Privacy-Preserving for Web Hosting

Tam T. Huynh[1](✉), Thuc D. Nguyen[2], Nhung T.H. Nguyen[2], and Hanh Tan[1]

[1] Posts and Telecommunications Institute, Ho Chi Minh City, Vietnam
{tamht,tanhanh}@ptithcm.edu.vn
[2] University of Science, Ho Chi Minh City, Vietnam
{ndthuc,nthnhung}@fit.hcmus.edu.vn

Abstract. Privacy-preserving for data is one of the crucial responsibilities of service providers. For the web hosting service, customers' information and source codes of websites are considered sensitive data that need to protect. The solution decentralized for web hosting is a new technology trend, provides an effective mechanism for storing and accessing websites. This paper addresses some problems related to privacy concerns of the decentralized web hosting service. Based on the blockchain technology, cryptography, and interplanetary file system (IPFS) platform, we propose a protocol for web hosting that provides three features. The first one ensures anonymity for information of customers. The second feature provides confidentiality and authentication for transmitting source codes of websites between customers and the service provider (SP). And the last one is responsible for securing the source code of websites from other nodes on the public IPFS network. The experiments demonstrate that the proposed solution efficiently protects privacy for the decentralized web hosting service.

Keywords: IPFS · Blockchain · Privacy · Web hosting

1 Introduction

Nowadays, most websites are hosted based on the client-server model [1], where, the centralized servers are responsible for storing source codes and data of websites and handling queries of users, and users can use their web browsers to send requests to these web servers. Some challenges for web hosting based on this model: (1) with websites provide data sharing features, when the number of requests to large files increases significantly, the network traffic becomes more congested and the hosting servers become slower in responding. To improve data retrieval speed, content delivery networks can be used to cache and deliver contents to users based on their geographic location [2–4]. However, these solutions may leak information about users [5], and have some challenges concerning scalability, quality of service and flexibility [6], moreover, these entire systems are still the centralized model. (2) If the data is not well protected and centralized systems stop working due to overload, denial-of-service (DoS) or distributed denial-of-service (DDoS) attacks, system errors, etc., users cannot access the websites at that time.

To solve these challenges, the authors in [7] used IPFS to build distributed web applications. In [8] we proposed a decentralized solution for web hosting, which was a

N.-S. Vo and V.-P. Hoang (Eds.): INISCOM 2020, LNICST 334, pp. 314–323, 2020.
https://doi.org/10.1007/978-3-030-63083-6_24

combination of the IPFS protocol and the blockchain technology to form a decentralized platform for the web hosting service. IPFS, was proposed by Juan Benet in 2014 [9], is a distributed file system, provides efficient methods for data storage and retrieval. It is also considered the distributed and permanent website. Files on IPFS are segmented inside objects of 256 kbyte, identified by the SHA-256 hash function. Nodes on the IPFS network connect and transfer objects to each other over a peer-to-peer network. Each node owns a distributed hash table to locate the nodes that store certain objects being requested. Nodes can use the IPFS pinning feature to keep the objects available to the community, and can activate the interplanetary name space (IPNS) service for creating and updating mutable links to contents of websites [9, 10]. Meanwhile, the blockchain technology provides anonymity, transparency, decentralization, auditability [11–13], is used to conduct transactions between customers and the SP. All valid transactions are mined and stored on the shared ledger. To easily deploy electronic contracts between customers and the SP, we select the ethereum blockchain which is a new distributed blockchain network [14].

In this paper, we present three models of web hosting, and illustrate the pros and cons of them. We also propose an overall protocol used for web hosting that provides three privacy-preserving features as follows: (1) ensure anonymity of customers' information; (2) secure data sharing between customers and the SP over a peer-to-peer network; (3) protect source codes of websites from illegal copying. Our experimental results also demonstrate the improvement of privacy-preserving for web hosting, and it is a suitable and necessary protocol for the web hosting service provider based on the distributed platform.

The rest of the paper is organized as follows: Sect. 2 describes the three deployment models for web hosting and analyzes the privacy requirements for the system. Section 3 presents our proposed protocol. Section 4 reports our experimental results, and finally we conclude with Sect. 5.

2 Models and Privacy Requirements

2.1 Models

The decentralized solution for web hosting can be implemented on the three models as shown in Figs. 1 and 2. The SP can build a private IPFS network as shown in Fig. 1a, the size of the network is the number of nodes on the network. In this model, the SP uses some nodes having high performance (especially large storage capacity) to run the IPFS cluster service which is used for data replication between cluster nodes. All websites of customers are stored long-term and replicated on these nodes. Other nodes, called miner nodes, must connect to cluster nodes directly or indirectly, and act as gateways of the network. Websites are segmented and put into objects, and each miner node on the network can cache some objects or all objects of some websites depend on its storage capacity. Objects are stored on miner nodes temporarily and removed periodically by the garbage collection feature or when the cache memory is full. In order to access websites hosted on the SP's network, there are two ways that users can select. One is to use web browsers to access websites through one of the gateway nodes published by the SP, this connection model can be considered the sub-client-server models for the private IPFS

network, with each server is a gateway node. The gateway nodes can be congested when receiving a lot of requests from users or due to DoS/DDoS attacks; the other is that each user uses a device to join the network as a node of the network, and use a web browser to access websites via the local address of the joined device (such as http://127.0.0.1:8080/ipfs/<website>). This way is recommended for users because it reduces the network congestion at gateway nodes and users can access websites faster. In general, the model in Fig. 1a will not operate more effectively than the traditional model if the size of the network is small.

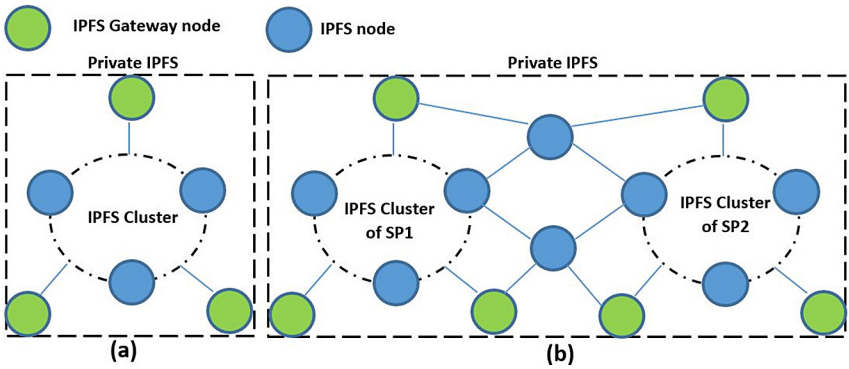

Fig. 1. The private IPFS network for web hosting.

As shown in Fig. 1b, the model is the combination of the two private IPFS networks of the two SPs to form a new private IPFS network. In fact, the more networks join, the more effective the IPFS network operates. However, the models in Fig. 1 can only operate effectively within areas that the network covers because many nodes of the SPs are always in active state to serve users. Users in other regions do not always join the network to access or cache data for other nodes in the regions, hence the stability of the IPFS network will not be guaranteed. Therefore, the model in Fig. 1b is suitable for SPs whose websites only serve users in a specific geographical area such as a region or a country.

In order to serve a lot of users around the world effectively. The SP's network needs to join the public IPFS network. Currently, the *ipfs.io* is one of the largest public IPFS networks, is built by the Protocol Labs [15]. Nodes of the SP can easily join this network to form a private IPFS network and operate based on the public network. In Fig. 2, nodes 1, 2, 3, and 4 of the SP join the public IPFS network. These nodes only store long-term the websites of the SP's customers, other data on the network are cached in a certain period. This is the suitable model for web hosting with lower deployment costs than the two previous models, it is also selected for the experiment section.

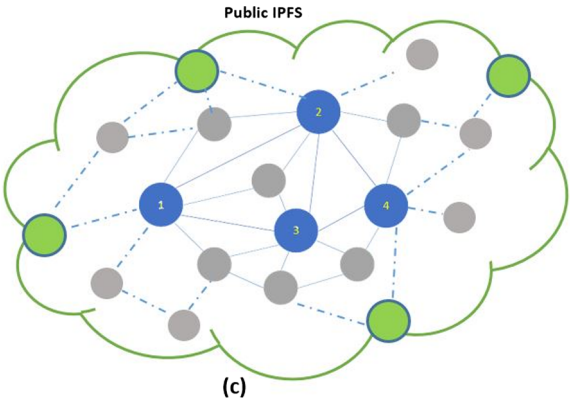

Fig. 2. The private IPFS network based on the public IPFS network.

2.2 Privacy Requirements

In order to implement this solution into practice, it is necessary to consider aspects concerning security and privacy for the service. We indicate three requirements that need to solve for the decentralized web hosting service including: protect customers' information, guarantee the security and privacy of websites' source codes when transmitting from customers to the SP for hosting, and protect websites from illegal copying.

Concerning customers' information, with the traditional method, in order to register the web hosting service, customers are required to send certain information like full name, address, phone number, credit card information, etc. This information is used for making the contract and invoice. Although a non-disclosure agreement between the customer and the SP may be signed, the SP can secretly share this sensitive data with third parties or the SP's database is compromised by hackers. In our solution, the anonymity is guaranteed by the ethereum blockchain, we write smart contracts containing a set of rules under which the parties agree to interact with each other and automatically executed once the conditions of the agreement are met. In this way, customers may create accounts on the ethereum blockchain network with anonymous information and easily conduct transactions to the SP. Transactions are validated by miners and stored in the shared public ledger of the blockchain network. In addition, the service payment can be done by Ethereum coin.

Concerning transmission of the source codes of websites from customers to the SP, our protocol also conducts this work over the public IPFS network. However, IPFS is currently not provided any solutions for privacy-preserving data storage. Hence, nodes on the IPFS network can view data through hash values that they own. To ensure this property, source codes of websites must be encrypted by a symmetric algorithm before uploading on the public IPFS network, and the secret key is sent to the SP for decryption through a secure channel.

Concerning protection for source codes of websites from illegal copying, by default hosted websites on the IPFS network are accessed from normal web browsers like Chrome, Firefox, etc. and source codes of websites are easily downloaded by any users. To overcome this problem, the SP needs to select a symmetric algorithm or build a

private encryption algorithm to encrypt source codes of websites before uploading on the network. This way guarantees the privacy of websites because nodes on the IPFS network and normal web browsers cannot understand the contents of the websites. In order to access websites, the SP has to provide an application as common web browsers, which contains the secret key of the SP to decrypting websites that hosted by the SP, and limits the view source code feature. When a user uses this app to access a certain website, it will download objects of the website, and then decrypt and show results on the application window. In order to improve security, customers can also use cryptography functions in their websites to guarantee the privacy for storage data.

3 Proposed Protocol

The proposed protocol for web hosting satisfies the privacy requirements mentioned in Sect. 2.2. Transactions and related information are transparent on a blockchain. Figure 3 describes the general diagram of the protocol for web hosting. We explain the symbols in the diagram as follows: P/O represents a person or an organization who owns a website and wants to register the web hosting service, each P/O has a public key and a private key denoted by P_PO and p_PO respectively. The web hosting service provider is denoted by SP, has a public key P_SP and a private key p_SP.

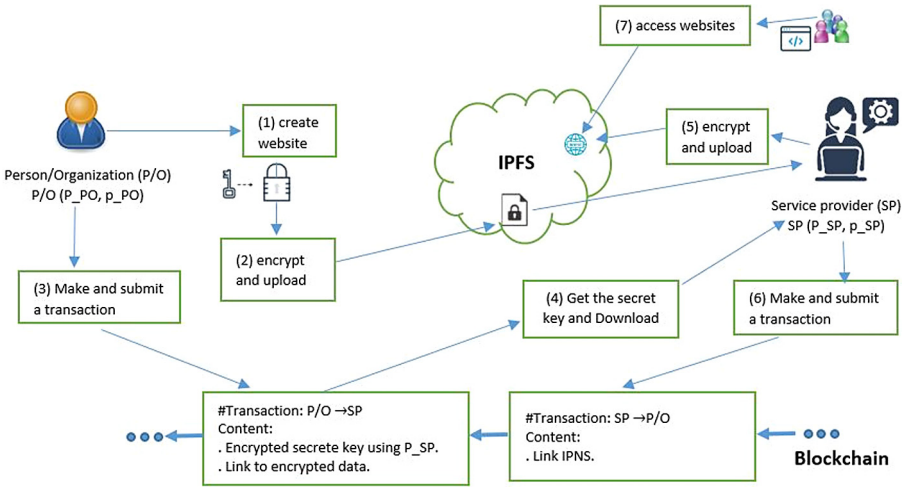

Fig. 3. The protocol diagram for web hosting.

The process of registering the service and accessing websites of the protocol as the following steps:

1. The P/O builds a website.
2. The P/O encrypts the website with a secret key (denoted by *k1*) and the selected symmetric cryptography. The P/O can use the symmetric cryptosystem provided

by the SP or in case of using a private cryptosystem, the P/O must provide this cryptosystem or the source code of the cryptosystem to the SP for decryption. Then the P/O uploads the ciphertext of the website to the public IPFS network. After the upload process has completed, the P/O receives the hash value of the ciphertext from the IPFS node.

3. The P/O performs a transaction to the blockchain network: In this transaction, besides the information about the sender and receiver addresses, it also has the following necessary information:

   - *k1*, for decrypting the original data, is encrypted by the SP's public key.
   - The link of the ciphertext on the public IPFS network.

4. The SP get *k1* and download the data: When receiving a transaction from the P/O, the SP uses p_SP to decrypt ciphertext of *k1*. Then, the SP downloads the ciphertext of the website on the IPFS network and uses *k1* to decrypt to get the source codes of the website.
5. The SP uses a symmetric cryptography and a secret key (denoted by *k2*) to encrypt the website. The output is uploaded to a cluster node of the SP. The returned hash value is mapped to a certain public key by the IPNS service of IPFS.
6. The SP creates and submits a transaction to the blockchain network to return this IPNS value to the P/O. The P/O can use this value to configure a domain name for the website by using the TXT record of the domain name system service [16].
7. In order to access the website, users have to download an application provided by the SP. This application can show websites hosted by the SP because it contains *k2* for decrypting websites.

In case the P/O wants to update the source codes of the website, the P/O has to perform steps similar to the service registration process.

4 Experiment Results

To deploy this protocol, we use the *ipfs.io* network which is a public IPFS network. For the blockchain network, we use Ganache to create a virtual ethereum blockchain. By default, Ganache generates ten different accounts, each account wallet contains 100 ETH.

We use the solidity language to build smart contracts for storing information of the web hosting service as shown in Fig. 4.

We write a simple website using HTML, CSS, and JavaScript programming languages, all files related to the website are placed in a folder named *Website*. For cryptography algorithms, we use the AES-256 algorithm in CBC mode to encrypt the *Website* as described at step 2 of our protocol. We use the RSA cryptosystem with 1024-bit keys to

```
pragma solidity >=0.4.21 <0.6.0;
import "github.com/Arachnid/solidity-
stringutils/strings.sol";
contract DataSharing {
    mapping (address => string) message;
    using strings for *;
     event AddData(string,string);
     event ReturnLink(string);
function SubmitData(address recipient, string  Key, string
link) public {
        string memory data;
        data=Key.toSlice().concat("|".toSlice());
        data=data.toSlice().concat(link.toSlice());
        data=data.toSlice().concat("|".toSlice());
        message[recipient] = data;
        emit AddData(Key, link);
    }
function SubmitIPNS(address recipient, string  ipns) public
{
        message[recipient] = ipns;
        emit ReturnLink(ipns);
    }
function GetData() public view returns (string ) {
        return (message[msg.sender]);
    }
}
```

Fig. 4. Code script of the smart contracts for web hosting.

generate key pairs for the P/O and the SP. We install the *IPFS-Desktop-Setup-0.10.4.exe* application on a computer to join the *ipfs.io* network [17], and then upload the encrypted form of the *Website* to the network. The secret key is encrypted by the public key of the SP and then sent together with the link of the file on the IPFS network to the blockchain network. The result of the transaction is shown in Fig. 5.

status	0x1 Transaction mined and execution succeed
transaction hash	0xdba5f84cfd91993d629a09b160460d7509d31f790c3ea03ce5a6cacbb91951ff
from	0x4b0897b0513fdc7c541b6d9d7e929c4e5364d2db
to	DataSharing.SubmitData(address,string,string) 0x0dcd2f752394c41875e259e00bb44fd505297caf
gas	3000000 gas
transaction cost	213694 gas
execution cost	175206 gas
hash	0xdba5f84cfd91993d629a09b160460d7509d31f790c3ea03ce5a6cacbb91951ff
input	0xa23...00000
decoded input	{ "address recipient": "0x4B0897b0513fdC7C541B6d9D7E929C4e5364D2dB", "string Key": "IvHCchsDOQWPp1jXPZIPjJyf0HHU6mGBHdG3BtdmZOocv0ci1mWXNQ5gQ3vxJ5WOoCdFfDFoV7Tp2Z20AwoK4LCzagrMogskwKe2JVCi41QWKLzj60Xtt7fE8cGI3u+M60oyDh9iUSwbCx7i7R1cSFZpCdZbY10IJGe4FCsiPiE=", "string link": "QmXJCD89bhVSwqVK3VmbP8omf69UhJ3YfrcYhePfamtPNZ"

Fig. 5. The decoded result of the transaction.

We use two virtual machines (OS Windows 7, RAM 1 GB, CPU 2.6 GHZ) joined to the public IPFS network as the nodes of the SP, Fig. 6 shows the node's information on the IPFS network. These nodes are enabled the clustering service and the pinning

feature. The *Website* is encrypted by the AES algorithm and *k2*, and then the output is uploaded to the *ipfs.io* network as shown in Fig. 7, all files of the *Website* are also encrypted as shown in Fig. 8. The *Website* is stored long-term in these two nodes.

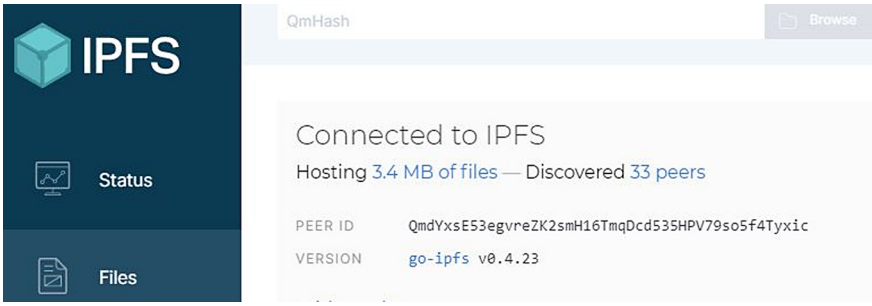

Fig. 6. The information of the SP' node on the IPFS network.

Fig. 7. The hash value of the Website folder on the IPFS network.

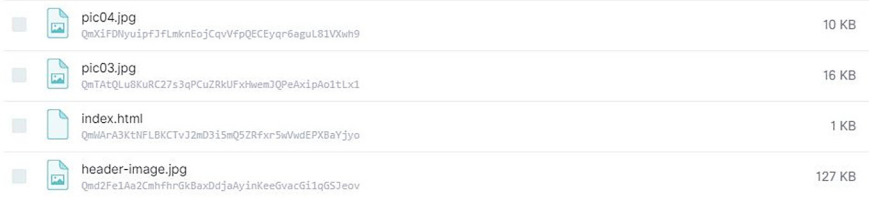

Fig. 8. Some files of the Website folder on the IPFS network.

Users can access the *Website* through the gateway address (https://www.ipfs.io/ipfs/<hash_value>) or can also access to the localhost of the devices joined to the network (http://127.0.0.1:8080/ipfs/<hash_value>). As shown in Fig. 9, the normal web browsers can access the *Website* but cannot understand its contents. Similarly, other nodes on the public IPFS network will also receive the ciphertext when accessing the hash value of the website, hence the privacy of websites is guaranteed. In order to show the *Website*, we use the C# programming language to create an application named *Web_Client* which contains *k2* for decrypting the source codes of the *Website,* as shown in Fig. 10.

Fig. 9. The result of accessing the website from a normal web browser

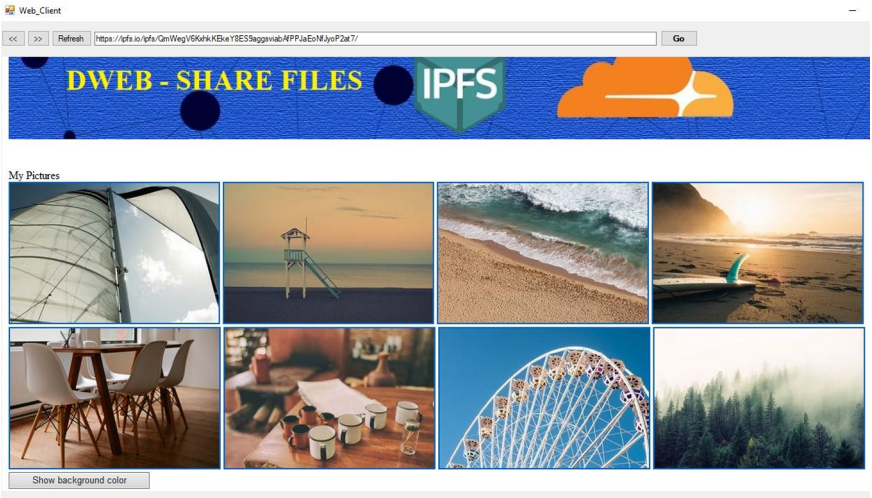

Fig. 10. The website is accessed from the Web_Client application.

5 Conclusions

Websites hosted on IPFS can access faster and more secure than the traditional model because objects of websites are identified by the cryptographic hash of their contents and objects of a certain website can retrieve from peer-to-peer nodes. In this paper, we have built the protocol for web hosting presented on a previous paper [8]. Our protocol can protect the privacy of websites on the public IPFS network. The protocol is built from the advantages of the IPFS platform for storing distribution, the blockchain technology for providing anonymous, and cryptography for protecting confidentiality and authentication. The experimental results show that our protocol operates efficiently, can easily implement in practice.

Acknowledgements. This research is funded by Vietnam National University Ho Chi Minh City (VNU-HCM) under grant number NCM2019-18-01.

References

1. Xiao, Z., et al.: A new architecture of web applications-the widget/server architecture. In: 2010 2nd IEEE International Conference on Network Infrastructure and Digital Content, pp. 866–869. IEEE (2010)

2. Shafiq, M.Z., Liu, A.X., Khakpour, A.R.: Revisiting caching in content delivery networks. In: The 2014 ACM International Conference on Measurement and Modeling of Computer Systems, pp. 567–568. ACM (2014)
3. Mokhtarian, K., Jacobsen, H. A.: Caching in video CDNs: building strong lines of defense. In: Proceedings of the 9th European Conference on Computer Systems, pp. 1–13 (2014)
4. Hosanagar, K., Krishnan, R., Smith, M., Chuang, J.: Optimal pricing of content delivery network (CDN) services. In: 37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the IEEE. IEEE (2004)
5. Sengupta, A., Tandon, R., Clancy, T.C.: Fundamental limits of caching with secure delivery. IEEE Trans. Inf. Forensics Secur., **10**(2), 355–370. IEEE (2014)
6. Fan, Q., et al.: Video delivery networks: challenges, solutions and future directions. Comput. Electr. Eng. **66**, 332–341 (2018)
7. Dias, D., Benet, J.: Distributed web applications with IPFS, tutorial. In: Bozzon, A., Cudre-Maroux, P., Pautasso, C. (eds.) ICWE 2016. LNCS, vol. 9671, pp. 616–619. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-38791-8_60
8. Huynh, T.T., Nguyen, T.D., Tan, H.: A decentralized solution for web hosting. In: 2019 6th NAFOSTED Conference on Information and Computer Science (NICS), pp. 82–87. IEEE (2019)
9. Benet, J.: IPFS-content addressed, versioned, P2P file system. arXiv preprint arXiv:1407.3561 (2014)
10. Steichen, M., Fiz, B., Norvill, R., Shbair, W., State, R.: Blockchain-based, decentralized access control for IPFS. In: 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 1499–1506. IEEE (2018)
11. Huynh, T.T., Nguyen, T. D., Tan, H.: A survey on security and privacy issues of blockchain technology. In: 2019 International Conference on System Science and Engineering (ICSSE), pp. 362–367. IEEE (2019)
12. Zheng, Z., Xie, S., Dai, H. N., Wang, H.: Blockchain challenges and opportunities: a survey. In: International Journal of Web and Grid Services, pp. 352–375 (2018)
13. Conti, M., Kumar, S., Lal, C., Ruj, S.: A survey on security and privacy issues of bitcoin. IEEE Commun. Surv. Tutorials, **20**(4), 3416–3452. IEEE (2018)
14. Buterin, V.: A next-generation smart contract and decentralized application platform. White paper 3(37) (2014)
15. The IPFS network. https://ipfs.io. Accessed Apr 2020
16. DNS txt record. https://tools.ietf.org/html/rfc6763. Accessed Apr 2020
17. The client application of the ipfs.io network. https://github.com/ipfs-shipyard/ipfs-desktop. Accessed Apr 2020