Towards Complete Decentralised Verification of Data with Confidentiality: Different ways to connect Solid Pods and Blockchain

Towards Complete Decentralised Verification of Data with Confidentiality: Different ways to connect Solid Pods and Blockchain

Manoharan Ramachandran, Niaz Chowdhury, Allan Third, John Domingue, Kevin Quick, and Michelle Bachler
Knowledge Media Institute, The Open University
Milton Keynes, United Kingdom
{manoharan.ramachandran,niaz.chowdhury,allan.third,john.domingue,kevin.quick,michelle.bachler}@open.ac.uk

ABSTRACT

Over-centralisation of data leads to tampering and sharing user information without the consent of the owners. This problem has been studied extensively in recent times providing separate solutions involving distributed storage, Blockchain technology and Solid Pods. Individually these solutions are not sufficient to build realistic applications in a decentralised environment; however, a combination of them can effectively provide more powerful and useful use-cases. In this paper, we propose the methods of combining Solid Pods and distributed ledgers in introducing complete decentralisation of data with total user-control, keeping the integrity of the stored information intact through Blockchain-based verification. We demonstrated multiple configurations of our solutions, offering several new use-cases in various sectors. These configurations introduce new dimensions on the Web and mobile applications' data storage that developers can benefit from building Distributed Applications (DApps) in a complete decentralised environment.

CCS CONCEPTS

• **Information systems → Distributed storage**; • **Security and privacy → Information accountability and usage control**; • **Social and professional topics → Privacy policies**.

KEYWORDS

Blockchain, Linked Data, decentralisation, data verification

ACM Reference Format:
Manoharan Ramachandran, Niaz Chowdhury, Allan Third, John Domingue, Kevin Quick, and Michelle Bachler. 2020. Towards Complete Decentralised Verification of Data with Confidentiality: Different ways to connect Solid Pods and Blockchain. In *Companion Proceedings of the Web Conference 2020 (WWW '20 Companion), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3366424.3385759

1 INTRODUCTION

Over-centralisation has been a subject of mounting concern as social awareness surrounding how users control their data continues to grow. The centralised approach can cause alteration of data for numerous reasons including updates by the authors, corruption and most importantly deliberate manipulation by the controlling administrators leading to tempering or removal of data without the owner's consent or knowledge. Confidentiality could be another solicitude as data can be viewed, shared or sold by the possessors. These potential risks make users feel the need for more authority on their data more than ever triggering interest in decentralisation.

While both centralised and decentralised approaches have their drawbacks and advantages, decentralisation can potentially provide substantial benefits in areas such as storage of financial, medical, scientific, personal and sensitive data where data integrity and accessibility is of paramount importance. There has been various decentralisation approaches tried and tested in the literature. These include the use of peer-to-peer decentralised data storage, storing data on distributed ledgers and a combination of both where the former holds the data and the latter contains a hash of it ensuring the integrity. These approaches have their shortcomings preventing them from becoming mainstream solutions. Amongst, the most prominent weakness is for the data to disappear if the original host no longer remains online with no cached copies being available on the network; hence cannot act as practical solutions on their own and requires further improvement.

In addressing this problem, we identified that Solid could act as a valuable tool to improve the existing approaches. Initiated by Sir Tim Berners-Lee from Massachusetts Institute of Technology (MIT), Solid aims to decentralise the Web by transferring control of data from a central authority to users. In doing so, it allows users to retain complete ownership of their data.

In this paper, we propose an approach which combines two technologies: Solid Pods and distributed ledgers to facilitate the complete decentralisation of data. Our methods give users total control over their data while maintaining the integrity of the stored information through Blockchain-based verification. We have developed multiple configurations of our solutions, offering several new use-cases in various sectors. They are, Configuration 1: pod-stored Files as Hashes, Configuration 2: pod-stored Files as Smart Contracts and Configuration 3: pod-stored files as Blockchain key store. These configurations show new dimensions in the Web and mobile applications' data storage that developers can benefit from while building Distributed Applications (DApps) in a complete decentralised environment.

WWW '20 Companion, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7024-0/20/04.
https://doi.org/10.1145/3366424.3385759

2 LITERATURE REVIEW

The practice of over-centralising data has much controversy in recent times. The Cambridge Analytica scandal exposed how readily data are available to manipulate users in fulfilling one's agenda [1]. This incident also raised fingers towards the exercise of holding of personal data by single entities; for instance, in this case, Facebook. The issue of over-centralisation also linked to confidentiality as centrally-held data can be used, sold or manipulated without proper or no consent of its owners. The privacy issue also comprises usage control beyond single transactions and transparency requirements [2]. The aftermath of such events does not stay contained in the digital domain; rather, their impacts go beyond this realm. For example, incidents surrounding the loss of centralised records and distrust of non-centrally-held records led to the deportation of as many as 63 British residences in recent time with around further fifty thousand being at risk of losing their UK residency rights [3]. It is, therefore, no exaggeration to say that the problem needs immediate addressing and a push forward to finding breakthroughs to reverse the practice, a move towards complete decentralisation.

2.1 Linked Data

Linked Data (LD) is a form of structured data interlinked with other data to become useful through semantic queries of associative and contextual nature. It extends the capability of web data originally meant for only human readers to share information in a way that can be read automatically by machines [4]. LD plays a vital role in integrating data in the presence of multiple data sources making them interoperable.

Sir Berners-Lee, in his note Linked Data, coined the term and outlined four principles that he referred to as four rules for LD. These are as follows: Linked Data, i) Uses URIs as names of things, ii) Uses HTTP URIs to look up those names, iii) At the time of looking up a URI, provides useful information using the standards such as Resource Description Framework (RDF) and SPARQL, and iv) Includes links to other URIs to discover more things [5].

2.2 Distributed Ledger

Distributed Ledger is a record of decentralised entries with no central registry. Although not all distributed ledgers are Blockchains, the terms are considered synonymous. A Blockchain is a linked list of blocks that contains ledger entries more commonly known as transactions. A copy of the Blockchain is held by every participating node in a peer-to-peer network. The first block of the chain is called genesis block with subsequent blocks added through a process of consensus between nodes. Various consensus methods, such as proof of work, proof of stake, proof of authority etc. are used in different protocols that allow nodes to compete for a pole position enabling them to insert the new block. The design of a Blockchain ensures that once entered, blocks' contents cannot be changed even by the authors as long as entities control more than 50% of the nodes. This property of Blockchain makes the entries of a distributed ledger trustworthy [6, 7].

The progress in the development of distributed ledger has taken the technology beyond the storage of records and includes distributed computing in the form of smart contracts. These are blocks of executable source code stored on a Blockchain with a published interface describing the methods and their parameters. The code gets executed when the corresponding transaction is added on the distributed ledger. Because the code fulfils the same requirements of the immutability of Blockchain data, smart contracts form trustworthy distributed computation [7].

2.3 Distributed Storage

Distributed Storage is a decentralised approach of storing data in one or multiple servers that act as a filesystem for Linked Data. Interplanetary Filesystem (IPFS), Swarm, and FileCoin are some of the instances of distributed file storage [8].

2.4 Solid

Sir Berners-Lee originally viewed the World Wide Web as decentralised network. It was close to a peer-to-peer network assuming each user of the Web would be an active editor and contributor, creating and linking content to form an interconnected web of links [12]. The Internet, however, gradually turns out to be the opposite - an ideal example of the centralised paradigm. Prof Berners-Lee's response to this evolution of the World Wide Web is Solid. Solid, derived from **S**Ocial **LI**nked **D**ata, is a set of rules and tools for developing decentralised social applications based on Linked Data. It uses as much as possible the existing W3C standards and protocols [13].

Several technical challenges need overcoming to accomplish decentralisation of the Web. One approach is, instead of modifying the centralised client-server paradigm, improving peer-to-peer networking in a manner that adds more control and performance features than its traditional concept such as BitTorrent. Solid is a project that aims to achieve that goal. Its central focus is to enable the discovery and sharing of information in a way that preserves confidentiality. It allows users to store personal data in Pods (Personal online data stores) hosted at the location of users' desire. This also have the flexibility to distribute data among several pods; allowing them to organise various types of data (personal, contact, health, financial) in multiple pods with varying degree of access control. In a nutshell, Solid allows users to retain complete ownership of their data, including where to store the data and who has permission to access it [14].

3 ISSUES IN DATA VERIFICATION

Previous attempts of decentralisation include the use of distributed storage, distributed ledger and a combination of both. For example, data can be distributed across multiple servers by duplication with anyone wishing to use the desired copy must know its precise location. This approach, however, fails to ensure the integrity of the data as there remains no straightforward way to identify if the data is altered. An improved method could be making the distributed storage to act as a filesystem for storing data with clients keeping copies of hashes of all files locally. Clients can then run the queries with these hashes to retrieve the data (e.g. IPFS). This technique helps to verify the integrity of the data because if the stored data gets altered, there will be a mismatch between the locally stored hash and the hash of the data, tearing apart the connection. In such cases, clients' query does not return the altered data, and in the event of no results, we can assume that either the data got altered or went missing [9].

Distributed ledgers can also serve the act of decentralisation in various ways. Ideally, for a fully decentralised solution, both data and querying can be performed on a distributed ledger. It requires a smart contract datastore and a smart contract index and query engine. This method provides a fully distributed storage and a firm guarantee of data integrity. The tradeoff is, however, the cost as it requires payment for every execution as well as the initial cost of populating the smart contract datastore. The use of a distributed ledger with distributed storage can help to make the solution cost-effective. This approach enables clients to avoid the need for maintaining the hashes locally; instead, data goes to a distributed storage while hashes and their associated timestamps stay as a trustworthy record on the distributed ledger.

The above solutions, however, have problems. Despite the ongoing research of improving the file availability in distributed storage systems, the risk of having data disappear cannot be ruled out [10]. If the original host no longer remains online with no cached copies being available on the network, despite having hashes on the distributed ledger, users can no longer retrieve their corresponding data. Therefore, while storing data directly on the distributed ledger is expensive, storing them on a distributed storage creates the risk of losing them. As a result, these existing approaches do not seem to be practical solutions on their own to the problem of decentralised storage with verification and require further improvement where Solid can play a pivotal role [11].

4 PROPOSED APPROACH

To provide a complete decentralised verification of data with confidentiality, we arranged Solid Pods and Blockchain connections in three major configurations with further variations in each of them. These configurations do not consider any particular context. Users will have to decide what to use depending on the type, volume and frequency of their data. Nevertheless, we show some use cases where configurations can be adopted to help users make their decisions later in Section 5.

4.1 Pod-stored Files as Hashes

A cryptographic hash function is a mathematical algorithm that maps data of arbitrary size to a fixed size string. It is a one-way function, i.e. practically impossible to invert [15]. Hashes are used to convert any form of data (text, image, transaction etc.) into a fixed-length of string, in which a simple modification of a single character/bit changes the fixed-length string. As storing data directly on Blockchain is not an efficient method, hashes of files are stored on the distributed ledger to prove the authenticity. This section explores how files stored in Solid pods can be hashed and put on the Blockchain.

4.1.1 Configuration 1.a: Figure 1 shows the first configuration where data stored in the Solid pod are individually hashed before storing on the Blockchain. This will facilitate tamper-proof verification with confidentiality to all pod-stored file. Equations 1 – 3 presents the mathematical representation of the configuration 1.a where Equation 1 and 2 represent hashing, while Equation 3 represents verification. In Equation 1, bh_{1-n} represents the hash of files stored in the pod ranging from 1 to n that are then put on the Blockchain. Here, n is the total number of files and al represents the hashing algorithm used to hash the files from f_1 to f_n. In Equation 2, ph_{1-n} represents the hash of files stored on the Solid pod that needs verification where al represents the hashing algorithm and $((f_1)|...|(f_n))$ represents the pod-stored files from 1 to n. v_{1-n} in Equation 3 is a boolean variable representing the verification of the pod-stored files in which the hashes of the pod-stored files (ph_{1-n}) are compared with the Blockchain-stored hashes (bh_{1-n}).

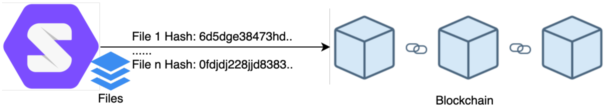

Figure 1: Storing individual hashes of Solid pod-stored files on the Blockchain

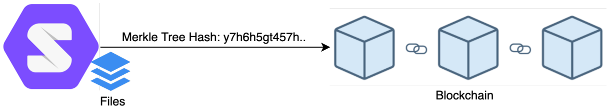

Figure 2: Storing the Merkle Tree hash of Solid pod-stored files on the Blockchain

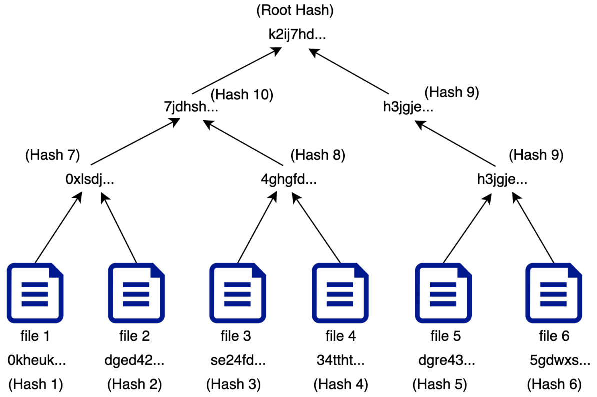

Figure 3: Example of a Merkle Hashing

$$bh_{1-n} = al(f_1)|...|al(f_n) \tag{1}$$

$$ph_{1-n} = al(f_1)|...|al(f_n) \tag{2}$$

$$v_{1-n} = comp_{1-n}(ph_{1-n}, bh_{1-n}) \tag{3}$$

This configuration is suitable when there is a need for data confidentiality with decentralised verification. For example, if a patient aspires to keep personal medical records secured and decides to share data with a particular doctor, the above configuration can be used where doctors can verify the integrity of the data in the presence of the Blockchain. Many other related use cases can also be created for different areas having similar scenarios.

Configuration 1.b: Figure 2 shows the next configuration in which the set of files in the Solid pod are hashed in a Merkle tree, and the root hash of the tree is then stored on the Blockchain. A Merkle tree is a tree of hashes. It is developed by hashing pairs of nodes repeatedly until there is only one hash left [7]. This approach provides tamper-proof verification to each pod-stored file where a small change invalidates the entire tree. The Merkle tree also benefits from storing a single hash on the Blockchain, making it cost-effective compared to the previous configuration.

$$bm = merkle(ph_{1-n}) \tag{4}$$

$$pm = merkle(ph_{1-n}) \tag{5}$$

$$vm = comp(pm, bm) \tag{6}$$

Equation 4 and 5 represent the Merkle hashing, while Equation 6 represents the verification of this configuration. In Equation 4, bm denotes the Blockchain-stored Merkle root hash of the pod-stored files. Equation 2 (ph_{1-n}) is then utilised in 4 to initially get the list of hashes of the pod-stored files followed by Merkle hashing using the $merkle()$ function. Equation 5, on the other hand, represents the hashing of pod-stored files in a Merkle tree form where pm denotes the root hash of the Merkle tree of pod-stored files that needs to be verified in which Equation 2 is utilised to generate the initial list of pod-stored file hashes.

Figure 3 shows an example for Merkle hasing using a set of 6 files (file 1 to 6) stored in the Solid pod. A hash of the files 1 and 2 are hashed together to form the hash number 7. Similarly, hash number 3 and 4 are used to form hash 8 and hash number 5 and 6 are used to from the hash 9. Now, hash number 7 and 8 are used to from the hash number 10. Finally, hash number 10 and 9 are used to form the root hash. A small change in any file completely changes the root hash there by invalidating the entire tree. By recording the hash numbers, it is possible to identify which file has been tampered with.

This configuration has use cases similar to the previous configuration but it can also be used when there is a need for verifying a set of files together. For example, a company with millions of files of their financial data can use this configuration to effectively verify all the files' authenticity without the need for any third party involvement.

4.2 Pod-stored Files as Smart Contracts

In the previous section, the verification of pod-stored files by storing the hashes of the files on the Blockchain was explored. In this section, storing the pod data directly on a Blockchain in the form of smart contracts and using solid pods as a storage space to support the Blockchain-related activities will be discussed.

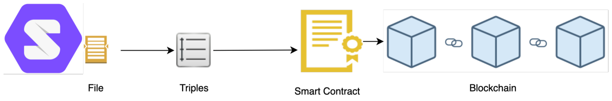

Figure 4: Storing the triples of a pod-stored file on Blockchain in the form of a smart contract

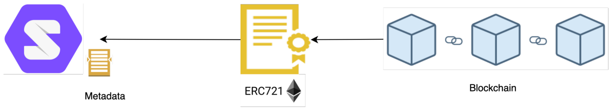

Figure 5: Storing the metadata of an ERC721 token smart contract in Solid pod

4.2.1 Configuration 2.a: Figure 4 shows the first configuration of this section, where a file from a Solid pod is converted into triples and stored in the form of a smart contract on the Blockchain. Triples are a group of three entities that codifies semantic data statement in the form of subject-predicate-object expression. This format enables any data to be represented in a human and machine-readable way. The major advantage of storing the data in this format is that every triple can be verified individually and also can be understood by the computers adequately to perform any search on it. Possible use cases for this configuration is when a user wants to store structured data on their Solid pod and wants to provide decentralised verification with semantic search functionality. This configuration, however, may not be suitable for large or personal data due to it being stored directly on the Blockchain in triples format in the form of smart contracts. Executing a substantial amount of data on a Blockchain is very slow while storing personal data could infringe the General Data Protection Regulation (GDPR).

4.2.2 Configuration 2.b: This configuration suits a particular type of smart contract called ERC721, which represents an asset or utility and can be traded on Ethereum Blockchain. They are also called token contracts standardised using the Ethereum Request for Comments (ERC) standard. ERC721 is the most common token standard that used to represent non-fungible assets which became quite famous because of crypto kitties Blockchain game [16]. When creating an ERC721 smart contract, the metadata for the contract has to be defined somewhere, and the URL for that metadata should be included in the smart contract. Each time when an ERC721 contract is read, the contract extracts its metadata using the URL provided. Using this configuration, complete confidentiality over the metadata can be provided to the user. Figure 5 shows the configuration in which ERC721 token metadata is stored in the Solid pod. As the metadata is stored in the Solid pod, a user can restrict who can view their ERC721 token and who cannot.

4.3 Pod-stored Files as Blockchain Key Store (Software Wallet)

The standard way that users access data on a Blockchain is through a wallet. For example, bitcoin owners will access their bitcoin tokens through a bitcoin wallet. In this section, we explore how Solid pods can be adapted to behave like wallets.

4.3.1 Configuration 3: Figure 6 shows a configuration utilising a Solid pod-based software wallet for accessing data on a Blockchain. In this configuration, the wallet stores the key pair (public key and private key) in a Solid pod and depending on the types of Blockchain, its API gains access to login to the distributed ledger using the credentials stored in the Solid pod. An ideal use case of this configuration is the Blockchain wallet that can be embedded with the Solid browser making users utilise their Solid login to send data to anyone. This approach also enables decentralised applications (DApps) to be accessed via a Solid pod.

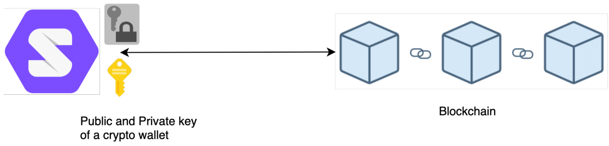

Figure 6: Use of Solid Pod as a Software Wallet for Crypto

5 EVALUATION AND DISCUSSIONS

This paper reports an ongoing work. We are currently in the process of implementing and testing the configurations presented in Section 4, but so far only developed two of them, 1.a and 2.b, in fitness and education domains. The rest of the configurations will be developed and tested in future.

5.1 Fitness Domain

Solid and Blockchain-based Mobile Application: Wearable fitness devices collect various data about users around the clock. Companies that manufacture these devices own the data collected and potentially could use it for profiling and marketing. As the use of wearables has been rapidly increasing, a significant number of people will be using them in their regular activities in future. It potentially puts them in a situation where providers can use their data without their knowledge [17]. To avoid this, we have come up with a mobile application that uses our configuration (1.a) and lets a user store their fitness information from the fitness device directly in a Solid pod. Also, we have used fitness ontology to convert the fitness data into Resource Description Format (RDF) format so that both machines and humans can read it. It works as follows: The fitness source file and the converted file are both hashed, and the hash placed on a Blockchain for verification purposes. We used an express NodeJs service to transfer data between the mobile phone and the Solid pod due to certain limitations of Solid in supporting communications between Solid pods and native mobile applications. We expect developers of Solid to overcome these limitations in future when the true potentials of our proposal can flourish.

5.2 Education Domain

Solid and Blockchain-based Credential Verification: There is an increasing number of people falsifying their academic credentials and job recruiters are spending their resources to authenticate the validity of educational degrees. The United Kingdom (UK) Department for Education reported that there are a growing number of instances of misrepresentation and forgery in the presentation of academic credentials [18]. A recent analysis of 5,500 CVs by the Risk Advisory Group found that 44% of CVs had discrepancies in their education claims with 10% of those having false grades [18]. This is a growing issue and needs addressing urgently. To solve this problem, we proposed and implemented a system called LinkChains that provides Blockchain-based verification for academic credentials while storing the credential and supporting data in Solid pod [8, 11]. We have used the configuration 1.a and 2.b in this use case. It works as follows: When an academic credential is issued on LinkChains platform, it hashes the credential followed by storing that hash on a Blockchain. At the same time, it also uploads the academic credential to Solid pod of the student. In parallel, the platform provides an ERC721 token to the student with storing the metadata in decentralised storage and a Solid pod. Our approach significantly decreases the verification time for the employers to verify students' credentials while the authenticity of the credential remains protected.

6 CONCLUSION

This paper explored the idea of trustless verification with confidentiality showing how a combination of Solid and Blockchain can support the verification of data without the loss of users' control in a decentralised environment. The paper also demonstrated the techniques of creating useful use cases in a wide variety of domains. If there is data involved in any field which requires confidentiality and trustless verification, these proposed configurations could be of good use. The paper reports about a work in progress and the authors are in the process of extending it into a full-length journal paper with more use cases, domains, and evaluation of cost involves in verification soon.

REFERENCES

[1] Ingram, D., Factbox: Who is Cambridge Analytica and what did it do?, Reuters, March 2018.
[2] Bonatti, P. A., Kirrane, S., Polleres, A., Wenning, R., Transparent Personal Data Processing: The Road Ahead, in proceedings of the International Conference on Computer Safety, Reliability, and Security, pp 337–349, Trento, Italy, September 2017.
[3] Domingue, J., Third, A., and Ramachandran, M., The FAIR TRADE Framework for Assessing Decentralised Data Solutions, in proceedings of World Wide Web Conference, San Francisco, USA, May 2019.
[4] Wood, D., Zaidman, M., Ruth, L., Hausenblas, M., Linked Data, Manning Publications, ed. 1, 2014.
[5] Berners-Lee, T., Linked Data: Design Issues, w3.org, 2006, URL: www.w3.org/DesignIssues/LinkedData.html
[6] Franco, P., Understanding Bitcoin, ed 1, Wiley, 2015
[7] Chowdhury, N., "Inside Blockchain, Bitcoin, and Cryptocurrencies", ed 1, Taylor and Francis, 2019
[8] Third, A. and Domingue, J., LinkChains: Exploring the space of decentralised trustworthy Linked Data, in proceedings of Decentralising the Semantic Web, Jul 2017, Vienna, Austria.
[9] Third, A. and Domingue, J., Linked Data Indexing of Distributed Ledgers. in proceedings of the 1st International Workshop on Linked Data and Distributed Ledgers, Perth, Australia, April 2017.
[10] Huang, H., Lin, J., Zheng, B., Zheng, Z., and Bian, J. When Blockchain Meets Distributed File Systems: An Overview, Challenges, and Open Issues, IEEE Access, 2020
[11] Third, A. and Domingue, J., LinkChains: Trusted Personal Linked Data, in proceedings of BlockSW, Auckland, New Zealand, October 2019.
[12] Berners-Lee, T., Weaving the Web: The Past, Present and Future of the World Wide Web by its Inventor, HarperCollins, 1999.
[13] Sambra, A., Mansour, E., Hawke, S., Zereba, M., Greco, N., Ghanem, A., Zagidulin, D., Aboulnaga, A., and Berners-Lee, T., Solid: A Platform for Decentralized Social Applications Based on Linked Data, Technical Report, MIT CSAIL & Qatar Computing Research Institute, 2016.
[14] Mansour, E., Sambra, A., Hawke, S., Zereba, M., Capadisli, S.„ Ghanem, A., and Berners-Lee, T. A Demonstration of the Solid Platform for Social Web Applications, in proceedings of the 25th International Conference Companion on World Wide Web, Montreal, Canada, April 2016.
[15] Tilborg, H., Fundamentals of Cryptology: A Professional Reference and Interactive Tutorial, ed 1, Kluwer Academic Publishing, 2000.
[16] EIP-20, https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20.md
[17] Chowdhury, M. J. M., Ferdous, M. S., Biswas, K, Chowdhury, N., Kayes, A., Watters, P, and Ng, A., Trust Modeling for Blockchain-based Wearable Data Market, in proceedings of the IEEE CloudCom, Dec 2019, Sydney, Australia
[18] DoE Advice and guidance on degree fraud, Department for Education, 2017.