Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
A blockchain-based infection tracing and notification system by non-fungible tokens

Alessio Ferone *, Antonio Della Porta
Department of Applied Science, University of Naples Parthenope, Italy

A R T I C L E   I N F O
Keywords:
Blockchain
Contact tracing
Smart contract
ERC721
SARS-CoV2

A B S T R A C T
SARS-CoV2 pandemic is heavily affecting our lives. Many actions have been undertaken to slow down its expansion and, among the others, contact tracing applications are the less invasive to monitor the spread of the virus. The idea behind contact tracing is to track contacts between people by the exchange of identifiers, not linked to individuals, exploiting the use of Bluetooth Low Energy (BLE) technology to estimate the duration and proximity of contacts. The data collected in this way is used for the sole purpose of notifying a potential contact with an infected person without revealing their identity and location. This paper presents a contact tracing protocol based on blockchain technology that exploits smart contracts for reporting contacts at risk of contagion. The novelty of the proposed solution is the use of Non Fungible Tokens (NFT) to guarantee user privacy through a decentralized approach, equipped with a reliable non-proprietary notification mechanism that allows public access to anonymous infections data.

1. Introduction
In recent years, worldwide attention has been captured by the SARS-CoV-2 pandemic that has caused extensive damage to health and global economy. The current situation has highlighted how unprepared we are to deal with emergencies that may arise in the future. In this context, it is of fundamental importance to have tools to support the management of such emergencies. One of these is represented by applications for contact tracing [1], software applications that use devices equipped with BLE [2] technology (e.g. smartphones) to facilitate the tracking of contacts between people with the dual purpose of informing them of a possible contagion and monitor the spread of the virus. In particular, following [3], proximity-based contact tracing consists in the identification of proximity events, or situations in which two people are at a certain distance for a time sufficient to determine a possible contagion. Such an event is relevant for the purpose of monitoring infections when someone results positive to a diagnostic test and it is necessary to alert all the people with whom she came into contact during the infective period. The use of smartphone applications is not mandatory and is therefore necessary to incentivize their adoption by offering guarantees about security and privacy. Many of the models presented so far (DP-3T [4], the Exposure Notification framework by Apple/Google [5]) use a centralized back-end system for the notification of contagion and contact events at risk. The main goal of the proposed system is to guarantee users’ privacy through a decentralized approach by exploiting blockchain technology, yielding transparent operation, consultation and prevention of wide data breaches which can impact a massive group of people [6]. In particular, the novelty proposed in this work consists in the use of a special type of token called Non Fungible Tokens (NFT) whose characteristics allows to uniquely represent a contagion that can be anonymously notified to tested person and also made public. This approach allows the implementation of a non-proprietary information exchange mechanism capable of generating notifications of infections in a reliable way. The decentralized and public nature of blockchain system allows access to the recorded data about infections making them publicly available. The paper is organized as follows: in Section 2 a brief introduction to the employed technologies is given; in Section 3 works related to the proposed approach is analyzed; in Section 4 details of the proposed system are explained, while in Section 5 implementation details are given; Section 6 discusses the aspects related to the performance and the security of the proposed system; Section 7 concludes the paper.

2. Background
In this Section some of the background technologies employed in the proposed system are introduced.

2.1. Blockchain
A blockchain (or chain of blocks) can be defined as an open and distributed ledger that can record transactions between two or more parties efficiently and in a verifiable and permanent way. The data
structure consists of a list of records called blocks, which are linked together thanks to the use of cryptography. Each block contains the hash of the previous block, a timestamp and transaction data generally stored in a space-saving data structure, called the Merkle Patricia Tree. The blockchain is replicated independently on various nodes of the network in order to create a distributed ledger. To this end, a peer-to-peer network is created, in which each peer adheres to a common protocol for the exchange of information and for the validation of new blocks. Thanks to this feature, a block cannot be modified without altering all the previous blocks, which requires the consent of the majority of the peers in the network. The blockchain was born and developed as a solution to the more general problem of consensus in a distributed environment. Therefore a blockchain is definable as an immutable sequence of records if there are the conditions to reach consensus on a block in the presence of malevolent (or Byzantine) actors who try to change the order or content of transactions. In particular, the blockchain is a sequence of blocks which manages a complete list of transactions. Each block $B_i$ has only one parent block $B_{i-1}$ which is the block whose hash is contained within $B_i$. The $B_i$ block of a blockchain is called the genesis block, which is the only block that does not have a parent block. A block consists of:

- **Block header**: which includes information about the protocol, the block timestamp (in UNIX time) and the hash value of the parent block
- **Body of the block**: which includes transactions and other information depending on the protocol.

To limit the size of a block (and consequently the number of possible transactions), blockchain projects use various techniques. For instance, in Ethereum blockchain, Merkle Patricia Trees are used to reduce the size of the header while the size of a block depends on the gasLimit of the block (more details below). In the following we refer to the Ethereum blockchain. An extensive survey on blockchain technology and its applications can be found in [7], while, NG2021e819 specifically analyzes the use of blockchain technology in health-care applications. For instance, [8] proposes a blockchain-based platform for healthcare application that employs two separate chains for storing transactions, and each message needs the gasLimit field since the contract receiving the message could perform computational steps. The date field of a message is used to pass parameters to the contracts that receive them. In addition, contracts that are activated by a message can return results to senders. This behavior allows you to implement the procedure calling paradigm.

Transactions in a blockchain are made possible thanks to asymmetric cryptography. The purpose of its use is not to encrypt transactions but to sign and verify the transactions that are exchanged in the system. This mechanism is called a digital signature. Ethereum use the Elliptic Curve Digital Signature Algorithm (ECDSA) as a digital signature algorithm [10]. Each user using a blockchain system has a pair of cryptographic keys. The key pair consists of a private key and a public key (which derives in some way from the private key). The private key is kept secret (usually in software called wallets), while the public key is used to derive the address belonging to the user and for other transaction verification operations. An Ethereum address is a 20-byte string corresponding to the last 20 bytes of the Keccak256 (Kpub) value, where Kpub is the public key (derived from the private key thanks to ECDSA) and Keccak256 is the Keccak-256 hashing function. The status of the blockchain in Ethereum is made up of objects called accounts. Each account is represented in the state (key–value store) by:

- **Nonce**: a counter to make sure that each transaction can only be executed once;
- **Ether balance**: the total ether of the current account;
- **Contract code**: if present;
- **Storage**: space allocated to the account for data. Blank by default.

There are two types of accounts: EOA (externally owned accounts) and Contract Accounts. EOAs are accounts controlled by anyone who controls the private key associated with the address of the same, while Contract Accounts are accounts controlled by the code in the contract code field. A message that contains ether or information can be sent from an EOA to any other account. A Contract Account, on the other hand, is activated only when it receives a message from another account. The activation of the code of a contract can result in other messages, I/O on the storage or in the creation of new contracts.

### 2.2. Smart contracts and tokens

Smart contracts are programs written in a Turing-complete language (EVM Code) that allow anyone who writes them to define custom ownership rules, transaction formats and state transition functions. Smart contracts can be seen as autonomous actors that are located within the Ethereum execution environment. They execute a piece of code, specified during their creation, following a message or transaction and have direct control over their ether balance and allocated storage space to maintain their state. The term transaction refers to a data packet signed by an EOA that contains a message from the same to another account. A transaction contains:

- **nonce**: integer value equal to the number of transactions performed by the user;
- **gasPrice**: value corresponding to the fee that the user wants to pay for each computational step;
- **gasLimit**: value corresponding to the maximum number of computational steps that the transaction can perform;
- **to**: the recipient address of the transaction;
- **value**: the total of ether to be transferred from the sender to the recipient;
- **signature**: user’s signature;
- **data**: field to be filled with the data to be passed to a smart contract, optional.

Also, a transaction can contain the init field which is used to initialize a new contract. Smart contracts have the ability to send messages to other contracts. Messages are virtual objects that are never serialized and exist only in the Ethereum Virtual Machine (EVM). They are essentially like a transaction but are produced by contracts and not by EOA. A message is produced when a contract that is executing code executes the CALL opcode, which invokes another contract. In this way it is possible for the contracts to interact with each other as EOA. However, each message needs the gasLimit field since the contract receiving the message could perform computational steps. The date field of a message is used to pass parameters to the contracts that receive them. In addition, contracts that are activated by a message can return results to senders. This behavior allows you to implement the procedure calling paradigm.

Ethereum allows the implementation of a Token system through smart contracts. Cryptographic tokens represent programmable assets or access rights, managed by a smart contract and an underlying distributed ledger. They are accessible only by the person who has the private key for that address and can only be signed using this private key. They are executed in a distributed way by the network offered by the blockchain system on which they are deployed. They can also be publicly inspected both in the code and in the stored data. **Smart contracts** therefore offer a useful tool to build a **public and transparent** service, in which the code executed by the system can be verified by anyone who is granted access to the blockchain.

Cryptographic tokens represent a value associated with a good, a service or a right. They have common characteristics that make them advantageous and interesting compared to current tokenization systems:

- **Liquidity**, thanks to the fact that they are easily convertible into current currency or cryptocurrency;
- **Divisibility**, which allows the value to be divided into smaller units;
- **Negotiable**, which makes it possible to sell and transfer the property on value;
- **Proof of immutable ownership**, thanks to encryption.
Furthermore, the use of blockchain systems makes the processes related to tokens more transparent and accessible. Technically, cryptographic tokens are all based on the same concept: a smart contract that manages the ownership of a quantity of tokens and manages their governance. What characterizes them, however, are the rules and behaviors that a token follows, through which it is possible to classify them. Due to the variety of tokens applications, they can be divided into two types: fungible and non-fungible. The Ethereum community allows users to propose protocol changes through EIPs (Ethereum Improvement Proposals). Among these you can find proposals called ERCs (Ethereum Request for Comments), technical documents used by Ethereum’s smart contract developers. These documents define sets of rules to implement the tokens that will circulate in the Ethereum ecosystem. Among the others, two standards have been defined for these two types of tokens: ERC20 (fungible tokens) and ERC721 (non-fungible tokens). The standards are described in the documents EIP-20 [11] and EIP-721 [12] where the basic interface that smart contracts must implement to represent the assets is documented. ERC20 proposes a token with standard functions such as the exchange, the budget of addresses and the approval of operations on its budget by external actors (EOA or contracts). It also allows you to characterize the implemented token with a name, a symbol and a number of decimals (0–18). From this standard many others were born that add functionality to it or solve problems. ERC721 standard proposes a non-fungible or unique token. The difference from ERC20 and derivative tokens is that each ERC721 token is unique and identified by a 256-bit integer. Each non-fungible token (NFT) is transferable.

2.3. BLE and iBeacon

IBeacon technology uses Bluetooth Low Energy (BLE) technology to create a “region” that allows nearby devices to know exactly when they enter or leave that region. The iBeacon protocol specifies a packet of 25 bytes, of which the 20 bytes contained in the following fields can be modified:

- **Proximity UUID**: A 16-byte value that identifies the main region
- **Major**: 2 bytes used to define a sub-region within the region defined by theUUID
- **Minor**: 2 bytes that can be used to further subdivide the region defined by the Mayor field

The byte contained in the Measured Power field indicates the output power at which the beacon has been set, measured in dBm at a distance of one meter.

3. Related works

With the advent of the pandemic, numerous software solutions for contact tracing have been proposed, mostly based on the use of mobile devices equipped with BLE technology. This technology is exploited to emit and collect temporary pseudonyms from nearby devices, so that a user, if positive to a diagnostic test, can authorize the health authorities to disclose the identifiers produced by her/his device in the contagious period, allowing the other devices of the system to verify if there have been contacts at risk.

The protocols for contact tracing can be classified into centralized and decentralized according to the way in which the pseudonyms are generated. In the centralized protocols, identifiers are generated by a central server and assigned to devices upon registration. An example of such protocols is ROBERT [13]. A centralized system presents a severe security risk if considering the amount of data held by a single authority so that the consequences of an attack on the database could be catastrophic. The decentralized protocols, on the other hand, implement the procedure to generate the identifiers directly on the devices. Among the first decentralized protocols presented, DP-3T (Decentralized Privacy-Preserving Proximity Tracing), released by an European working group, presents two ways of creating pseudonyms: Low-cost and Unlinkable. In Low-cost mode the pseudonyms are generated starting from a single seed, that, once revealed, is used to reconstruct all the pseudonyms generated by the user to whom it belongs. In the Unlinkable mode, on the other hand, pseudonyms are not linked to each other and their disclosure is performed through a compact form, in the form of a probabilistic data structure. In addition to the privacy and management problems of the data collected by the applications, it is also necessary to consider the compatibility problems between the various mobile systems available. This aspect was addressed by a joint effort of Apple and Google who developed the protocol called Exposure Notifications. This protocol includes only a mechanism for creating and managing identifiers, inspired by the DP-3T Low-cost mode, at the operating system level. Developers employing this protocol are responsible of managing the backend and handling the notifications system. The Apple/Google framework is used for the development of various applications, including the Immuni [14] app, adopted by the Italian government as the official contact tracing application. The application is entirely open-source, both in the part of the mobile application and in the part concerning the server-side application.

An extensive review on contact tracing approaches can be found in a recent systematic literature review [15], where authors discuss the importance of adopting less-invasive and privacy-preserving technologies such as blockchain. According to the review, a blockchain-based contact tracing system has been found in the literature [16], whose main issues can be identified in the need for the user to register to the service and the use of users’ route locations. Both issues have been identified in [15] as factors that could prevent the adoption of the tracing system. Moreover the system is based on 5G networks that, at the time of writing, are not yet globally available. Also Authors in [17] highlight how the presence of many intermediary processes for reporting infections from the hospitals to the final institution, can increase the reporting time, which can make it difficult to respond promptly to infectious diseases. In [18] a detailed discussion on the blockchain-based contact tracing mechanism is provided and also the apps developed so far to combat the COVID-19 pandemic are reviewed. Authors conclude that the properties of the blockchain may facilitate the retrieval of accurate information in a transparent manner while preserving its integrity and users’ privacy. In [3], a systematic review of blockchain-based contact tracing applications is performed. [19] proposed the Connect platform that is the first example of blockchain-based Self Sovereign Identity to handle identities and activities of the users that can notify and eventual contagion to people in the proximity. In [20] , in order to solve the problem of blockchain’s scalability [21] , a framework that uses off-chain scaling mechanism of Interplanetary File System (IPFS) is proposed. The main idea is that any data that requires transactions should be stored on-chain and all the remaining should be stored off-chain. In [22] Authors propose a novel scheme to decouple user privacy by using a tracing chain with desensitized personal location information, only accessible by authorized personnel, for contact matching and a notification chain, where the match results are published by means of pseudonyms. The main problems of this approach are the complexity for handling two blockchains and the need of authorized person that can access users’ location info. In [23] a blockchain-based contact tracing solution that uses sound and BLE to detect proximity is proposed, where close contact is established if both Bluetooth and sound are detected between the communicating devices. Although the interesting use of sound to infer proximity, its use in conjunction with BLE can impose a severe load on battery life of devices. [24] proposed the P2B-Trace that is based on a zero-knowledge proximity verification scheme. Although the implementation demonstrates the feasibility of the proposed approach, the use of zero-knowledge scheme could determine high drain of battery in a real life scenario. In [25] an integrated contact tracing system is proposed to verify, track, and detect new cases of COVID-19.
The proposed system consists of three components: an infection verifier subsystem, a mass surveillance subsystem and a P2P mobile plus a blockchain employed for managing all transactions between the three subsystems. Differently from the other approaches, here the blockchain is used for exchanging information. [26] introduced the use of Diffie–Hellman (DF) secret sharing protocol that allows users to autonomously call each other to notify a detected infection. The main issues with this approach are two: the computational complexity of DF scheme that can affect battery life; the need of at least 256 bit ID that cannot be saved into the BLE identifier.

In order to overcome the limitations of the approached already present in the literature, the proposed approach introduces the use of NFT for granting the tested user positive permission to anonymously publish its contact on the blockchain. Moreover, the publication step can be automatized through the reception of the NFT so that the notification of possible contagion is not demanded to an user action. The list of contacts is save in a compact data structure called XOR filter that allows the anonymous publication of the contacts id. Last but not least, in order to solve the problem of id generation, we introduce the use of HD wallets that can be used to produce a very large number of blockchain address that can be easily inserted into the BLE identifier. Hence contributions can be summarized as follows:

- Use of NFT to uniquely and anonymously represent a positive test.
- Novel notification scheme that exploits NFT issue to grant the user app the permission to public its contacts list.
- Use of XOR filter to public contacts list in a compact and anonymous way.
- Use of HD wallets to generate contact id as blockchain addresses that can be saved into the BLE identifier.

4. Proposed system

In this work, a decentralized system aimed to handle the phases of COVID19 contagion is presented. The system operated in two different contexts:

- Off-chain context, which consist in tracking contacts between users. This part of the system consists of a smartphone application that uses BLE technology, specifically iBeacon protocol, for broadcasting and receiving anonymous identifiers.
- On-chain context, which concern receiving data about infections and notifying users. This part of the system is implemented by smart contracts issued on the blockchain based on the Ethereum Virtual Machine (Ethereum, Hyperledger Burrow).

Fig. 1 shows an overview of the proposed system.

The operation in the two contexts can be divided into three basic phases:

1. Broadcast and receipt of temporary IDs Users generate temporary IDs in the form of Ethereum addresses, with a size of 20 bytes, which are emitted to surrounding devices exploiting to the iBeacon [27] protocol which uses the BLE technology. In particular, the 20 byte of the address are mapped on the 20 bytes of the iBeacon packet composed by the UUID (16 bytes) Major number (2 bytes) and Minor number (2 bytes) fields. At the same time, users listen for temporary IDs, sent from nearby devices, that are stored in a local database.

2. Notification of contagion If a diagnostic test is positive, the infected user receives a token with an unique identifier. The token therefore allows to uniquely represent each positive test and authorize the user to anonymously report his positivity to other users of the system.

In case of positive result, the application has to anonymously publish the addresses used in the timeframe. To this end, we propose the use of XOR Filters [29], a probabilistic data structure to represents a set that allows to add keys and check whether a key belongs to the set. It is possible to have false positives, but not false negatives. An XOR filter stores, for each key, a fingerprint of a fixed number of bits obtained by an hash function $h$. In order to test the membership of a key $y$ to the set, each fingerprint from the data structure is compared with $h(y)$). If $y$ is part of the set, the fingerprints match, otherwise they differ with a probability that depends on the size of the fingerprint.

The representation of a set of $n$ elements is made possible by a list of $32 + (1.23 \times n)$ bits strings. By setting the length of the bits string required to represent a portion of the set to $\approx 20$ bits, a false positive probability of $\approx 10^{-6}$% can be achieved. Further details about the algorithms to construct an XOR filter and test the membership of a given key to the set can be found in [29]. The characteristics of this data structure allow to have a compact representation of a set of addresses, used as IDs, to be published on the blockchain. The execution of smart contracts...
requires an upper limit as regards the size of the data transferred and the operations performed. This limit is represented in Ethereum by the \textit{gasLimit} parameter of a given transaction to a smart contract. The costs for data transfer via transactions are defined as \cite{30}:

- 68 units of gas for each byte other than 0
- 4 units of gas for each byte equal to 0

At the time of this writing, the \textit{gasLimit} for the Ethereum public network is \(\approx 12\) million. Following tests conducted with one of the libraries that implements the XOR Filter \cite{31} we have collected the data present in Table 1, which shows that the current conditions of the Ethereum network would allow the transfer of an XOR Filter. In Section 6 we will discuss the problem of the costs associated with these transactions and how they may not be incurred by system users.

| Number of contacts | XOR filter dimensions (in kB) | gas spent     |
|--------------------|-------------------------------|---------------|
| 1000               | \(\approx 2.6\) kB           | \(\leq 172992\) |
| 2000               | \(\approx 5\) kB             | \(\leq 367200\) |
| 5000               | \(\approx 12\) kB            | \(\leq 842112\) |
| 10000              | \(\approx 24.7\) kB          | \(\leq 1678512\) |

5. Implementation details

In order to verify the operation of the proposed approach, the system has been implemented on a 4 nodes blockchain network employing Hyperledger Burrow \cite{32} and Besu \cite{33} client. Each node has been created in Google Cloud Platform as e2-standard-2 computing engine.
endowed with 2 vCPU and 8 GB memory. The operations carried out on-chain are managed by a set of smart contracts. The structure of this set and the possible operations are described in Fig. 3 In particular:

- **C19Test** is a smart contract that implements the EIP-721 standard. In particular, the C19Test tokens can only be created by the users in the Maintainer group, for the publication of XOR filters.

- **Maintanance Contract**, is a smart contract that implements the operations by the users of the Maintainer group. Users of this group can notify the outcome of a diagnostic test, which corresponds to a unique token. The Maintenance Contract contract is associated with a Storage type contract, implemented by a different smart contract to allow the updating of the single Maintanance Contract. This contract stores the token IDs (associated to an Ethereum address) that have the permissions to report positivity.

- **Report Contract**, is a smart contract that implements the operations that can be performed by the users of the system. Through this contract, users can notify contacts at risk by means of XOR Filter. Report Contract deals with the notification of new infected users by generating events stored in the blockchain, that is publicly inspectable.

### 5.1. Protocol phases

An iOS application has been created in order to test the protocol phases and in particular the exchange of addresses employing BLE/Beacon technology and the receipt notification of the ERC721 token. As stated before, the contact tracing protocol takes place in a distributed way through an application that uses BLE technology to broadcast the addresses generated by an HD Wallet (implemented using web3swift library). The path to generate these addresses is $m/44'1'/Q'1'/0'/0/n$ where $n$ is the index of the generated address. At the same time, the application listens to nearby devices from which it captures the communicated addresses. The latter are stored in a local database for a period useful for the possible detection of a contagion. Because of the use of 20 bytes Ethereum addresses, the maximum number of addresses in the system is $2^{160}$. For instance, if a new address is generated every 10 minutes, each device would use 52560 addresses per year. Although an HD Wallet can handle a much larger number of addresses, we can speculate that every year a new wallet can be created with a new seed phrase, hence generating a completely new sequence of addresses. With this settings, the path to generate contact addresses is $m/44'/0'/0'/0/month[1 − 12]/day[1 − 31]/hour[0 − 23]/[1 − 6]$, that is, for each month, for each day of the month, for each hour of the day, every 10 min the application starts emitting a new address starting from 1 to 6.

When a user undergoes the epidemiological test, he generates and issues to the institution an address corresponding to the path $m/44'1'/1'/0'/0/n$, where $n$ is the index of the address used (one for each test). The address is transferred using a QR code generated by the same application. At this point, the user waits for the test result, which is communicated by sending an ERC721 token. The event is produced through a call to the Maintenance Contract by a system maintainer. In particular, in the case of a positive result, a maintainer calls the method of the contract Maintenance Contract corresponding to the notification of a positive result by issuing a token with a unique ID. The contract places the ID of the token in the list of the IDs authorized to communicate the addresses used for contact tracing. At this point, the user receives a notification of the issued token and has the possibility to send it to the Report Contract using an XOR Filter $f$ locally computed by the mobile application. $f$ represents all the addresses used by
the user in the time window considered for the possible contagion. According to construction performance presented in [29] the time needed to compute the XOR filter for 2000 addresses is in the order of 0.5 ms. Upon the reception of the filter, the Report Contract generates a serialized event as \((\text{timestamp, filter})\), which is received by the devices participating in the system. The devices must then rebuild and interrogate the XOR Filter to verify the presence of received IDs in the considered time interval. In case of a hit (an address can be found in filter) the user receives a notification through the application. As shown in Table 1, from the tests conducted employing an open source implementation of the XOR filters, it is possible to observe that their size is quite small, resulting in a low communication overhead. In particular, considering a device that generates 2000 addresses in a time window of 14 days (that roughly corresponds to a new address every 10 min), a 5 kB XOR filter has to be transmitted from the device of the user tested positive, while considering 10000 contagions, 50 MB of XOR filters should be downloaded on the device. Moreover, XOR filters are smaller and faster [34] than Bloom filters used in [16,35].

6. Discussion

In this section we will discuss some aspects related to the choice of the blockchain, the security and performance of the proposed protocol.

6.1. Blockchain model

The model defined in this document is based on the use of a blockchain system. The latter could be the main Ethereum network or a public permissioned instance. Several Ethereum-based projects allow the implementation of a permissioned blockchain system (Hyperledger Burrow [32], Besu [33]). In any case, the use of smart contracts for the collection and notification of new infections raises the need to distribute native currency of the blockchain system for the execution of transactions towards the system’s smart contracts, making difficult to join the system without the necessary funds. It is possible to overcome this limit through the use of GSN (Gas Station Network) [36]. The use of this system allows you to delegate the payment of transactions to a third party (called Relayer) with a decentralized solution that allows protocol participants to control the submission of transactions to the network.

6.2. Privacy and security analysis

As far as possible threats to user privacy are concerned, attacks on a tracking system mainly aim at the user’s de-anonymisation or at the tracking of the user’s meetings. All the proposed and analyzed attacks involve a malicious actor who has the economic and technical capabilities to carry out the attacks. The characteristics of the proposed protocol allow to solve some of these vulnerabilities. Attacks to which a contact tracing system is vulnerable, often involve collusion between a malicious actor and the central server. In the proposed system such collusion is not feasible due to the absence of a central server. Other types of attacks could include the possibility for an attacker to produce a malicious report once it has the necessary credentials to communicate with the server. In the proposed model, such attacks are made unlikely by the attacker’s need to be in possession of a positivity token for reporting such information. The token, required for activate the smart contract that takes care of collecting the filters, is impossible to replicate or falsify since the reliable instance (the Ethereum address) of this token is stored in the Maintenance Contract. In case of collusion of an attacker who intends to falsify the positivity token with the Mainmaintainer, the change of this instance could not take place in a secret way since the transactions are publicly available. However, through the use of HD wallet generated addresses, the proposed protocol is vulnerable to the class of attacks known as reply attacks [37] in which an attacker could collect and replicate addresses captured by a device in charge of the operation, generating contacts that never occurred. However, in the BLE frame there is no field that identifies the specific application that produced it, so any attacker would be forced to collect a large amount of data that is not necessarily linked to a contact and therefore would have no way of selecting valid IDs (in our case, mapping the 20 bytes of the Ethereum address to the 20 bytes of the iBeacon frame, makes impossible to identify the application from the UUID field).

In [38] some of the attacks made possible by an attacker with these abilities are described and analyzed. In particular, the attacker has as its objective the de-anonymization of a user (or a group of them) and the surveillance of the same. Some of the attacks analyzed require collusion between the server and the attacker himself. In the proposed model, collusion with the server is made impossible by the absence of the server. The only possible collusion for the attacker is with the health authority which could allow him to produce fake reports (discussed below). In another possible attack, an attacker could collect the addresses generated by users in order to track their meetings. The proposed system is protected against this kind of attack for the following reasons: first, BLE advertisements do not use predefined UUID so that an attacker cannot easily determine which ones are related to the contact tracing app; second, contact data are stored in XOR filters hence they are not public visible (addresses cannot be extracted from the XOR filters); third, an infected user receives the NFT on the address that only the authority deputed to test infection owns and this same address is used to send the XOR filter containing contact data to the blockchain. An attacker could listen for transactions communicated to the blockchain and analyze the IP addresses of users who report their contagion. This analysis cannot be mitigated through the use of dummy packets or similar techniques since the attacker could still analyze all the packets and select the packets that correspond to a transaction registered at a node of the blockchain system. Nevertheless, this attack can be mitigated by using the GSN network or other levels of obfuscation between users and the blockchain network. These additional levels could delay the communication of the transaction signed by the user to a node of the blockchain network but not in such a way as to make the use of the system itself ineffective. An attacker in collusion with the health authority or with an infected user could report on the blockchain a set of addresses chosen by the attacker himself that will later be alarmed. In the case of collusion with the health authority, the attacker could also produce tokens that signal the positivity of a user. The mitigation of this attack could consist in requiring more cryptographic signatures to produce a token and thus making the Maintainer account a multisig account.

6.3. Performance analysis

In order to evaluate the performance of the proposed system, the Besu client configuration showed in Table 2 has been employed.

With these settings, the throughput of the proposed system is \(\sim 200\) TPS with a network load of 320 TPS and a latency of \(\sim 2.5\) seconds. For what concerns the number of transactions required in case of contagion, considering 10000 daily positive tests the following transactions would be necessary: 10000 transactions for transferring NFT; 10000 transactions for uploading XOR filters for a total of 50MB (5kB each). The critical aspects concerns the number of transactions needed to download the XOR filters and test them for possible contacts.

| Table 2 | Besu configuration. |
|---------|---------------------|
| Contract visibility | Public |
| Number of peers | 4 |
| Block time | 1 second |
| Block gas limit | 41943040 |
| Transaction pool limit | 4096 |
| Consensus protocol | IBFT 2.0 |
In fact, assuming a population of \( \sim 60000000 \) people, in the worst case there would be an equal number of transactions on the blockchain resulting in increased latency. A possible solution to overcome this limitation, is to offload all the XOR filters on the IPFS [20]. Although comparing contact tracing systems is very difficult due to different implementations and blockchain settings, in order to provide more information about the performance of the proposed approach, Table 3 shows a comparison with a recent approach found in literature [35]. The comparison has been performed considering the throughput and latency in the token issuing phase and throughput and latency during the filter transfer. The performance are almost equal for transferring the tokens whereas the proposed system performs better, both in terms of throughput and latency, because of the reduce size of XOR filters w.r.t. Bloom filters.

### 7. Conclusions

A contact tracing protocol is presented that exploits the decentralized nature of blockchain technology and the possibility of automating operations through the use of smart contracts for reporting contacts at risk of contagion. The proposed solution guarantees the privacy of system users and is equipped with a notification mechanism based on ERC721 tokens. The non-fungible nature of these tokens guarantees the impossibility of producing false notifications and also guarantees the unrepeatable outcome of the diagnostic test. The latter is therefore permanently recorded on the blockchain, in the form of sending an ERC721 token, allowing anyone with reading access to the recorded data to be able to monitor the number and progress of infections. Furthermore, the proposed system, using the iBeacon standard and the blockchain address format, can be implemented without the support of the ID generation functions provided by the manufacturers of the mobile operating systems, representing in fact an independent choice of the ID generation functions provided by the manufacturers of the mobile operating systems.

The performance activity currently concerns the improvement of the proposed protocol regarding the robustness with respect to replay attacks. Furthermore, the possibility of inserting the proposed system in a wider context of public blockchain on which it is possible to implement various services for citizens implemented via smart contracts is being studied.

### CRediT authorship contribution statement

**Alessio Ferone:** Conceptualization, Methodology, Software, Writing – review & editing. **Antonio Della Porta:** Software, Validation, Writing.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### References

[1] B. Sowmiya, V.S. Abhijith, S. Sudersan, R. Saktij Jaya Sundar, M. Thangavel, P. Varalakshmi, A survey on security and privacy issues in contact tracing application of Covid-19, SN Comput. Sci. 2 (3) (2021) 1-6.

[2] M. Hoskam, A. Lappetelaan, K. Rivekas, Low end extension for bluetooth, in: Preetendings, 2004 IEEE Radio and Wireless Conference, IEEE Cat. No.04TH8746, 2004, pp. 199–202.

[3] L. Ricci, D.D.F. Maes, A. Favenza, E. Ferro, Blockchains for COVID-19 contact tracing and vaccine support: A systematic review, IEEE Access 9 (2021) 37956–37960.

[4] C. Troncoso, M. Payer, J.-P. Hubaux, M. Salathé, J. Larus, E. Bugnion, W. Lukes, T. Stadler, A. Pyrgelis, D. Antonioli, et al., Decentralized privacy-preserving proximity tracing, 2020, arXiv preprint arXiv:2005.12273.

[5] Apple, Google, Exposure notification, 2020, https://www.google.com/covid19/exposurenotifications.

[6] G. Garibotto, P. Murrieri, A. Capra, S. De Muro, U. Petillo, F. Flammini, M. Esposito, C. Pragliola, G. Di Leo, R. Lengu, N. Mazzino, A. Paolillo, M. D’Urso, R. Vertucci, F. Narducci, S. Ricciardi, A. Casanova, G. Fenu, M. De Mizio, M. Savastano, M. Di Capua, A. Ferone, White paper on industrial applications of computer vision and pattern recognition, in: A. Petrosino (Ed.), Image Analysis and Processing –ICIAP 2013, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 721–730.

[7] A.I. Sanka, M. Irfan, I. Huang, R.C. Cheung, A survey of breck-trade in blockchain technology: Adoptions, applications, challenges and future research, Comput. Commun. 169 (2021) 179–201.

[8] K. Mohammad Hosseini, M.E. Esmaili, T. Dargahi, A. Khonsari, M. Conti, BCHealth: A novel blockchain-based privacy-preserving architecture for IoT healthcare applications, Comput. Commun. 180 (2021) 31–47.

[9] R.M. Amir Latif, K. Hussain, N.Z. Jhanji, A. Nayyar, O. Rizwan, A remix IDE: smart contract-based framework for the healthcare sector by using blockchain technology, Multimedia Tools Appl. (2020).

[10] A.M. Don Johnson, S. Vanstone, The elliptic curve digital signature algorithm (ECDSA), Int. J. Inf. Secur. 1 (2001) 36–63.

[11] F. Vogelsteller, V. Buterin, ERC20 Token Standard (EIP 20), Ethereum. Org, Ethereum Improvement Proposals, 2015.

[12] W. Estrinhen, D. Shirley, J. Evans, N. Sachs, ERC721 Non-Fungible Token Standard (EIP 721), Ethereum. Org, Ethereum Improvement Proposals, 2018.

[13] C. Castelluccia, N. Bievela, A. Bouet, M. Cunche, C. Lauraudoux, D. Le Métayer, V. Rocca, ROBERT: Robust and privacy-preserving proximity tracing, 2020.

[14] Immuni - Official website, https://www.immuni.it/.

[15] A. Akinbi, M. Forshaw, V. Blinkhorn, Contact tracing apps for the COVID-19 pandemic: a systematic literature review of challenges and future directions for neo-liberal societies, Health Inf. Sci. Syst. 9 (1) (2021) 18.

[16] C. Zhang, C. Xu, K. Sharif, L. Zhu, Privacy-preserving contact tracing in 5G-integrated and blockchain-based medical applications, Comput. Stand. Interfaces 77 (2021) 103520.

[17] M.C. Chang, D. Park, How can blockchain help people in the event of pandemics such as the COVID-19? J. Med. Syst. 44 (5) (2020) 102.

[18] S.M. Ideyes, M. Nowostawski, R. Jameel, Blockchain-based digital contact tracing apps for COVID-19 pandemic management: Issues, challenges, solutions, and future directions, JMIR Med. Inform. 9 (2) (2021) e25245.

[19] E. Bandara, X. Liang, P. Foytik, S. Shetty, C. Hall, D. Bowden, N. Ranasinghe, K. De Zoysa, A blockchain empowered and privacy preserving digital contact tracing platform, Inf. Process. Manage. 58 (4) (2021) 102572.

[20] N. Bari, U. Qamar, A. Khalid, Efficient contact tracing for pandemics using blockchain, Inf. Med. Unlocked 26 (2021) 100742.

[21] X. Wang, W. Ni, X. Zhu, G. Yu, R.P. Liu, N. Georgalas, A. Reeves, Capacity analysis of public blockchain, Comput. Commun. 177 (2021) 112–124.

[22] H. Xu, L. Zhang, O. Onireti, Y. Fang, W.J. Buchanan, M.A. Imran, BeeTrace: Blockchain-enabled privacy-preserving contact tracing for COVID-19 pandemic and beyond, IEEE Internet Things J. 8 (5) (2021) 3915–3929.

[23] Z. Hee, I. Salam, Blockchain based contact tracing: A solution using bluetooth and sound waves for proximity detection, 2022, Cryptology ePrint Archive, Report 2022/209, https://ia.cr/2022/209.

[24] Z. Feng, C. Xu, H. Wang, J. Huang, J. Xu, X. Chu, P<sup>&lt;0.001</sup> in: Proceedings of the 2021 International Conference on Management of Data, Association for Computing Machinery, New York, NY, USA, 2021, pp. 2589-2593.

### Table 3

| Method          | Throughput token | Latency token | Throughput filter | Latency filter |
|-----------------|------------------|---------------|------------------|---------------|
| DIMY [35]       | 50               | ~ 1 sec       | ~ 10             | ~ 25          |
| Proposed system | 50               | ~ 1 sec       | ~ 25             | ~ 5 sec       |
[25] M. Torky, E. Goda, V. Snasel, A.E. Hassanien, COVID-19 contact tracing and detection-based on blockchain technology, Informatics 8 (4) (2021).

[26] G. Avitabile, V. Botta, V. Iovino, I. Visconti, Towards defeating mass surveillance and SARS-CoV-2: The Pronto-C2 fully decentralized automatic contact tracing system, IACR Cryptol. EPrint Arch. 2020 (2020) 493.

[27] X.-Y. Lin, T.-W. Ho, C.-C. Fang, Z.-S. Yen, B.-J. Yang, F. Lai, A mobile indoor positioning system based on ibeacon technology, in: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC, IEEE, 2015, pp. 4970–4973.

[28] P. Wuille, Hierarchical deterministic wallets, vol. 1, 2013, p. K2, Online Specification for BIP32 At https://Github.com/Bitcoin/Bips/Blob/Master/Bip-0032. MediaWiki Time: Time K.

[29] T.M. Graf, D. Lemire, Xor filters: Faster and smaller than bloom and cuckoo filters, 2019, CoRR abs/1912.08258.

[30] G. Wood, et al., Ethereum: A secure decentralised generalised transaction ledger, Ethereum Proj. Yellow Pap. 151 (2014) (2014) 1–32.

[31] Header-only Xor Filter library, https://github.com/FastFilter/xor_singleheader.

[32] Hyperledger Burrow, https://www.hyperledger.org/use/hyperledger-burrow.

[33] Hyperledger Besu, https://besu.hyperledger.org/en/stable/.

[34] T.M. Graf, D. Lemire, Xor filters, ACM J. Exper. Algorithm. 25 (2020) 1–16.

[35] N. Ahmed, R.A. Michelin, W. Xue, G.D. Putra, S. Ruj, S.S. Kanhere, S. Jha, DIMY: Enabling privacy-preserving contact tracing, J. Netw. Comput. Appl. 202 (2022) 103356.

[36] OpenGSN Network, https://opengsn.org/.

[37] S. Vaudenay, Analysis of DP3T., IACR Cryptol. EPrint Arch. 2020 (2020) 399.

[38] G. Avitabile, V. Botta, V. Iovino, I. Visconti, Towards defeating mass surveillance and SARS-CoV-2: The Pronto-C2 fully decentralized automatic contact tracing system, IACR Cryptol. EPrint Arch. 2020 (2020) 493.

[39] M. Vukolic, On the interoperability of decentralized exposure notification systems, 2020, arXiv preprint arXiv:2006.13087.