A Privacy-Preserving Platform for Recording COVID-19 Vaccine Passports

Masoud Barati, William J. Buchanan, Owen Lo
Blockpass ID Lab
Edinburgh Napier University
Edinburgh, UK
{M.Barati, B.Buchanan, O.Lo}@napier.ac.uk

Omer Rana
School of Computer Science & Informatics
Cardiff University
Cardiff, UK
ranaof@cardiff.ac.uk

***Abstract*—Digital vaccine passports are one of the main solutions which would allow the restart of travel in a post COVID-19 world. Trust, scalability and security are all key challenges one must overcome in implementing a vaccine passport. Initial approaches attempt to solve this problem by using centralised systems with trusted authorities. However, sharing vaccine passport data between different organisations, regions and countries has become a major challenge. This paper designs a new platform architecture for creating, storing and verifying digital COVID-19 vaccine certifications. The platform makes use of InterPlanetary File System (IPFS) to guarantee there is no single point of failure and allow data to be securely distributed globally. Blockchain and smart contracts are also integrated into the platform to define policies and log access rights to vaccine passport data while ensuring all actions are audited and verifiably immutable. Our proposed platform realises General Data Protection Regulation (GDPR) requirements in terms of user consent, data encryption, data erasure and accountability obligations. We assess the scalability and performance of the platform using IPFS and Blockchain test networks.**

***Index Terms*—Blockchain, smart contracts, InterPlanetary file system (IPFS), data privacy, General data protection regulation (GDPR)**

I. Introduction

COVID-19 has provided the world with an exceptional challenge. The first relates to the methods that countries use to suppress the spread of the virus and the second is how to re-start economies [1]. One of the core methods that could be used to re-start travel within each country is to implement a vaccine passport, and which could be used in future pandemics. Some of the initial approaches to the creation of a vaccine passport have turned to mandatory and centralised approaches using the PKI (Public Key Infrastructure) [2], others have proposed a non-mandatory and decentralised approach [3]. Within a centralised approach, we normally use a single trust authority to check the signatures on passports. This authority will then check the public key of the signer of the passport, and accept it, if it is a trusted entity. In this way, trusted health authorities can sign their own passports whenever someone is immunised. This approach, though, while relatively easy to implement, leaves the centralised infrastructure open to breaches, along with a major problem around the revocation of signing keys.

Blockchain and smart contracts have been combined with platforms creating vaccine certificates, to increase trust in such certificates and privacy of citizens data. In [4], a Blockchain-based solution involving self-sovereign identification, decentralised storage and re-encryption proxies was proposed. The approach used Ethereum smart contracts to run transaction calls and record events containing medical information and COVID-19 test updates. A Blockchain-based approach that prevents information tampering such as COVID-19 test results was presented in [5]. The approach supports monitoring of COVID-19 spread at an earlier stage via a passengers' travel history. A scalable, Blockchain-based platform for data sharing of COVID-19 and vaccination passports was proposed in [6]. The evaluation of the designed platform was performed with 27 Blockchain nodes, each of which represents a European member state. A platform for secure COVID-19 passports and digital health certificate, called NovidChain, was built in [7]. The platform restricted the propagation of COVID-19 while supporting privacy concerns and ensuring citizen's self-sovereignty for accessing their data. Although all these approaches provide secure mechanisms for creating vaccine certificates, none of them considers user consent before processing their personal data. Moreover, the data accountability of actors and the right to be forgotten which are the main requirements of General Data Protection Regulation (GDPR) were not studied in the reviewed approaches. GDPR is a European legislation for protecting personal data and enables citizens to control how their data is collected and manipulated by processing entities (actors) [8, 9].

In order to address the aforementioned challenges, this paper presents an architecture for a Blockchain-based platform that creates and verifies digital COVID-19 vaccine passports. The platform makes use of IPFS for storing and distributing citizens' passport data securely. It also involves a smart contract factory to improve transparency, trust and data privacy of citizens.

Identify applicable funding agency here. If none, delete this.

Our proposed platform supports GDPR by implementing smart contracts that: (i) receive and record user consent, (ii) verify the operations of actors on passport data, and (iii) track the realization of the right to be forgotten. The proposed smart contracts are deployed and tested using the Ethereum virtual machine, and their costs and mining time are investigated in Blockchain test networks.

The rest of the paper is structured as follows. Section II briefly describes digital vaccine passports and IPFS. Section III describes the architecture of the platform together along with a number of protocols used in the implementation. Section IV assesses the time taken for creation of vaccine passports via IPFS and the costs required for deploying our implemented smart contracts and their transactions. Finally, Section V concludes the paper and identifies directions for future work.

II. Background & Context

This section provides brief descriptions of digital vaccine passports and the InterPlanetary File System (IPFS).

A. Vaccine Passports

Phelan [11] outlines that those who hold vaccination passports could be exempt from self-isolation when they travel.

1) Digital Green Certificate: In order to re-start international travel, the EU Commission has defined the Digital Green Certificate (DGC) programme [2], which defines three certificates: *vaccination certificates*, *test certificates* (NAAT/RT-PCR test or a rapid antigen test), and *certificates for persons who have recovered from COVID-19*. This makes use of a centralised system that will receive the digitally signed records. Each of these will be signed by a trusted health care entity, and checked against the public key of that trusted entity. In the best case, key pairs would be issued to every trusted health care professional to sign passports. This would allow fine-grain control on signing and audit each signing authority. If there was a breach of the private key though, the public key would be revoked, and it would have a minimal impact. What is likely to happen is that a countrywide health authority will have a single signing key pair. A single breach of the private key will bring down every single DGC signed by that authority, as all of the passports will be marked as untrusted. For cybercriminals, the private key will be a key target, as it will be of significant worth to them on the open market.

B. IPFS

IPFS [12, 13] is a distributed, Peer-to-Peer (P2P) content sharing protocol. Traditionally, resources have been shared using a location-based approach (e.g. a URL or file path). In IPFS, every individual resource which is to be shared on the P2P network is identified and located using an identifier which is derived directly from the content of the resource.

IPFS does not centralise the storage of resources. Instead, peers on the network will distribute data individually and any peer who downloads the content will also become a distributor of that data. Furthermore, the P2P nature of IPFS means we can reduce the latency involved in sourcing data, along with building resilience. IPFS can be used to represent any number of digital content including websites, folders, images, documents and even databases.

IPFS distributes data on the P2P network by first breaking down resources into blocks of 256×1024 (262,144) bytes by default [14]. Breaking down resources into blocks allows for deduplication (thus saving space) and storage of content in a distributed manner. Each block of data is content addressable using a Content-Identifier (CID). A Merkle Directed Acyclic Graph (DAG) data structure is used to represent each block of data and dictate the relationship between resources (e.g. parent folder and child files). Lastly, a Distributed Hash-Table (DHT) is used to allow peers to route and locate desired resources on the network (i.e. which peer is storing certain blocks and where they are located). The sections which follow describe CIDs, Merkle DAG and DHT in greater detail.

CID: are used in IPFS to achieve the goal of sharing resources using a content-based approach, assigning a unique identifier to each content resource (e.g. text file, image file, images and so on) to be shared on the IPFS network. *self-describing* identifier [15] which uses a cryptographic hash (SHA-256) to address the content. Since the CID hash is derived directly from the content (i.e. the text "hello world"), the CID will remain the same regardless of the filename or any other associated metadata. This allows for a degree of assurance that one is downloading the correct content from the IPFS network so long as the CID is known and trusted.

Merkle DAG [16] is a cryptographic hashing function to represent and derive nodes of data from a root node. It is commonly described as a Merkle *tree* due to the fact that the data structure it represents resembles an upside down tree (where the top node is the root). A Merkle tree can be used to verify the integrity of a data structure in a scalable manner since the root node's cryptographic hash can be used to verify the entire data structure represented by this algorithm. A Merkle DAG is an acyclic variation of a Merkle Tree with unique properties. Firstly, data structures represented by Merkle DAG are *directed* which means there is a forward direction defining the relationship between two nodes (e.g. parent folder points to child document). In the context of IPFS, Merkle DAG is used to represent folders and files shared on the P2P network.

DHT: is a key-value lookup table which maps content hash values (CID) to the location of content (i.e. peers which host the files) in IPFS. DHT is used for routing and informing peers where resources are located on the P2P network [12]. Each peer on the IPFS network will store and maintain a list of known peers as new nodes

Fig. 1. Overview of the aims of GLASS [10]

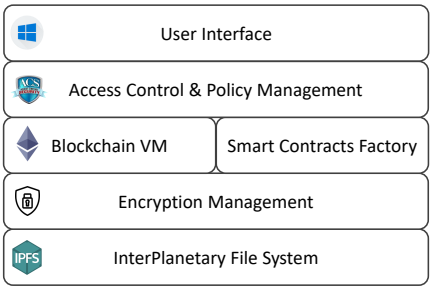

Fig. 2. The architecture of platform registering vaccine passports

join the network. The DHT algorithm implemented by IPFS is named Kademlia [17]. Kademlia uses a unique address in the range of 0 to 2^{256-1} to identify each peer on the IPFS network [18] and uses the exclusive OR (XOR) function to calculate the distance between each peer (thus allowing nodes to determine their nearest neighbours). This approach allows for the node lookup time to be $O(log(N))$ (logarithmic time) [18, 17] therefore ensuring scalability in the IPFS network even with a large number of peers.

III. Systems Architecture

A conceptual architecture for recording and verifying online COVID-19 vaccine passports is proposed in Fig. 2, and consists of the following layers: **InterPlanetary file system (IPFS)** is a peer-to-peer network for storing and sharing the information relevant to the citizens who have been vaccinated. In Scotland, such information includes *surname(s)*, *forename(s)*, *DOB*, *country of vaccination*, *identification number*, *dose number*, *dates of dosage*, *manufacturer*, *vaccine product* and *vaccine/prophylaxis*. The information is stored in IPFS through an administrator in a medical centre offering the vaccines. IPFS generates a content identifier (CID), which is a label used to point to each citizen's record/ file.

Encryption management anonymises or creates a hash for each CID generated by IPFS. The anonymisation is used for protecting CIDs from unauthorised access. The layer also keeps CIDs and their hashed versions in a local database.

Blockchain virtual machine and smart contracts factory hosts the following smart contracts for storing and monitoring immunity passports through a Blockchain.

- **Policy contract** involves two functions, called as `purpose()` and `vote()`. The former records what operation (i.e., read, write etc.) will be executed by which actor on citizen's data. Each record shows a purpose of data processing by the actor who is a third party processing passports' data. As an example, an actor can be the provider of a cloud-based service who has a contract with medical centers in order to collect and profile the data for analytic purposes.
  The latter function retrieves the purposes of data processing from the Blockchain and stores the positive/negative citizen's consent for each retrieved purpose in the Blockchain. The deployer of the contract is administrator that provides citizens with deployment address to receive their votes (positive or negative consents).
- **Log contract** sends the anonymised version of CID along with its creation time into the Blockchain network. Such records are used as public keys for accessing the passports details. The *contract deployers* are trusted administrators identified by medical centres.
- **Access contract** logs every access to CIDs in a Blockchain. It logs the operation (i.e., read, write, delete, and so on) which is executed by an actor on citizens' data within IPFS and submits it to the Blockchain. The contract is deployed by the *access control manager* in the system.
- **Verification contract** provides the audit trail of actors processing or accessing to citizens' data. The contract involves a function, called as `verify` which identifies the actors collecting or manipulating vaccine records without getting positive consents from citizens as violators. The function is activated by a trusted third party, as referred to *arbiter*.

The deployers or agents existing in upper or lower layers interact with the proposed contracts to record data in Blockchains.

Access control and policy management establishes a role-based mechanism for reading or updating citizens data. Users based on their roles can access to CIDs and vaccine passports details, and a copy of such access will be sent into Blockchain.

The layer also determines a set of privacy policies in forms of ⟨"*actor*", "*operation*", "*purpose*"⟩, as referred *data processing purposes*. An administrator in the layer communicates with the smart contract factory to store such purposes of data usage in a Blockchain.

User interface implements a user-friendly and front-end decentralized application (DApp) for citizens in order to readily interact with the platform. Technically, it is connected to the contracts' interfaces created and hosted on Ethereum Blockchain virtual machine. The interface also enables citizens to retrieve the purposes of data processing from Blockchain with a more legible format and get their votes (positive/ negative consent) to the predefined purposes. In fact, the citizens' consents will be considered as the inputs for `vote()` function involved in the *policy* smart contract.

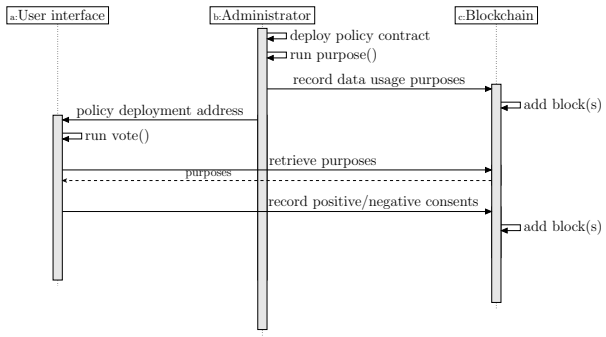

Fig. 3. A protocol for agreement phase

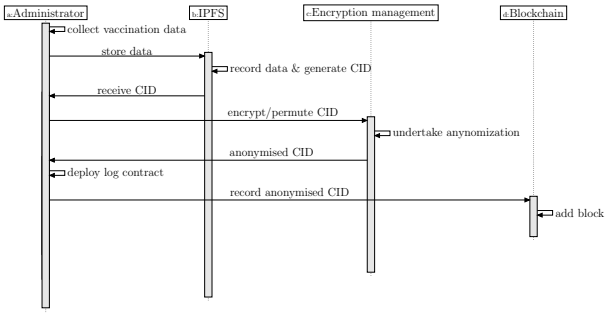

Fig. 4. A protocol for passport creation phase

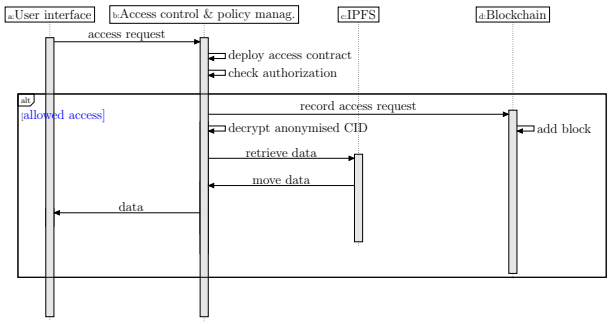

Fig. 5. A protocol for accessing to passport data

There are four phases for realizing the architecture: *agreement*, *passport creation*, *access control* and verification.

A. Agreement phase

This phase presents a protocol for demonstrating the attractions among citizens, system administrator and Blockchain for recording purposes of data processing and citizens' consents. Figure 3 shows the protocol in the form of a sequence diagram. As seen, the main entities are user interface, system administrator and Blockchain. The administrator as the data controller, first, deploys the policy contract and activates the purpose function to send data usage purposes into Blockchain. Precisely, each record contains: (i) *actor* who will update or collect citizen' passport data (ii) *operation* that shows what actions (e.g., read, write etc.) will be carried out by the actor on citizens' data, and (iii) *purpose* that describes the operation is used for what. Once such records have been added to the Blockchain network, the administrator provides a user interface with the deployment address of policy contract in order to make the records accessible to end users (citizens). The user interface, then, by activating the vote function, enables users to retrieve the data usage purposes from Blockchain and freely give their consents to them before any processing on their personal data. The users' votes will be kept in the Blockchain as evidence for future verification. This phase realises Recitals (32) and (43) of GDPR under which data subjects (citizens) should give their consent for any operation undertaken by data processors (actors) on their personal data.

B. Passport creation phase

This phase represents the steps in which the vaccine passports details will be stored in IPFS and their associated anonymised CIDs are recorded in Blockchain. Figure 4 depicts the protocol of the phase. After collecting citizens data by passport administrator, the data is sent to IPFS. Then, the data is recorded and a CID is automatically generated via IPFS. Once the CID has been received, it is forwarded to the encryption management layer so as to be anonymised. Following that, the passport administrator, by deploying log contract, submits the anonymised version of CID to Blockchain.

There are several techniques for data anonymisation [19] such as hashing, permutation among others, each of which can be exploited for mapping the CIDs into the anonymised ones.

C. Access control phase

This phase monitors and verifies all the access requests to passport citizen data. The sequence diagram depicted in Fig. 5 is a protocol for the access control. Citizens and trusted parties through user interface are able to send their request for access to passports data. The request also contains the operation (such as update and so on) that will be carried out on the data. Upon the receipt of request, the access control management service's agent checks the authorisation of requester. In case of authorised access, the agent deploys the access smart contract in order to record *requester ID*, *access time* and *permitted/ executable operation(s)* (e.g., view, update etc.) in the Blockchain.[1] Such records are used for future verification. Following that, the hashed/anonymised version of CID is decrypted, and finally the passport data is retrieved from IPFS to be accessible for the requester.

[1]For privacy purposes, a hashed version of requester ID, which refers to their Blockchain account is stored on-chain.

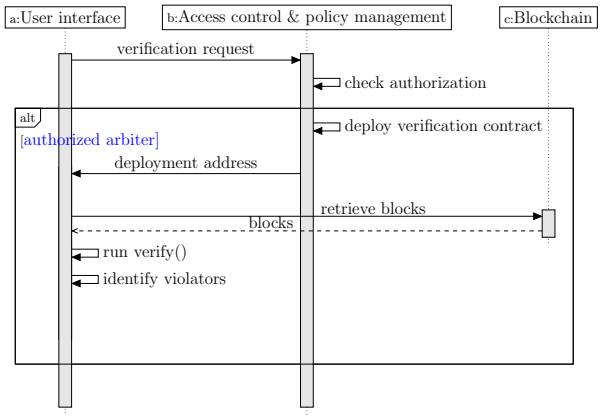

Fig. 6. A protocol for verifying GDPR violations

Algorithm 1 Verifying actors

Let V be a set denoting violators
Input: policy & access deployment addresses
Output: V
function VERIFY
  $V \leftarrow \emptyset$;
  if $A_e \not\subseteq A_c$ **then**
    $V \leftarrow V \cup A_e \setminus A_c$;
  end if
  for all $a \in A_c$ **do**
    if $\mathcal{O}_a \not\subseteq O_a$ **then**
      $V \leftarrow V \cup \{a\}$;
    end if
  end for
  return V;
end function

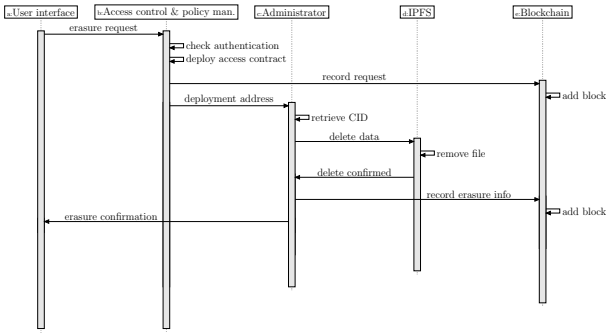

Fig. 7. A protocol for erasure of vaccine data

D. Verification phase

This phase, through the protocol represented in Fig. 6, detects the violators who access or manipulate citizens data without getting their consents. The arbiter via user interface sends their verification request for reporting violators once claimed by citizens or legal offices. After approving the authorization of arbiter by access control management service's agent, the verification contract is deployed and its address is forwarded to the arbiter. The deployment address enables the arbiter to access the records already stored by both *policy* and *access* contracts. Then, the arbiter, by running the verify function, flags any violation on citizens data and identifies violators. The verify function checks the following items to detect violations:

1) whether the actors stored by access contract conform to those logged via policy contract or not;
2) whether the operations of each actor recorded through access contract conform to those recorded via policy contract or not;
3) whether the operations of each actor logged by access contract were already confirmed by the data subject (citizen) or not.

Assuming that A_c is the set of actors with positive consent from data subject via the policy contract; A_e is the set of actors executed operations on citizen passport's data and recorded by the access contract; O_a is the set of operations of actor $a \in A_c$ got positive consent from data subject via the user policy contract; and $\mathcal{O}_a$ is the set of operations executed by $a \in A_c$ on passport data and recorded via the access contract.

Given these assumptions, Algorithm 1 presents the verification of actors implemented as a part of *verify* function.

As seen from the algorithm, a violation is flagged if: (i) an actor processes passport data without the confirmation of data subject; and (ii) an accepted actor executes an operation already rejected by the data subject.

E. Right to be forgotten

Citizens have the right to get from the passport administrator (data controller) the erasure of their data without any delay (Art. 17 of GDPR). In order to realize this GDPR principle through our proposed architecture, a protocol is presented in Fig. 7. It provides a basis for data accountability of administrators to track whether the citizen's data have been deleted without undue delay. As represented from the figure, a citizen, through user interface, is able to submit a data erasure request, which is received by access control management service's agent. The identification and permission of the citizen is then verified by the agent and the access smart contract is deployed so as to record a copy of the request in a Blockchain. Upon the receipt of the deployment address of the contract by the passport administrator, the CID related to vaccine passport's data is collected and the citizens' personal data is removed within IPFS. After erasing the data, a confirmation is sent to the citizen. Moreover, a copy of such confirmation denoting the data have been removed by who and when is stored in the Blockchain as evidence for future verification.

In order to detect any violation with, the *verification* smart contract is extended to include a `erase_verify()`

function. By retrieving the blocks containing erasure requests/ confirmation and created by the access contract, the function flags the violator if:

- the erasure confirmation has not been recorded by the administrator in the Blockchain; or
- the time difference between erasure request and erasure confirmation logged by the administrator is greater than a short deadline already determined through the purposes of data processing in the agreement phase.

Both cases are investigated by the arbiter with regards to a claim received from the citizen who is the owner of the passport. For instance, after the submission of an erasure request, if the citizen observes that their data is still available in IPFS while the deadline had been passed, a claim can be made by the citizen and submitted to the arbiter. The claim should involve the erasure request's time and anonymised CID.

IV. Experimental Results

Our experiments cover the evaluations related to the creation of vaccine passports using IPFS and the implementation of our proposed smart contracts using Blockchain test networks.

A. IPFS CID Generation Time for Vaccination Passports

To demonstrate the scalability of our proposed solution, the CID generation time in IPFS was evaluated. We chose to measure the time taken to generate 10, 20... up to 100 CID values for simulated vaccination passports which may be added to the IPFS network. Our vaccination passport is encoded as a JSON object, and the contents are derived from the fields used by NHS Scotland in real-world vaccination scenarios (described in Section III). The JSON string for an example passport is shown in Appendix A.

A script was used to generate the passport data. Each passport created will generate a unique Community Health Index (CHI) number: a 10-digit value used to identify patients in Scotland. The use of a unique CHI number for each passport ensures the IPFS CID generated will also be unique (recall that the IPFS CID is derived from the content of a resource using cryptographic hashing). Each passport object generated is 452 bytes in size.

A private instance of the IPFS network [20] was created as a Docker image for the purposes of this evaluation. A private IPFS network (one which is isolated from the public network) was used to ensure no experimental data was accidentally added to the public P2P network. The Linux **time** [21] command was used to monitor the time taken for the IPFS command to generate CIDs for 10, 20, ... 100 CIDs. An example of the *time* command used in conjunction with IPFS for evaluating time taken for generating ten unique vaccination passport CIDs is as follows[2]:

[2]We attribute this approach to redirecting the time output to [22]

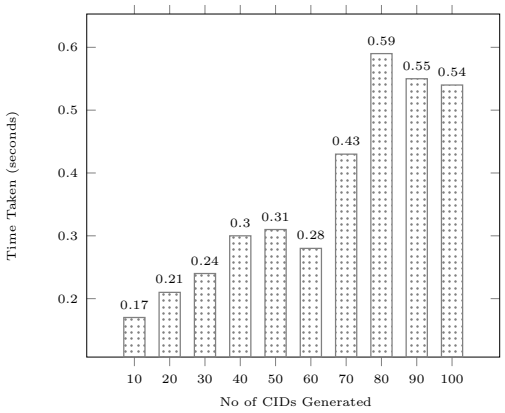

Fig. 8. IPFS CID Generation Time

```
{ time ipfs add -r 10 ; } 2> 10_result.txt
```

In the above code listing, we monitor the time taken (user+sys time) for the **ipfs add -r 10** command to execute. The **ipfs add -r** command is used to generate CIDs recursively for all content (i.e. 10 simulated vaccination passport JSON objects in this example) within a folder named **10**. The same measuring approach is taken for 20 unique passports, 30 and so on.

Figure 8 shows the generation time for 10, 20... 100 CIDs derived from simulated vaccination passport data. For 10 CIDs, the time varied between 0.37s and 0.59s. On 6 June 2021 there were 387,286 total vaccinations given to UK citizens based on figures released by the UK government [23]. If we assume a vaccination passport entry was to be created for all 387,286 vaccination events, and assume it takes a maximum of 0.59s to generate 100 CIDs, our results show that it would take around **38 minutes** to generate a CID for all passports. This demonstrates that the CIDs can scale to a large population size.

B. Investigation of proposed smart contracts

A prototype has been developed using both Ganache [24] and Ropsten [25] test networks. We implemented our smart contracts on Ethereum via Solidity language. Ganache is a local test network that provides multiple default gas and ether values, which can be applied as a currency to change Blockchain states when running function calls. Ropsten is a public test network involving a set of miners and gives detailed information relevant to miners. However, it has a gas limit of 4712388 for executing a smart contract. Our proposed smart contracts have been written with a minimum gas usage for each function activation. They were compiled and successfully tested using Remix, being a browser-based development environment for Solidity. The contracts *Policy*, *Log*, *Access* and *Verification* were deployed in the aforementioned networks. The amount of gas used for contract deployment was 792065 for *Policy*, 157339 for *Log*, 796253 for *Access*, and 1223998 for *Verification*. The

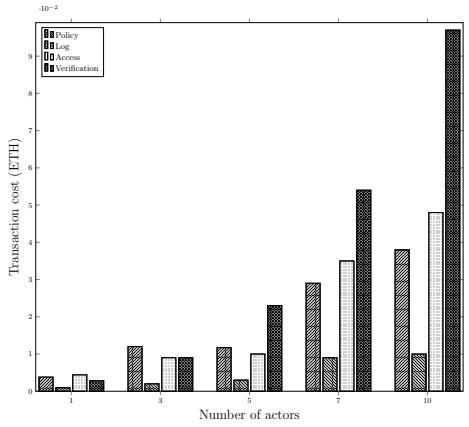

Fig. 9. The relationship between number of actors and cost

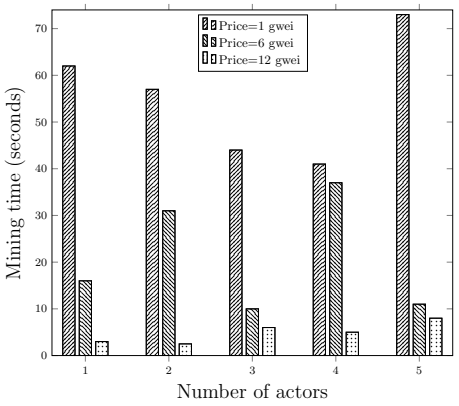

Fig. 10. The relationship between number of actors and mining time

results represent the computational cost for executing each contract. However, changing the number of actors and their access requests has an impact on the transaction costs and mining time.

1) Number of actors and transaction costs: The experiment which changes the number of actors, ranging from one to ten, evaluates the cost required for activating transactions. The assumption is that the gas price unit is 20 *gwei* and the number of operations (i.e., read, write, delete) carried out by each actor on vaccine data is three. Our proposed smart contracts have been deployed in the Ganache test network. We calculated the average costs in Ether (ETH) after five times execution of functions with different parameters (values). Figure 9 illustrates the results of this experiment. As seen, the lowest costs in Ether are allocated to the transactions with one actor and the highest values belong to those involving ten actors.[3] Furthermore, when the number of actors increases, the verification cost rises more sharply compared to the other contracts. In fact, the *Verification* contract has a high complexity, since it must call both *Policy* and *Access* smart contracts in order to check the GDPR compliance of actors and report violators.

2) Number of actors and mining time: This investigation represents the impact of changing the number of actors on the time taken for mining process under different gas prices. The number of actors is varied from one to five and each of which executes two operations on citizens data. We have various scales of gas prices (i.e., 1, 6 and 12 *gwei*). The Ropsten test network was used to get the results of this experiment, as it gives a measurement of the time taken from execution to mining of a block. The *verification* contract was deployed in Ropsten and its *verify* function was activated five times to reach an average mining time. Figure 10 shows the results of this evaluation. Given a fixed gas price, there is a fluctuation in the trend of the chart. As a result, the time totally depends on the interest of miners to validate/ mine the transactions created from the activation of *verify* function and the number of actors does not have an impact on mining time. However, when the amount of gas price increases, the mining time decreases significantly. It shows that higher gas prices motivate miners to accelerate the mining process and block creation.

[3]The actors for *Log* contract are citizens, whose anonymised CIDs are recorded in Blockchain.

V. Conclusion

The design of a Blockchain-based platform for the creation of online COVID-19 vaccine certificates is proposed. The platform uses IPFS and smart contracts to support privacy of citizens' information and supports data accountability for third party access/ processing of this information. The purposes of data processing (carried out by actors) is automatically sent into a Blockchain and the platform enables citizens to give a vote (positive /negative consent) for each purpose via a smart contract. The proposed approach meets GDPR requirements, and only non-sensitive data is stored within the Blockchain for auditing purposes. Compared to other Blockchain-based platforms for cloud and IoT ecosystems for keeping and verifying patient data [26, 27], our platform provides a technical solution for checking data erasure requests, which is a significant user right in GDPR (referred to as "right to be forgotten"). A vaccine passport template has been implemented as a prototype, and our evaluations shows that it takes less than one second to generate 100 passport CIDs in IPFS. The created smart contracts have been tested in both Ganache (on a local machine) and Ropsten (a global Blockchain network) environments and results in the transactions cost increasing noticeable when the number of actors increases (as expected). The results of these experiments can be used to support capacity planning of a vaccine certificate network. Given a fixed gas price used for execution of smart contract opcodes, the investigation demonstrates that miners can take an arbitrary time for mining blocks. Future work

focuses on the implementation of both access control and encryption management layers of the designed architecture. The development of the proposed platform in cloud environment and the management of CIDs generated by IPFS remain other aspects for future investigation.

Acknowledgment: This work has been carried out in the GLASS (SinGLe Sign-on eGovernAnce paradigm based on a distributed file exchange network for Security, transparency, cost effectiveness and truSt) project [10].

References

[1] L. H. Chen, D. O. Freedman, and L. G. Visser, "Covid-19 immunity passport to ease travel restrictions?" *Journal of travel medicine*, vol. 27, no. 5, p. taaa085, 2020.

[2] E. Commission, "Covid-19: Digital green certificates," 2021. [Online]. Available: https://ec.europa.eu/info/live-work-travel-eu/coronavirus-response/safe-covid-19-vaccines-europeans/covid-19-digital-green-certificates_en

[3] M. Ouellette and M. L. Shaw, "For a decentralized vaccine passport," *HEALTH POLICY*, 2021.

[4] H. R. Hasan, K. Salah, R. Jayarama, J. Arshad, I. Yaqoob, M. Omar, and S. Ellahham, "Blockchain-based solution for covid-19 digital medical passports and immunity certificates," *IEEE Access*, vol. 8, pp. 222 093–222 108, 2020.

[5] G. S. M. MK, S. R. K. Somayaji, T. R. Gadekallu, P. K. R. Maddikunta, and S. Bhattacharya, "An incentive based approach for covid-19 using blockchain technology," *arXiv preprint arXiv:2011.01468v1*, 2020.

[6] J. L. Hernandez-Ramos, G. Karopoulos, D. Geneiatakis, T. Martin, G. Kambourakis, and I. N. Fovino, "Sharing pandemic vaccination certificates through blockchain: case study and performance evaluation," *arXiv preprint arXiv:2101.04575v1*, 2021.

[7] A. Abid, S. Cheikhrouhou, S. Kallel, and M. Jmaiel, "Novidchain: Blockchain-based privacy-preserving platform for covid-19 test/vaccine certificates," *Software: Practice and Experience*, pp. 1–27, 2021.

[8] M. Virvou and E. Mougiakou, "Based on GDPR privacy in UML: case of e-learning program," in *8th International Conference on Information, Intelligence, Systems and Applications, Larnaca, Cyprus*. IEEE, 2017.

[9] A. Aljeraisy, M. Barati, O. Rana, and C. Perera, "Privacy laws and privacy by design schemes for the internet of things: A developer's perspective," *ACM Computing Surveys*, vol. 54, no. 5, pp. 1–38, 2021.

[10] G. P. Members, "Glass project," 2021. [Online]. Available: https://www.glass-h2020.eu/

[11] A. L. Phelan, "Covid-19 immunity passports and vaccination certificates: scientific, equitable, and legal challenges," *The Lancet*, vol. 395, no. 10237, pp. 1595–1598, 2020.

[12] J. Benet, "Ipfs-content addressed, versioned, p2p file system (draft 3)," *arXiv preprint arXiv:1407.3561*, 2014.

[13] IPFS, "What is IPFS?" 2021. [Online]. Available: https://docs.ipfs.io/concepts/what-is-ipfs/

[14] IPFS, "go-ipfs/add.go at master · ipfs/go-ipfs," 2021. [Online]. Available: https://github.com/ipfs/go-ipfs/blob/master/core/commands/add.go

[15] multiformats, "multiformats/cid: Self-describing content-addressed identifiers for distributed systems," 2020. [Online]. Available: https://github.com/multiformats/cid

[16] R. C. Merkle, "A digital signature based on a conventional encryption function," in *Conference on the theory and application of cryptographic techniques*. Springer, 1987, pp. 369–378.

[17] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," in *International Workshop on Peer-to-Peer Systems*. Springer, 2002, pp. 53–65.

[18] IPFS, "Distributed Hash Tables (DHTs) — IPFS Docs," 2021. [Online]. Available: https://docs.ipfs.io/concepts/dht/#kademlia

[19] S. Murthy, A. A. Bakar, F. A. Rahim, and R. Ramli, "A comparative study of data anonymization techniques," in *5th International Conference on Big Data Security on Cloud, High Performance and Smart Computing and Intelligent Data and Security, Washington, USA*, 2019, pp. 306–309.

[20] IPFS, "go-ipfs/experimental-features.md at master · ipfs/go-ipfs," 2021. [Online]. Available: https://github.com/ipfs/go-ipfs/blob/master/docs/experimental-features.md#private-networks

[21] Linux Foundation, "time(1) - Linux man page." [Online]. Available: https://linux.die.net/man/1/time

[22] January, "How to redirect the output of the time command to a file in Linux?" 2021. [Online]. Available: https://stackoverflow.com/a/13356654

[23] UK Government, "Vaccinations in the UK — Coronavirus in the UK," 2021. [Online]. Available: https://coronavirus.data.gov.uk/details/vaccinations

[24] Ganache, "Ganache testnet," 2021. [Online]. Available: https://github.com/trufflesuite/ganache

[25] Ropsten, "Ropsten testnet pow chain," 2021. [Online]. Available: https://github.com/ethereum/ropsten

[26] M. Barati and O. Rana, "Tracking GDPR compliance in cloud-based service delivery," *IEEE Transactions on Services Computing*, 2020.

[27] M. Barati, O. Rana, I. Petri, and G. Theodorakopoulos, "GDPR compliance verification in internet of things," *IEEE Access*, vol. 8, pp. 119 697–119 709, 2020.

Appendix

Example Vaccination Passport JSON object: Example of JSON encoded vaccine passport data for evaluation of IPFS CID generation time – based on data used by NHS Scotland. A total of 100 vaccination passport objects were created of 452 bytes each. A unique number was provided in the CHI field (as unique patient identifier) when creating a passport to ensure uniqueness of hash value when generating the IPFS CID.


```
{
"COVID-19 Vaccination Status": {
  "CHI": "0000000001",
  "Surname(s)": "Doe",
  "Forename(s)": "John",
  "DOB": "02/01/1965",
  "Disease targeted": "COVID-19",
  "Country of vaccination": "Scotland",
  "Issued by": "NHS Scotland",
  "Doses received": "2",
  "Dose 1 of 2": "01/06/2021",
  "Manufacturer": "Moderna Biotech Spain S.L.",
  "Vaccine medicinal product":
            "COVID-19 Vaccine Moderna",
  "Vaccine/Prophylaxis":
            "SARS-CoV-2 mRNA vaccine",
  "Dose 2 of 2": "30/06/2021"
  }
}
```