Transforming Higher Education Systems Architectures Through Adoption of Secure Overlay Blockchain Technologies

Foysal Miah[1], Samuel Onalo[1], and Eckhard Pfluegel[1]

Kingston University London, Kingston upon Thames, KT1 2EE, UK
foysal@kingston.ac.uk, k1450301@kingston.ac.uk, e.pfluegel@kingston.ac.uk

Abstract. The adoption of Distributed Ledger Technology (DLT) has been growing tremendously in recent years following the introduction of Bitcoin in 2009. However, the usefulness of DLT is not limited to the financial sector, and this paper investigates the viability of DLT architectures for use in Higher Education (HE). This sector faces challenging financial constraints, and one way to address this problem is to adopt emerging DLT technologies as architectures for HE systems. This article presents the ASTER Open Source system, a hybrid DLT integration within the context of a student submission system for assignment grading purposes. ASTER addresses many concerns of traditional system architectures such as centralisation, system downtime, and decoupling; all of which are mitigated through the use of blockchain technology. The advantages and drawbacks of such a new approach are discussed, including the aspect of security concerns relating to student work being submitted to a public ledger.

Keywords: Blockchain Security, Security Overlays, Decentralised Ledger Technologies, Higher Education Systems

1 Introduction

From both an institutional and student perspective, traditional learning management systems rely heavily on centralised infrastructure, and even though these solutions function well, there are limitations that require re-evaluation. Whilst the focus here is on HE systems, the above postulation can be applied to most systems currently in operation.

The first obstacle for an institution is cost. Universities in the UK and elsewhere are under extreme pressure to reduce costs with drastic measures being taken to ensure continued operation across the board. As mentioned previously, the current solutions function well, however, the cost of maintaining a centralised system is high, typically requiring dedicated staff to manage system issues and relentlessly update the software to keep in line with changing external factors. While the recent uptake of cloud infrastructure technology, presenting features such as Software as a Service (SaaS) and Platform as a Service (PaaS), have dramatically reduced overheads, this approach introduces a different set of problems.

Second, there are the concerns of the student to address. Submitting assignments to a centralised system means the student is reliant on the software vendor to keep their submission safe, not to mention their personal data, which is particularly important with the recent data protection changes decreed by the European Commission. Institutions and vendors must adhere to GDPR guidelines [5] to ensure personal data is kept secure or they could face severe fines. Submissions to a centralised system are prone to intermittent outages, especially during critical submission times, mainly due to insufficient infrastructure resourcing. The obvious solution would be to increase resource availability during peak times. However, the increased resources would remain idle during off-peak periods, the cost of which would need to be considered by the institution and by extension, passed to the student. Finally, insufficient security measures are also a problem with personal data being the primary target for cyber criminals. Even though the strict regulations imposed by GDPR legislation have obligated vendors to improve their security protocols, such measures incur costs that are passed down to the student to bear.

With the introduction of a decentralised submission system, the above issues would no longer be relevant. Such a proposed system would be released as Open Source software maintained by a community of developers; the infrastructure would be formed as a peer-to-peer network with the students, institutions, and the public running client software as processing nodes. This architecture would require little to no staff to maintain. A further potential benefit could be an additional revenue stream for universities, by selling off the currency that is generated for submission processing. In times of a pandemic, austerity, and an uncertain financial climate, this type of technology could potentially help bring running costs down dramatically while maintaining infrastructure integrity. Research regarding currently active blockchain networks has found no attempt so far in developing a blockchain that is explicitly targeting the HE sector. There is an existing blockchain that deals with the lengthy time it takes to publish a research paper to related journals [7], but this does not deal with assignment submissions by students.

The main contribution of this paper lies in secure blockchain technology. We present the design and development of a secure student submission system with a novel blockchain architecture based on security overlays. This has led to the creation of an Open Source prototype solution named ASTER [4], providing an end-to-end decentralised and secure system, mitigating the potential security risk of using a public, insecure blockchain as far as the confidentiality of blockchain data is concerned.

The paper is structured as follows: in Section 2, we review traditional systems and architectures including pertinent aspects of Blockchain technology. In the next section, our system design and implementation are present. This is followed by a description of the security in Section 4. Section 5 concludes the paper.

2 HE Systems Architectures

In this section, we review salient aspects of Higher Education system architectures and explain the advantages of DLT versus a centralised approach.

2.1 Removing Central Points of Failure

Time after time, there are reports identifying corporations that have had their systems breached in one way or another. The global governments then put legislation in place to ensure these breaches do not become commonplace. However, when a breach is found, and a company is fined, it is not the company that suffers in the long run. Fines, especially against the largest companies, can be recouped simply by raising the prices of their products affecting the consumer. Not to mention the costs involved in ensuring security is kept up to date, which are invariably passed on to the consumer. These attack vectors exist because of one fundamental architectural design flaw – the centralised nature of traditional systems.

Within the HE context, our ASTER system aims to remove all central points of failure, this is achieved by the very nature of DLT design, by replicating the network across a global cluster of nodes. By doing so the network is protected from security issues such as DDoS attacks thus eliminating availability issues plaguing traditional architectures.

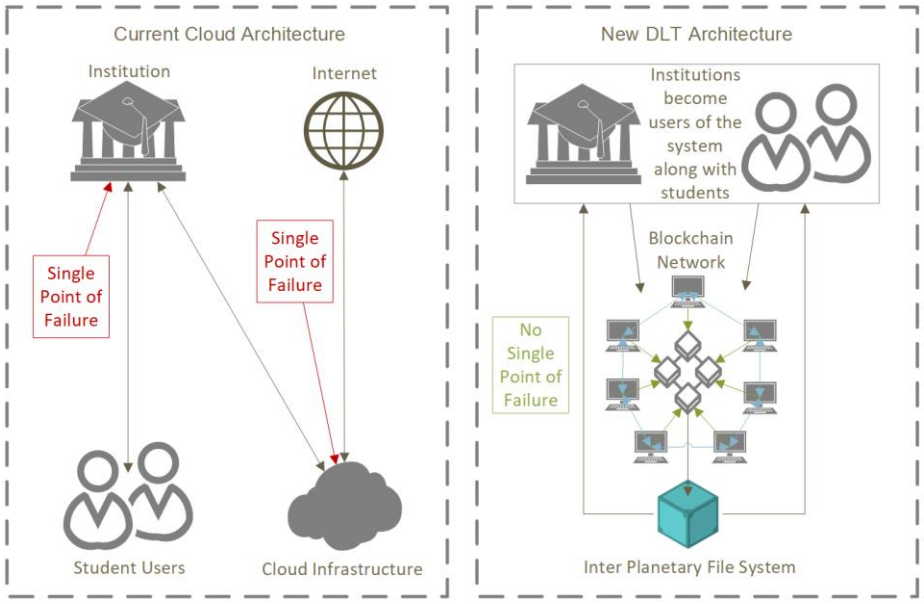

Figure 1 High Level Architecture Comparison.

Figure 1 shows a comparison between a traditional cloud system and a potential architecture utilising DLT.

DLT addresses these security concerns by design as there is no single point of failure to exploit. An attacker would need to target every single node on the network simultaneously to have any effect; any such attempt would be extremely cost-prohibitive.

2.2 Infrastructure and Software Architecture

Since the beginning of computing, traditional systems have been designed around the idea of centralisation, housing data and applications on a network server onsite. As time has gone on, methods such as multi-point failover processes have been introduced to mitigate data loss and cloud solutions have been adopted primarily to allow for service continuation and cost reduction.

The HLD below depicts a possible scenario that could be adopted in the form of a cloud-hybrid solution. The diagram depicts the application layer between the users and the LMS, the cloud platform, which would host the blockchain and the communication between the IPFS storage layer.

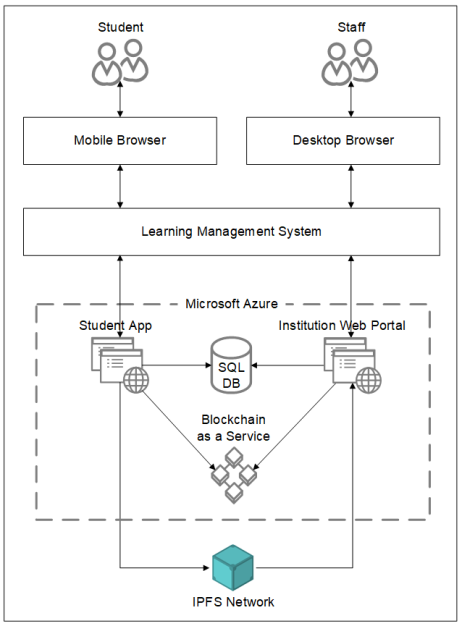

Figure 2 High Level Diagram.

2.3 Addressing System Downtime

Any system analyst maintaining an enterprise system will confirm that any form of system downtime creates added pressure to their workload. Even scheduled downtime is always a cause for concern. There is some semblance of control if the entire architecture is on-prem, but this is becoming more of a distant memory, especially when considering cloud solutions, where the baton of ownership is being passed on to a 3rd party with their supposedly iron-clad promise of adherence to accompanying SLAs.

No matter the operational process in place to address downtime, there is one inevitability. Traditional and current systems are always going to be prone to some level of system downtime which is why no service provider guarantees 100% uptime. A simple online search will point out services offering 99.x% uptime, and it has become commonplace for a service to be measured for reliability with the number of successive 9s. But this is where DLT disrupts the status quo – by offering 100% uptime. One might argue that performance may be an issue, and it may very well be, but the system would never suffer downtime – ever. The only time the system could be down is if every node stopped using the service.

3 System Design and Implementation

In this section, the design and implementation of our system will be presented.

3.1 General Architectural Considerations

Current architecture methods used are very archaic, even when considering using infrastructure in the cloud. These methods are still modelled around the idea of centralisation. A typical architecture might involve an application layer, middleware, services, data layer and a platform layer. Each of which will typically sit on a server making up the full stack. In a more modern design, these layers may be separated to ensure reliability and maintainability, but still, sit on servers. When it comes to cloud service design, all that is happening is that these servers are no longer owned and maintained by individual companies, but leased out by cloud service providers.

Currently, the vast majority of universities may maintain their infrastructure and equipment, or lease services from cloud service providers. Figure 3 shows how a typical architecture might look against the proposed architecture.

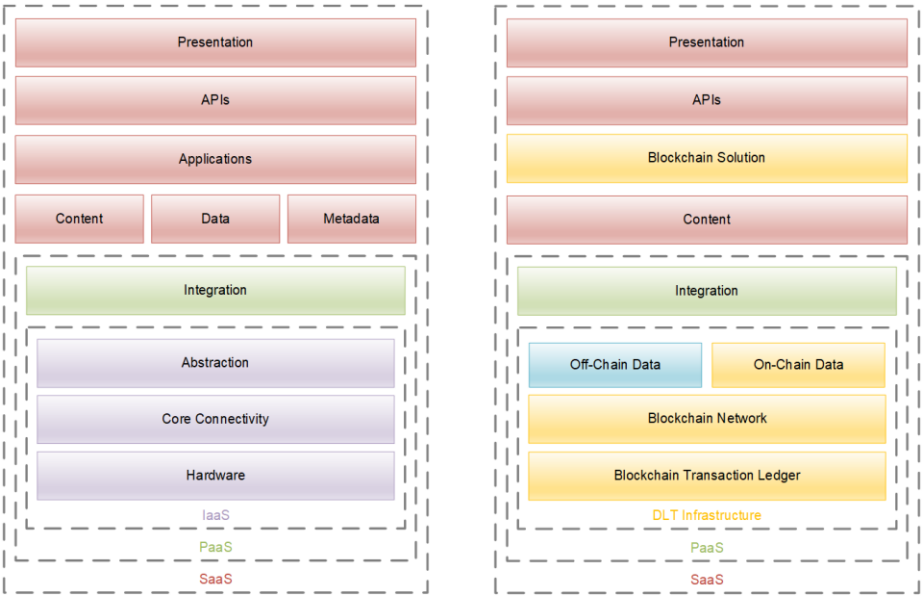

Figure 3 Architectural Comparison.

Our proposed architecture would replace the Infrastructure as a Service, with a DLT infrastructure, containing Off-Chain Data (in the case of the ASTER system, this would represent IPFS), On-Chain Data (the IPFS reference data on the EVM), the Blockchain Network (client nodes confirming the transactions taking place) and the Blockchain Transaction Ledger (the confirmed transactions). Stripping away the old infrastructure layer would mean universities would be saving on costly equipment and leasing costs as well as expensive on-going maintenance costs.

3.2 Potential Architecture Types

Several types of architecture can be designed to implement ASTER. There are many frameworks available to assist with this, and the difficulty here is choosing the correct tools for a useful implementation. This section will illustrate three methods of potential architecture designs: *Ethereum-Based*, *Independent Blockchain*, and finally, *Hybrid Solution*.

Ethereum Based Architecture An Ethereum based approach would entail storing files on the Ethereum blockchain, this can be achieved by creating an Ethereum contract between the university and the student, which would allow students to submit their work directly to the Ethereum blockchain. Once the work is submitted, the student will have proof the submission has taken place, as this will be reflected in their Ethereum wallet and the ethscan [3] explorer, where all transactions are logged.

This approach has numerous advantages. It consists of a straightforward method to store student submission and would bring an effortless setup only requiring the creation of an Ethereum contract. This solution seems an obvious choice until the costs are explored. Due to the monetisation of Ethereum, mitigate this cost for a university may prove challenging and may not be ethically practical. Also, the resulting system will still be reliant on legacy architecture where there is a potential point of failure. Furthermore, a significant disadvantage is the lack of data confidentiality. Entire student submissions would be stored openly on the public transaction database, with obvious potential detrimental consequences to students and lecturers alike.

Independent Blockchain The development of an independent, purpose-built blockchain would allow for a currency that is not monetised outside of its intended environment but used for assignment submission and content creation. The blockchain would require students and possibly the university to mine currency by processing the submissions and content to the blockchain. This mined currency would then be used for submission fees. Any additional currency accumulated could then be used to purchase other services from an institution, such as graduation tickets, merchandise, and other university products or services.

Reviewing the advantages of this architecture, it can be stated that while there is no real monetary value associated with the cryptocurrency, students may be able to use the currency to exchange for other university items. Furthermore, there would be little to no maintenance costs arising. On the other hand, this would require the development of a complete blockchain network, which is a serious challenge. The system will still be reliant on legacy architecture where there is a potential point of failure.

Hybrid Solution With a hybrid system, it is possible to use two separate systems to achieve the end goal, such as using IPFS and Ethereum. IPFS is a distributed storage system which allows any user to store any type and size of data [6]. Currently, there is no native feature within IPFS that establishes who or when a file has been submitted to its network, which means using the platform on its own is not viable, however, by decoupling the data from the user submission details, it is possible to design a system which stores the submissions on the IPFS network and the submission details on the Ethereum network. Both of which would create an immutable record. As discussed above, storage on Ethereum is highly cost-prohibitive; however, this hybrid system would only be storing up to 1KB of data on the Ethereum network, which equates to 54 pence per KB per submission. With an average of 6 submissions per year, at the Ethereum price stated earlier, submissions would cost £9.72 over the duration of a student's 3-year undergraduate course. However, this indicative cost is subject to market price fluctuations. The hybrid solution exhibits several positive aspects. It is highly cost-effective, has little to no maintenance costs and can be implemented using rapid development, as the storage system is already established. The inconveniences of this solution are that transaction costs are subject to Ethereum market price

fluctuations and that the system will still be reliant on legacy architecture where there is a potential point of failure.

3.3 The ASTER System

The ASTER proof of concept system is based on the hybrid architecture described previously. ASTER utilises an Ethereum smart contract to store the IPFS address created on file submission. ASTER can be thought of as a hybrid dApp which will use Metamask to transact between EVM and IPFS. The architecture consists of a mobile client front end and a web portal for administrative tasks. A data controller handles the data processing between the client front end and data storage. Finally, the data storage layer utilises IPFS, which will store the student submissions ready for lecturers to mark.

The mobile front end has been created using Xamarin forms allowing for a single codebase to be shared across various mobile platforms. The data controller is implemented as a web service, with C# being the coding language. The web portal for lecturers is coded using ReactJS. The storage layer uses the IPFS network storing all submissions to a public network of active storage nodes, with transaction data being stored on the Ethereum network. In order to simulate a transaction being processed, the Ethereum contract will be created using Solidity and submitted to the Rinkeby test network.

Finally, there is a need for a database to allow for credentials to be managed. The database is created using the data first approach using Microsoft Entity Framework and deployed to the Microsoft Azure Cloud Platform. A data controller API angles the business logic between the database and client application. The Azure platform provides commercial cloud computing services across many data centres across the globe. To simulate a typical HE back-end, Microsoft Azure is configured to host the web application and the data API, as well as the database back-end.

4 Implementing Security

Security is a fundamental requirement for the proposed system and its application area. Student submissions need to be protected concerning integrity and confidentiality against both internal and external attackers. In this section, we commence by illustrating the existing security mechanisms of blockchains. The need for additional security in the form of data confidentiality will be highlighted, and a novel mechanism for providing this security requirement and its role within ASTER will be presented. This continues our research on security protocols [9] and security overlays [10].

4.1 Standard Blockchain Security Features

It is vital to understand that while cryptography is used in particular, specific areas of blockchain technology, it does not provide complete and comprehensive security. However, the security qualities provided rival many centralised systems by a substantial margin. The use of these features is quintessential to the successful implementation of ASTER.

Secure Hash Functions A fundamental operation of the blockchain system is the block hashing process; this process is responsible for verifying every new block added to the public ledger and uses cryptography to achieve verification. Various cryptographic methods are in use within different blockchain implementations, the most popular being Secure Hashing Functions. Rapid integrity verification is achieved through the use of sophisticated data structures.

Public-key Cryptography Asymmetric (Public)-key Cryptography was first suggested in 1976 by Whitfield Diffie and Martin Hellman [2], their idea was to introduce a public and private key pair and is the underlying principle of industry-standard encryption and digital signature algorithms such as RSA [11] or DSA [11] used today. Public-key Cryptography is used to tackle the following main security challenges within the blockchain process: to validate the authenticity of a transaction and to provide ownership anonymity.

On the initiation of a transaction, to ensure ownership of the data contained, a cryptographic signature must be passed along as part of the transaction. As the cryptographic signature is created using a private/public key combination unique to the owner, the blockchain network and the client nodes within the network can then confirm the origins of the transaction thus validating the transaction as authentic.

Since the majority of blockchain implementations are public, ownership anonymity becomes a high priority, as anyone can interrogate the blockchain ledger, and if the transaction data are not anonymous, the ownership is easily identifiable. The blockchain process handles this aspect by allowing the transaction originator and recipient to create a wallet address using asymmetric encryption.

4.2 The Need for Data Confidentiality

The use of encryption is not necessarily available when creating data stored in a blockchain. However, in many systems nowadays, this necessity arises, partly due to the exposure of online systems to attacks, partly due to more sensitive data and transactions present. Major blockchain providers such as Hyper Fabric Ledger [1] and Multichain [8] have responded to this need by releasing permissioned blockchains, where access control can be managed using a central entity. Encryption of data is provided as an additional feature, sometimes as a paid premium feature. This approach contradicts the original philosophy behind Blockchain systems such as Bitcoin, as it is deviating from the idea of decentralisation. It also requires time and

overhead for managing these permissions and might require the setting up of a Public Key Infrastructure.

4.3 Virtual Private Security Overlays

In our previous research [10], we have suggested an alternative security approach, based on security overlay architectures. The basic idea is to apply a suitable secure information dispersal scheme such as *secret sharing* [12] in order to diffuse sensitive data on several blockchains. This achieves transaction confidentiality as long as a threshold number of individual blockchains is not inspected simultaneously. In particular, if this idea is applied to public blockchains, the resulting architecture may be seen as a blockchain with additional security properties and is referred to as *Virtual Private Blockchain* (VPBC) in analogy to a Virtual Private Network in traditional network security. In this approach, confidential transaction content is replaced with "fake" pseudo-content the precise choice of which will strongly depend on the specific application scenario. The transaction recipient will be able to retrieve the original data by combining a set of fake transactions, using a suitable method. Depending on how the transaction data is structured and what the specific blockchain application prescribes in terms of security requirements, an additional out-of-band channel might be required. The main advantage of this approach is that it does not rely on encryption, as it implements confidentiality through covertness. In addition, it is very flexible and can be based on any number of individual blockchains and transactions.

4.4 ASTER Security Approach

The main difference of ASTER to the VPBC approach is the restriction to a single Private Blockchain, in this case, the Ethereum system. Data diffusion will be achieved through multiple transactions, and the arising need for a secure out-of-band channel is implemented based on email. The motivation behind this design decision is the fact that one of the earlier versions of ASTER was already implemented based on Ethereum; and that the existence of an email channel between students and lecturers is a realistic assumption.

A secret sharing scheme with parameters m and n is also called a (m,n) threshold scheme and it has the property that given data (the *secret* s) can be divided into n parts (the *shares*) in such a way that m shares are sufficient to reconstruct s.

Consider an intended transaction with sensitive transaction data d, requiring protection. This will be shared as n shares $d_1,...,d_n$ using fake transactions and an email message if required. This will be explained in the following example: assume the submission of a student assignment. The transmitted information is the student name, ID number and the actual assignment document and a reasonable decision would be to consider the ID number ID (for data privacy reasons) and assignment document A (in order to prevent cheating) confidential. Hence, the data d are the concatenated latter two pieces of information.

In order to create suitable fake student assignments, one can proceed as follows, where without loss of generality we will discuss the individual pieces separately:

denote ID_i the ith share of the ID number. Rather than including this share information in the fake assignment, we can send the values $ID_i + R_i$ and $R_1 \oplus R_2 \oplus ... \oplus R_n$ using the email channel where the R_i are random numbers. Including the fake assignment documents requires additional care, as typically shares in a secret sharing scheme appear as random values. Unless it would be argued that documents would be encoded a (potentially proprietary) binary encoding scheme, the following approach could create plaintext documents: slightly abusing notation, we will use the same R_i to denote a new set of numbers to be determined. The aim is to create a fake set of assignment documents B_i. If we consider the set of equations $A_i \oplus R_i = B_i$ ($i = 1,...,n$) we can solve for the R_i and proceed as in the case of the student ID numbers, using an email.

A mechanism to explain the resulting proliferation of assignments submissions needs to be in place. This could be achieved by simply having an artificially large number of students enrolled for the assignment. In case of suspicion raised, this could be explained as having distance learning students, students from the previous cohort retaking the assignment, and so on.

5 Conclusion

This paper has investigated Distributed Ledger Technology and how the concept could be applied to improve the current student submission system implementation that is in use. We have investigated the viability of DLT integration within the context of a student submission system for assignment grading purposes and defined a detailed design of a prototype and chosen specific technologies to integrate with the ASTER prototype system. This prototype system has been developed which showcases the use of two unrelated blockchain technologies to submit and store student assignment submissions, with a legacy backend configured on the Azure platform simulating student data that would be in use by a university. An innovative mechanism to establish data confidentiality, an aspect often neglected in current blockchain technology, has been designed.

The ASTER system currently has limitations and will require additional work to become a fully functioning and production-ready application. The prototype was aiming to demonstrate the ability to produce an application that would connect to an already existing system and whilst the API is able to generate lists of assignments, students, courses, modules and lecturers, it is not possible to create lists assigned to particular users, but this can be achieved by revisiting the LINQ code.

Due to timing constraints, the Xamarin forms application could not be built and would have been an added benefit for those wishing to use ASTER on a mobile platform, however, the ASTER front-end client can be used on a mobile device as it is responsive. However, the main research question has been proven, which was to create a hybrid application that will allow student assignments to be stored on a decentralised system and also making use of distributed ledger technology. The benefit of using such emerging technologies is also highlighted successfully in the prototype, particularly where the cost of submitting a document or collection of files of any size is a fraction of a penny.

ASTER is potentially looking at re-defining how the IT infrastructure within HE currently operates, a move like this would usually require a cultural change across the institution, however it may be possible to cushion the change impact by introducing the architecture gradually. Starting with ASTER which deals with the core of HE business – the dissemination, collection and grading of student papers. Targeting this particular system initially will ensure buy-in from academics as well as the student body. With a successful implementation through the institutions existing VLE, additional services can be provisioned incrementally. Introducing DLT within the HE sector would also provide the much needed, positive exposure which has been marred by groups and individuals misusing the technology and tainting it with the perception of untrustworthiness.

References

1. Cachin, C., et al.: Architecture of the hyperledger blockchain fabric. In: Workshopon distributed cryptocurrencies and consensus ledgers. vol. 310 (2016)
2. Diffie, W., Hellman, M.: New directions in cryptography. IEEE transactions onInformation Theory **22**(6), 644–654 (1976)
3. Etherscan.io: Ethereum (ETH) Blockchain Explorer (2019), https://etherscan.io/
4. FoysalM: ASTER (2019), https://github.com/FoysalM/ASTER
5. Ico.co.uk: Penalties (2018), https://ico.org.uk/for-organisations/ guide-to-data-protection/guide-to-law-enforcement-processing/ penalties/
6. Ipfs.io: Datasets (2019), https://awesome.ipfs.io/datasets/
7. Mackey, T.K., Shah, N., Miyachi, K., Short, J., Clauson, K.: A Framework Proposalfor Blockchain-Based Scientific Publishing Using Shared Governance. Frontiers in Blockchain **2**, 19 (2019). https://doi.org/10.3389/fbloc.2019.00019, https://www.frontiersin.org/article/10.3389/fbloc.2019.00019
8. Multichain: Stream Confidentiality (2019), https://www.multichain.com/developers/stream-confidentiality/
9. Obinna, O., Pfluegel, E., Clarke, C.A., Tunnicliffe, M.J.: A Multi-Channel Steganographic Protocol for Secure SMS Mobile Banking. In: The 12th International Conference for Internet Technology and Secured Transactions (ICITST-2017). IEEE, Cambridge (dec 2017)
10. Onalo, S., Deepak, G.C., Pfluegel, E.: Virtual Private Blockchains : Security Overlays for Permissioned Blockchains. In: The Fifth International Conference on Cyber-Technologies and Cyber-Systems CYBER 2020. IARIA XPS Press, Nice, France (2020)
11. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signaturesand public-key cryptosystems. Communications of the ACM **21**(2), 120–126 (1978)
12. Shamir, A.: How to share a secret. Communications of the ACM **22**(11), 612–613 (1979)