hyperledger Fabric for the command-line interface usability [61]. In Figure 5d,e, the *CLI* container's CPU usage fluctuates rapidly and conceals the CPU usage of the blockchain nodes. This is reasonable as the *CLI* container is being directly used by the authors for each transaction. Furthermore, it is also noteworthy that the CPU usage of the blockchain nodes is less than 20%.

Table 2. Read Data/Write Data transaction time in milliseconds (ms) per number of DNS entries.

Number of DNS Entries		10	1000	10,000	100,000	1,000,000
PRESERVE DNS	Read Data	180 ms	180 ms	180 ms	180 ms	180 ms
	Write Data	230 ms	230 ms	230 ms	230 ms	230 ms
PostgreSQL Database	Read Data	2 ms	3 ms	10 ms	44 ms	220 ms
	Write Data	4 ms	5 ms	6 ms	9 ms	11 ms
Blockstack Ali et al. [58]	Read Data	360 ms	360 ms	360 ms	360 ms	360 ms
	Write Data	530 ms	530 ms	530 ms	530 ms	530 ms

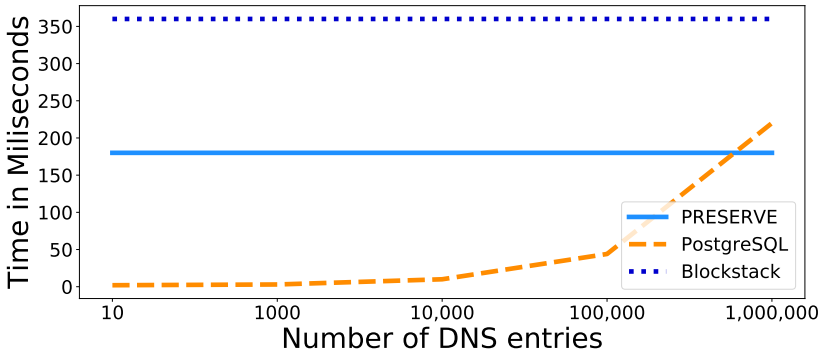
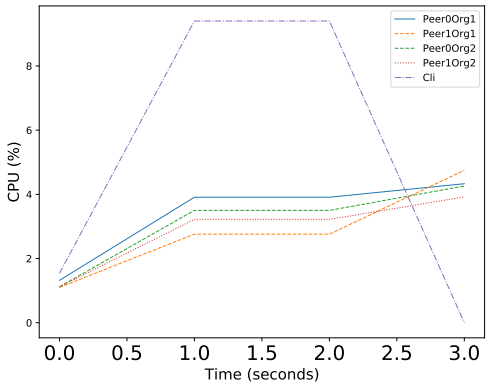
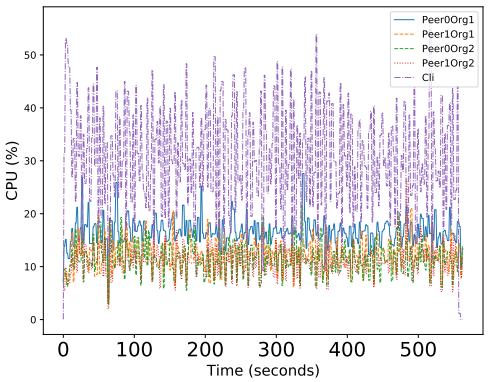


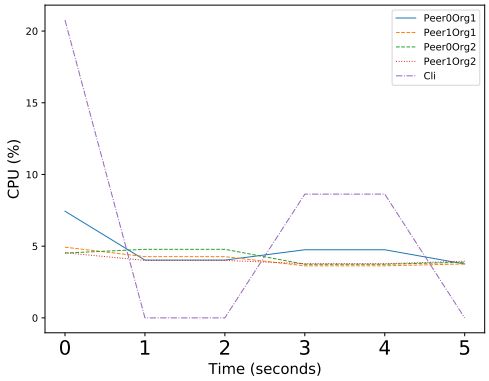
Figure 4. Read Data transactions overhead.



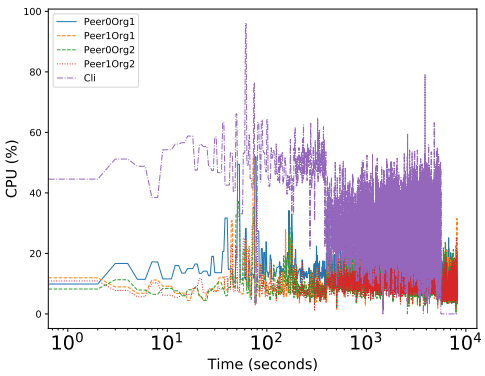
(**a**) Read queries workflow on 1000 DNS Entries



(**b**) Write queries workflow on 1000 DNS Entries

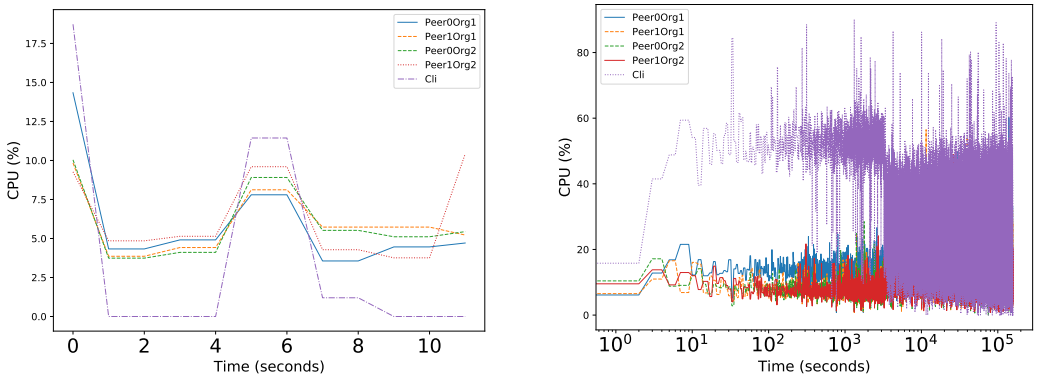


(**c**) Read queries workflow on 10,000 DNS Entries



(**d**) Write queries workflow on 10,000 DNS Entries

Figure 5. *Cont.*

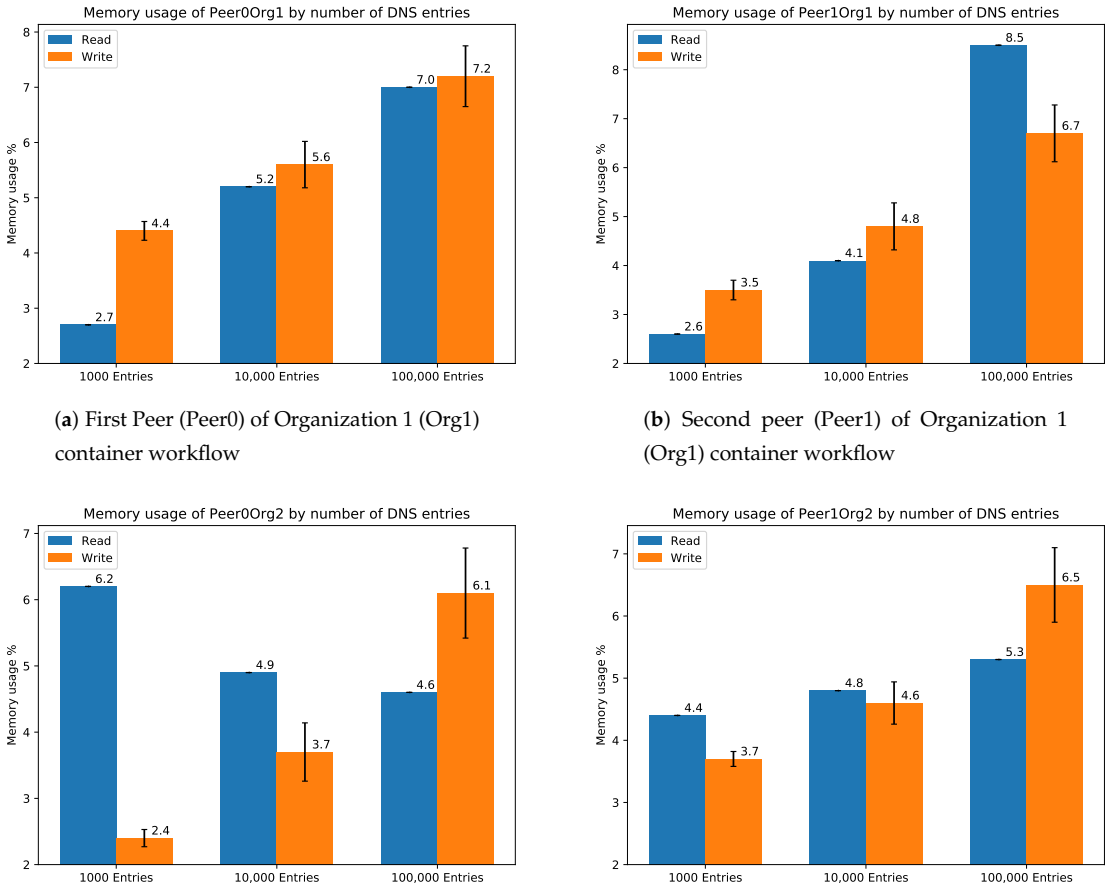


(**e**) Read queries workflow on 100,000 DNS Entries

(**f**) Write queries workflow on 100,000 DNS Entries

Figure 5. CPU Usage (%) of Nodes during workflow.

Figure 6 depicts the average memory usage of each blockchain node, in addition to the minimum and maximum values scored. The memory usage for each Read and Write query measured over a varying number of DNS entries (1000, 10,000, 100,000) is considerably low. In this figure the naming of the docker containers follows the convention described above. Contrary to Figure 5, the memory usage of the *CLI* container does not fluctuate significantly.



(**a**) First Peer (Peer0) of Organization 1 (Org1) container workflow

(**b**) Second peer (Peer1) of Organization 1 (Org1) container workflow

(**c**) First Peer (Peer0) of Organization 2 (Org2) container workflow

(**d**) Second Peer (Peer1) of Organization 2 (Org2) container workflow

Figure 6. Memory Usage (%) of Nodes during workflow on 1000, 10,000 and 100,000 DNS Entries.

6. Conclusions and Future Work

DNS is the Internet's phone book. Its services are and will continue to be invaluable for years to come. Its main setback is that it was built without strong consideration of its security, thus allowing malicious parties to abuse it. Part of the defence against such abuse is the processing of DNS resolution data, collected by passive DNS, towards identifying malicious actors such as, e.g., malicious domain names. However, this process involves collecting and processing personal end-user data. As the whole world becomes data centric, securing such data becomes paramount, hence the need for privacy-preserving passive DNS.

We proposed PRESERVE DNS, an environment that can be used for passive DNS data analysis whilst ensuring end-user privacy. We provided a proof-of-concept implementation and we evaluated the performance of PRESERVE DNS against another blockchain solution, namely Blockstack, and against a traditional database that offers column level privacy, such as PostgreSQL. PRESERVE DNS was found to be resilient to various attacks and to impose less query overhead in high volumes of data than that of existing alternatives. In our proof-of-concept implementation, PRESERVE DNS achieves a read query time of 180 ms in every tested number of DNS entries. Similarly, in every tested number of DNS entries, Blockstack achieves a flat read query time of 360 ms. In contrast, the traditional PostgreSQL database read query time increases over higher volumes of DNS entries, achieving 220 ms in 1,000,000 entries.

Existing techniques for analysing passive DNS data, such as Notos [62], EXPOSURE [63], and [12] can use PRESERVE DNS to achieve the same functionality without violating the privacy of the end-users. PRESERVE DNS can also be set-up online, to provide further services to end-users. It can be used as a public DNS resolver that also passively collects the queries and responses, for malicious domain name detection, transparently to the end-users. PRESERVE DNS can be part of an ISP's or a TLD's infrastructure, or it can be used as part of a passive DNS data analysis system. In the latter option, users that store their DNS data to the system for further analysis can be sure that their privacy is not violated and only themselves may query their personal data. PRESERVE DNS achieves this trustworthiness without the need for users to trust another third party or the system itself.

As future work, we plan to extend the functionality of PRESERVE DNS, and to produce a cloud-oriented implementation, that would facilitate its adoption by ISPs. The advantages of placing the services of PRESERVE DNS in the cloud or on a cloud service provider could multiply if it is placed on a Kubernetes cluster. This would provide operational continuity with semi-infinite scalability and expandability. Since PRESERVE DNS uses the Hyperledger Fabric platform that consists of Docker containers, making a transition to a Kubernetes cluster is a viable and realistic option.

Author Contributions: All authors contributed in the conceptualization and methodology of the manuscript; P.P. performed the data preparation; P.P. and N.P. contributed in writing; W.J.B., O.L., S.K. reviewed and edited the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

References

1. Mockapetris, P.V. Domain Names-Concepts and Facilities. RFC 1034. 1987. Available online: https://dl.acm.org/doi/book/10.17487/RFC1034 (accessed on 11 August 2012).
2. Antonakakis, M.; April, T.; Bailey, M.; Bernhard, M.; Bursztein, E.; Cochran, J.; Durumeric, Z.; Halderman, J.A.; Invernizzi, L.; Kallitsis, M.; et al. Understanding the mirai botnet. In Proceedings of the 26th USENIX Security Symposium (USENIX Security 17), Vancouver, BC, Canada, 16–18 August 2017; pp. 1093–1110.
3. Vissers, T.; Joosen, W.; Nikiforakis, N. Parking sensors: Analyzing and detecting parked domains. In Proceedings of the 22nd Network and Distributed System Security Symposium (NDSS 2015), San Diego, CA, USA, 8–11 February 2015; Internet Society: Reston, VA, USA; pp. 53–53.

4. Stout, B.; McDowell, K. System and Method for Combating Cybersquatting. U.S. Patent 8,285,830, 9 October 2012.
5. Weimer, F. Passive DNS replication. In Proceedings of the FIRST Conference on Computer Security Incident, Singapore, 26 June–1 July 2005; p. 98.
6. Patrick Breyer v Bundesrepublik Deutschland Case C-582/14 ECLI:EU:C:2016:779. 2016. Available online: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A62014CJ0582 (accessed on 11 August 2020).
7. Spring, J.M.; Huth, C.L. The impact of passive dns collection on end-user privacy. *Secur. Trust. Internet Names* **2012**, 1–11.
8. Google. Google Public DNS. 2018. Available online: https://developers.google.com/speed/public-dns/privacy (accessed on 11 August 2020).
9. Cloudflare. What Is 1.1.1.1? 2018. Available online: https://developers.cloudflare.com/1.1.1.1/what-is-1.1.1.1 (accessed on 11 August 2020).
10. OpenDNS. 2006. Available online: https://www.opendns.com/about/global-dns-infrastructure/ (accessed on 11 August 2020).
11. Federrath, H.; Fuchs, K.P.; Herrmann, D.; Piosecny, C. Privacy-preserving DNS: Analysis of broadcast, range queries and mix-based protection methods. In *European Symposium on Research in Computer Security*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 665–683.
12. Khalil, I.; Yu, T.; Guan, B. Discovering malicious domains through passive DNS data graph analysis. In Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, Xi'an, China, 30 May–3 June 2016; ACM: New York, NY, USA, 2016; pp. 663–674.
13. Gasser, O.; Hof, B.; Helm, M.; Korczynski, M.; Holz, R.; Carle, G. In log we trust: Revealing poor security practices with certificate transparency logs and internet measurements. In Proceedings of the International Conference on Passive and Active Network Measurement, Berlin, Germany, 26–27 March 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 173–185.
14. Farsight Security. *DNSDB*; Farsight Security: SAN Mateo, CA, USA, 2010.
15. VirusTotal. *VirusTotal Passive DNS Replication*; VirusTotal: Dublin, Ireland, 2013.
16. James, L. *Phishing Exposed*; Elsevier: Amsterdam, The Netherlands, 2005.
17. Yoon, C.; Kim, K.; Kim, Y.; Shin, S.; Son, S. Doppelgängers on the dark web: A large-scale assessment on phishing hidden web services. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 2225–2235.
18. Riederer, C.; Erramilli, V.; Chaintreau, A.; Krishnamurthy, B.; Rodriguez, P. For sale: Your data: By: You. In Proceedings of the 10th ACM WORKSHOP on Hot Topics in Networks, Cambridge, MA, USA, 14–15 November 2011; pp. 1–6.
19. Bortzmeyer, S. DNS Privacy Considerations. Available online: https://www.bortzmeyer.org/7626.html (accessed on 11 August 2020).
20. Wang, W.; Hoang, D.T.; Hu, P.; Xiong, Z.; Niyato, D.; Wang, P.; Wen, Y.; Kim, D.I. A survey on consensus mechanisms and mining strategy management in blockchain networks. *IEEE Access* **2019**, *7*, 22328–22370. [CrossRef]
21. Benisi, N.Z.; Aminian, M.; Javadi, B. Blockchain-based decentralized storage networks: A survey. *J. Netw. Comput. Appl.* **2020**, 102656. [CrossRef]
22. Liu, J.; Li, B.; Chen, L.; Hou, M.; Xiang, F.; Wang, P. A data storage method based on blockchain for decentralization DNS. In Proceedings of the 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), Guangzhou, China, 18–21 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 189–196.
23. Di Pierro, M. What is the blockchain? *Comput. Sci. Eng.* **2017**, *19*, 92–95. [CrossRef]
24. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Technical Report. Manubot. 2019. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 11 August 2020).
25. 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; IEEE: Piscataway, NJ, USA, 2017; pp. 557–564.
26. Dai, H.N.; Zheng, Z.; Zhang, Y. Blockchain for internet of things: A survey. *IEEE Int. Things J.* **2019**, *6*, 8076–8094. [CrossRef]
27. Luo, J.; Chen, Q.; Yu, F.R.; Tang, L. Blockchain-enabled software-defined industrial internet of things with deep reinforcement learning. *IEEE Int. Things J.* **2020**, *7*, 5466–5480. [CrossRef]

28. Xu, Q.; Su, Z.; Dai, M.; Yu, S. APIS: Privacy-preserving incentive for sensing task allocation in cloud and edge-cooperation mobile internet of things with SDN. *IEEE Int. Things J.* **2019**, *7*, 5892–5905. [CrossRef]
29. Wu, M.; Wang, K.; Cai, X.; Guo, S.; Guo, M.; Rong, C. A comprehensive survey of blockchain: From theory to IoT applications and beyond. *IEEE Int. Things J.* **2019**, *6*, 8114–8154. [CrossRef]
30. Karandikar, N.; Chakravorty, A.; Rong, C. Transactive energy on hyperledger fabric. In Proceedings of the 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS), Granada, Spain, 22–25 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 539–546.
31. Xu, X.; Weber, I.; Staples, M.; Zhu, L.; Bosch, J.; Bass, L.; Pautasso, C.; Rimba, P. A taxonomy of blockchain-based systems for architecture design. In Proceedings of the 2017 IEEE International Conference on Software Architecture (ICSA), Gothenburg, Sweden, 3–7 April 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 243–252.
32. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; De Caro, A.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger fabric: A distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*; ACM: New York, NY, USA, 2018; p. 30.
33. Hyperledger Fabric. Private Data. 2019. Available online: https://hyperledger-fabric.readthedocs.io/en/release-2.2/private-data/private-data.html (accessed on 11 August 2020).
34. Dhillon, V.; Metcalf, D.; Hooper, M. *Blockchain Enabled Applications: Understand the Blockchain Ecosystem and How to Make It Work for You*; Springer: Berlin/Heidelberg, Germany, 2017.
35. Zhao, W.; Yang, S.; Luo, X. On consensus in public blockchains. In Proceedings of the 2019 International Conference on Blockchain Technology, Atlanta, GA, USA, 14–17 July 2019; pp. 1–5.
36. Zdrnja, B. *Security Monitoring of DNS Traffic*; University of Auckland: Auckland, New Zealand, 2006.
37. Govil, J.; Govil, J. 4G mobile communication systems: Turns, trends and transition. In Proceedings of the 2007 International Conference on Convergence Information Technology (ICCIT 2007), Gyeongju, Korea, 21–23 November 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 13–18.
38. Xu, J.; Fan, J.; Ammar, M.H.; Moon, S.B. Prefix-preserving ip address anonymization: Measurement-based security evaluation and a new cryptography-based scheme. In Proceedings of the 10th IEEE International Conference on Network Protocols, Paris, France, 12–15 November 2002; IEEE: Piscataway, NJ, USA, 2002; pp. 280–289.
39. Kountouras, A.; Kintis, P.; Lever, C.; Chen, Y.; Nadji, Y.; Dagon, D.; Antonakakis, M.; Joffe, R. Enabling network security through active DNS datasets. In *International Symposium on Research in Attacks, Intrusions, and Defenses*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 188–208.
40. Liang, X.; Shetty, S.; Tosh, D.; Kamhoua, C.; Kwiat, K.; Njilla, L. Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, Madrid, Spain, 14–17 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 468–477.
41. Kalodner, H.A.; Carlsten, M.; Ellenbogen, P.; Bonneau, J.; Narayanan, A. *An Empirical Study of Namecoin and Lessons for Decentralized Namespace Design*; WEIS: Sunbury, PA, USA, 2015.
42. Ali, M.; Nelson, J.; Shea, R.; Freedman, M.J. Blockstack: A global naming and storage system secured by blockchains. In Proceedings of the 2016 USENIX Annual Technical Conference (USENIX ATC16), Denver, CO, USA, 22–24 June 2016; pp. 181–194.
43. Yu, Z.; Xue, D.; Fan, J.; Guo, C. DNSTSM: DNS cache resources trusted sharing model based on consortium blockchain. *IEEE Access* **2020**, *8*, 13640–13650. [CrossRef]
44. Liu, C.; Albitz, P. *DNS and Bind*; O'Reilly Media, Inc.: Newton, MA, USA, 2006.
45. Vayghan, L.A.; Saied, M.A.; Toeroe, M.; Khendek, F. Deploying microservice based applications with Kubernetes: experiments and lessons learned. In Proceedings of the 2018 IEEE 11th international conference on cloud computing (CLOUD), San Francisco, CA, USA, 2–7 July 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 970–973.
46. Ager, B.; Mühlbauer, W.; Smaragdakis, G.; Uhlig, S. Comparing DNS resolvers in the wild. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*; ACM: New York, NY, USA, 2010; pp. 15–21.
47. Fjellskal, E. Gamelinux Passive DNS. 2011. Available online: https://github.com/gamelinux/passivedns (accessed on 11 August 2020).
48. Hyperledger Fabric. Certificates Github. 2019. Available online: https://github.com/hyperledger/fabric-ca (accessed on 11 August 2020).

49. Thakkar, P.; Nathan, S.; Viswanathan, B. Performance benchmarking and optimizing hyperledger fabric blockchain platform. In Proceedings of the 2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Milwaukee, WI, USA, 25–28 September 2018; IEEE: Piscataway, NJ, USA; pp. 264–276.
50. Hyperledger Fabric. Chaincode for Developers. 2019. Available online: https://hyperledger-fabric.readthedocs.io/en/release-2.0/chaincode4ade.html (accessed on 11 August 2020).
51. Sompolinsky, Y.; Zohar, A. Secure high-rate transaction processing in bitcoin. In Proceedings of the International Conference on Financial Cryptography and Data Security, San Juan, Puerto Rico, 26–30 January 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 507–527.
52. Bashir, I. *Mastering Blockchain*; Packt Publishing Ltd.: Birmingham, UK, 2017.
53. Kambourakis, G.; Moschos, T.; Geneiatakis, D.; Gritzalis, S. Detecting DNS amplification attacks. International Workshop on Critical Information Infrastructures Security, Malaga, Spain, 3–5 October 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 185–196.
54. Ranjan, S. Detecting DNS Fast-Flux Anomalies. U.S. Patent 8,260,914, 4 September 2012.
55. Schuba, C. Addressing Weaknesses in the Domain Name System Protocol. Master's Thesis, Purdue University, West Lafayette, IN, USA, 1993.
56. Piscini, E.; Dalton, D.; Kehoe, L. *Blockchain and Cyber Security. Let's Discuss*; Deloitte: London, UK, 2017.
57. English, E.; Kim, A.D.; Nonaka, M. Advancing Blockchain Cybersecurity: Technical and Policy Considerations for the Financial Services Industry. *Cybersecur. Policy Resil.* **2018**, *81*.
58. Ali, M.; Shea, R.; Nelson, J.; Freedman, M.J. Blockstack: A new decentralized internet. *Whitepaper*; 2017.
59. Momjian, B. *PostgreSQL: Introduction and Concepts*; Addison-Wesley: New York, NY, USA, 2001; Volume 192.
60. Boettiger, C. An introduction to Docker for reproducible research. *ACM SIGOPS Oper. Syst. Rev.* **2015**, *49*, 71–79. [CrossRef]
61. Hyperledger Fabric. Command-Line Interface (CLI). 2019. Available online: https://bmos299-fabric.readthedocs.io/en/latest/API/CLI.html (accessed on 11 August 2020)
62. Antonakakis, M.; Perdisci, R.; Dagon, D.; Lee, W.; Feamster, N. Building a dynamic reputation system for dns. In Proceedings of the USENIX Security Symposium, Washington, DC, USA, 11–13 August 2010; pp. 273–290.
63. Bilge, L.; Kirda, E.; Kruegel, C.; Balduzzi, M. *EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis*; Ndss: New York, NY, USA, 2011; pp. 1–17.