applied sciences

MDPI

Article

A WSN Framework for Privacy Aware Indoor Location

Aleksandar Tošić [1,2,*,†,‡], Niki Hrovatin [1,2] and Jernej Vičič [1,‡]

[1] Faculty of Mathematics, Natural Sciences and Information Technologies, University of Primorska, 6000 Koper, Slovenia; niki.hrovatin@innorenew.eu (N.H.); jernej.vicic@upr.si (J.V.)
[2] InnoRenew CoE, Livade 6, 6310 Izola, Slovenia
* Correspondence: aleksandar.tosic@upr.si
† Current address: Glagoljaška 8, 6000 Koper, Slovenia.
‡ These authors contributed equally to this work.

Abstract: In the past two decades, technological advancements in smart devices, IoT, and smart sensors have paved the way towards numerous implementations of indoor location systems. Indoor location has many important applications in numerous fields, including structural engineering, behavioral studies, health monitoring, etc. However, with the recent COVID-19 pandemic, indoor location systems have gained considerable attention for detecting violations in physical distancing requirements and monitoring restrictions on occupant capacity. However, existing systems that rely on wearable devices, cameras, or sound signal analysis are intrusive and often violate privacy. In this research, we propose a new framework for indoor location. We present an innovative, non-intrusive implementation of indoor location based on wireless sensor networks. Further, we introduce a new protocol for querying and performing computations in wireless sensor networks (WSNs) that preserves sensor network anonymity and obfuscates computation by using onion routing. We also consider the single point of failure (SPOF) of sink nodes in WSNs and substitute them with a blockchain-based application through smart contracts. Our set of smart contracts is able to build the onion data structure and store the results of computation. Finally, a role-based access control contract is used to secure access to the system.

Keywords: WSN; indoor location; privacy; blockchain; COVID-19

Citation: Tošić, A.; Hrovatin, N.; Vičič, J. A WSN Framework for Privacy Aware Indoor Location. *Appl. Sci.* **2022**, *12*, 3204. https://doi.org/10.3390/app12063204

Academic Editors: Asadullah Shaikh, Uffe Kock Wiil, Yousef Asiri and Agostino Forestiero

Received: 20 December 2021
Accepted: 11 March 2022
Published: 21 March 2022

1. Introduction

We have recently witnessed the coronavirus disease 2019 (COVID-19) outbreak caused by the Severe Acute Respiratory Syndrome Coronavirus-2 (SARS-CoV-2).At the time this manuscript was written, SARS-CoV-2 was still spreading and affecting billions of lives globally [1]. It is now well established from a variety of studies that the SARS-CoV-2 primary infection vectors are the respiratory droplets of infected people produced by coughing, sneezing, or talking [2–4]. Therefore, the rapid spread is driven by the social aspects of everyday life, which in recent days have been altered by the guidelines for preventing infection spread, such as the mandatory use of masks, cleaning and disinfection, and the introduction of social distancing. Respecting the mentioned guidelines is of particular concern in public buildings where multiple people share the same space, and the infection spread could endanger not only individuals but also halt the operations of organizations. Moreover, in confined spaces, the probability of infection is higher than outdoors since infection transmission is dependent on ventilation [4].

The role of IoT (Internet of Things) to prevent the spreading of the COVID-19 disease has already been discussed in [5–8], which conceptualize frameworks for monitoring the spread of the COVID-19 disease through heterogeneous sensor technology and apply data-driven inferences to forecast new outbreaks and predict virus mutations. However, the mentioned literature barely discusses privacy concerns and only recent studies [9] are deliberating over the privacy aspect of integrating such monitoring solutions in everyday

life. Even though encryption effectively provides data privacy, monitoring indoor activities by relying on wireless IoT devices could disclose contextual information on data transmission [10,11], not only posing risks to the privacy of individuals, but also compromising building security.

This has motivated us to extend the research on our privacy-aware IoT and blockchain-based indoor location system to counter the spread of the COVID-19 disease. The presented indoor location system is particularly suitable for application in medical facilities, public buildings, and residential homes as a framework for privacy-aware indoor location monitoring. The proposed solution could be applied for structural health monitoring, studying behavioral patterns of a building's occupants and health-related issues such as locating lost patients with memory and orientation disorders, fall detection, and also identifying violations of social distancing, counting the number of persons in a room, and determining when and which room needs surface disinfection due to over-utilization, etc.

The key contributions of the privacy-preserving framework are:

- A novel privacy-preserving indoor location system with querying capabilities: The network of sensors is embedded in the floor and senses the local force applied over it. It is non-intrusive and does not require active user interaction. Moreover, the raw sensory data collected by sensors describe the force applied to the floor and can only lead to unique user identification via walking gait analysis. However, the walking gait analysis [12] requires large amounts of data from individual users, and in our privacy-aware framework, the raw data do not leave the source sensor, therefore inhibiting similar attempts.
- A secure WSN with anonymous source location and sensor network identity: We propose a new querying protocol for WSN, which uses multi-layer encryption to conceal the network identity of sensor nodes, obfuscating the computation described in [13]. The protocol relies on particular messages similar to those used in the onion routing [14] to convey edge data processing information to sensor nodes and privately retrieve data.
- A blockchain-based fault tolerant indoor location system with no single point of failure(SPOF): We address the fault tolerance shortcomings of sink nodes [15] in traditional WSNs by substituting it with a smart contract, which handles the processing of queries, and storing the results. A decentralized role-based access control (RBAC) contract provides user access authorization to monitor individual building spaces defining privacy boundaries and further improves the security over traditional centralized approaches.

The remainder of the paper is structured as follows: In Section 2, we present the relevant literature. Section 3 highlights the core features of the proposed solution. In Sections 4 and 4.1, we present our onion route protocol and filtering. In Section 5, we detail how blockchain smart contracts can replace sink nodes. In Section 6, we provide the validation of the proposed framework, and finally give final remarks in Section 7.

2. Literature Review

Indoor real-time locating systems (RTLS) have been gaining relevance due to the widespread advances of devices and technologies and the necessity of location-based services. The interest of the mobile industry to accelerate the adoption of indoor position solutions turned into the foundation of the InLocation Alliance (ILA (InLocation Alliance): inlocationalliance.org, accessed on 19 December 2021). The goal of this alliance is to facilitate a rapid market adoption so that new business streams are opened up with context-aware applications in indoor environments. The ILA chose Wi-Fi and Bluetooth as their preferred technologies. Both proposed technologies require specialized apps on the mobile devices in order to produce satisfactory results [16].

A thorough and contemporary survey of the Indoor Positioning Systems (IPS) for IoT is presented in [17]; it presents indoor positioning concepts and a list of already used criteria that define IPS for IoT. Brena et al. [16] provide a classification of Indoor Positioning Systems

(IPS), basing the classification on a set of papers comparing different IPS approaches. This is a list of identified technologies: Infrared mobile reader, Infrared (IR), laser (passive), ultrasound passive, audible sound, magnetic, RFID mobile tag, RFID mobile reader, Wi-Fi, Bluetooth, ZigBee, UWB, tomographic technology (water resonance), camera infrastructure, cameras (portable), floor tiles, air pressure, inertial, ambient light, artificial light, indoor AGPS, cellular technology, TV, and FM. All the technologies that need any intervention from the user are out of the scope of this experiment, so all technologies based on wearables, which demand the installation of software on mobile devices or the users to act in a certain way, are out of the scope of the paper. All technologies based on audible and visible changes in the environment (such as the usage of fluorescent lighting) pose a distraction. Additionally, the use of video cameras and microphones presents a huge privacy concern and were thus eliminated from this study. Most IR systems require line-of-sight (LOS) clearance from the emitter to the sensor; in the context of IR IPS systems, the requirement of LOS clearance is a great disadvantage, as it suffers from no-detection areas, and the system performance is also affected by sunlight [18].

A metaheuristic for anomaly detection in IoT is proposed in [19], which is an extension of the work presented in [20]. The method is based on an activity footprints-based method to detect anomalies in IoT, but with small changes it can be used to track indoor activity.

The technology that "survived" the criteria posed by the presented study was "intelligent tiles", usually using pressure sensors. There has been some research in the area of employing pressure sensors to track the users' indoor behaviors, ranging from person tracking and indoor localization to fall prediction. The Smart Floor project at Georgia Institute of Technology [21] and ORL Active Floor at The Olivetti and Oracle Research Laboratory [21] provide location and identification without encumbering the users, but their highest levels of precision will not be reached until the user steps on the exact centers of the floor tiles, which for a reliable measurement would require conscious attention. Chan et al. [22] present a smart-sensored floor setting that draws energy to power the motion sensors from the integrated generators that are powered by normal floor activity such as walking or sport activity. Kaddoura et al. [23] present a cost-effective intelligent floor setting using pressure-sensing sensors that functionally competes with higher-cost systems. Shen and Shin [24] report on the development of a distributed sensing floor using an optical fiber sensor. However, all presented intelligent floor systems fail to properly address the privacy and data-sensitivity issues.

Privacy preservation in location systems has already been addressed in some works, although in different domains, such as [25], which proposes a location privacy method based on k-anonymity, and [26], which uses blockchain to achieve the desired behavior.

Cumulative pressure sensors [27] for large areas have been proposed to present a rough estimate of the number of persons present in a designated area (effectively measuring/counting the occupancy of a room). This technology is only useful for counting the number of occupants in the observed are; it lacks all the other IPS properties.

Google and Apple have jointly developed an exposure notification system (https://www.google.com/covid19/exposurenotifications/, accessed on 19 December 2021) based on a shared sense of responsibility to help the global community fight the pandemic by keeping track of contact. In the background, users' phones and surrounding phones share randomly generated privacy IDs via Bluetooth. Routinely, the application checks if some of the IDs that the phone has been exposed to have a "compromised" ID, the IDs of owners who have anonymously proclaimed to be infected. The exposure notification system does not monitor users' locations; Google, Apple and other users cannot see users' identities; and the data are only available to the public health authorities. This system does not address the same issues as the system proposed in this paper as the proposed system cannot be utilized as a substitute of the Google/Apples solution for the lack of a backward loop (the information of the infection case cannot be linked to the pseudo-anonymous identities used in our system).

Tošić et al. [28] present a non-intrusive fall detection solution based on a smart floor, which this paper extends to an indoor location system. The system enables a non-intrusive (with no need for special applications based on wearable devices, smartphones or any other devices) indoor location system with additional privacy preserving properties such as anonymity and sensor location/network anonymity. We achieved this by using the smart floor, coupled with onion routing for source location anonymity and blockchain for the final sink personal pseudo-anonymity.

2.1. Secure Data Processing in Network of Sensors

The data sourcing from a network of sensors is usually processed in a system external to the network; the processing system is often a cloud service. Solutions such as Transport Layer Security (TLS) are applied to provide a secure data transfer from sensor nodes to the data processing system. However, even though TLS solutions ensure data confidentiality, a number of studies [29–31] show that it is possible to associate TLS traffic patterns with activities monitored by the network of sensors.

The technique of Compressive Sampling (CS) found application in WSNs to severely reduce the sending data size by representing the data using a smaller number of samples than dictated by the Nyquist theorem [32,33]. Furthermore, the CS was not applied only to reduce the communication overhead but also to provide data confidentiality by changing the CS coefficients at each transmission by relying on a secure seed at sensor nodes [34]. In [35], the authors propose a CS data-gathering scheme that provides data confidentiality and protection against traffic analysis via the use of public-key Homomorphic Encryption [36] to secure the transmitted data [35]. However, in CS techniques, the data recipient can reconstruct and identify the data from individual nodes, and therefore, it is an appealing target for attackers since, if compromised, it could disclose the private data of several nodes. Moreover, CS requires that the data recipient node solves a linear programming equation to recover the original data; therefore, the computation load is introduced and does not take advantage of the processing power of nodes forming the sensor network.

Numerous studies [37,38] have focused on preserving sensor network privacy by aggregating data as they flow through the network. The technique is dubbed as in-network data aggregation and relies on aggregator nodes that aggregate the data from multiple sensor network nodes; however, it does this without the possibility for the aggregator node to disclose the private data of individual nodes. The survey [37] provides a classification of privacy-preserving data aggregation techniques, categorizing and describing them.

Even though privacy-preserving data aggregation could preserve the data privacy of individual nodes, the current solution only allows computing aggregates such as SUM, MAX, AVG, variance, etc. The mentioned aggregates could provide an overview of the monitored environment; however, they are not sufficiently descriptive for indoor location requirements. In this study, we propose a data acquisition layer based on the General Purpose Data and Query Privacy Preserving Protocol described in [13]. This technique allows the retrieval of arbitrary aggregated data without disclosing which nodes contribute to the data retrieval. The generated network traffic is uniform due to randomized paths and the sojourn time, therefore preventing traffic analysis attacks. The computing power of sensor nodes is utilized for data processing in situ. Moreover, in the present contribution, we present a technique coupled to a blockchain solution to secure query creation, ensure that only the message origin knows nodes contributing to the data retrieval, and eliminate the aggregator/sink node SPOF.

2.2. Role Based Access Control—RBAC

Traditional IoT access control schemes are mainly built on top of the well-known access control models, including the role-based access control model (RBAC) [39,40], the attribute-based access control model (ABAC) [41], and the capability-based access control model (CapBAC) [42]. In the RBAC-based schemes, the access control is based on the roles (e.g., administer and guest) of the subject. RBAC oversees the user role assignment and permission

assignment. Three implementations currently exist in the form of smart contracts for the Ethereum network [43]: RBAC-SC [44], Smart policies and OpenZepplin contracts (OpenZepplin contracts: https://github.com/OpenZeppelin/openzeppelin-contracts, accessed on 19 December 2021). The blockchain and RBAC service were used as an off-the-shelf service providing the necessary functionality and the scope of the paper does not support any analysis on the comparable properties of the presented solutions.

3. Architecture

Our framework makes use of three main innovations to implement unique properties, which we rely upon to address the limitations of existing indoor location systems. The architecture encompasses these as modules such that it allows interoperability between them to achieve an additive effect of their unique properties. In our implementation, we design a unique cost-effective passive indoor location system that relies on off-the-shelf sensors embedded in an additional layer between the tiling described in more detail in Section 3.2. At the local level, the sensors form a WSN which reduces the complexity of the large-scale implementations. The security and network anonymity [45] concerns are addressed by a specially designed computational model that relies on onion-routing [46] messages for network anonymity, and a general-purpose obfuscated computing model. By using multi-layer encryption and onion routing, nodes are able to collaborate in federated and distributed computations without ever revealing what the global computation is, nor the origin of the computation; further details can be found in Section 4. In the third module, we further improve the security and reliability of the solution by decentralizing the system to introduce much-needed fault tolerance, and secure the entire solution against a single point of failure (SPOF). By using blockchain, we are able to replace sink nodes with smart contracts. We implement an access control module that protects the underlying WSN against unauthorized queries, further detailed in Section 5.

In our vision, different deployments of indoor location systems have different requirements, ranging from personal home deployments (smart home) to health providers (hospitals, homes for older adults, clinics, nursing homes, etc.), and public buildings (municipalities, government buildings, etc.) illustrated in Figure 1. Using a global blockchain network, which stakeholders can participate in, we can inherit the same security level on all of the underlying sensor deployments.

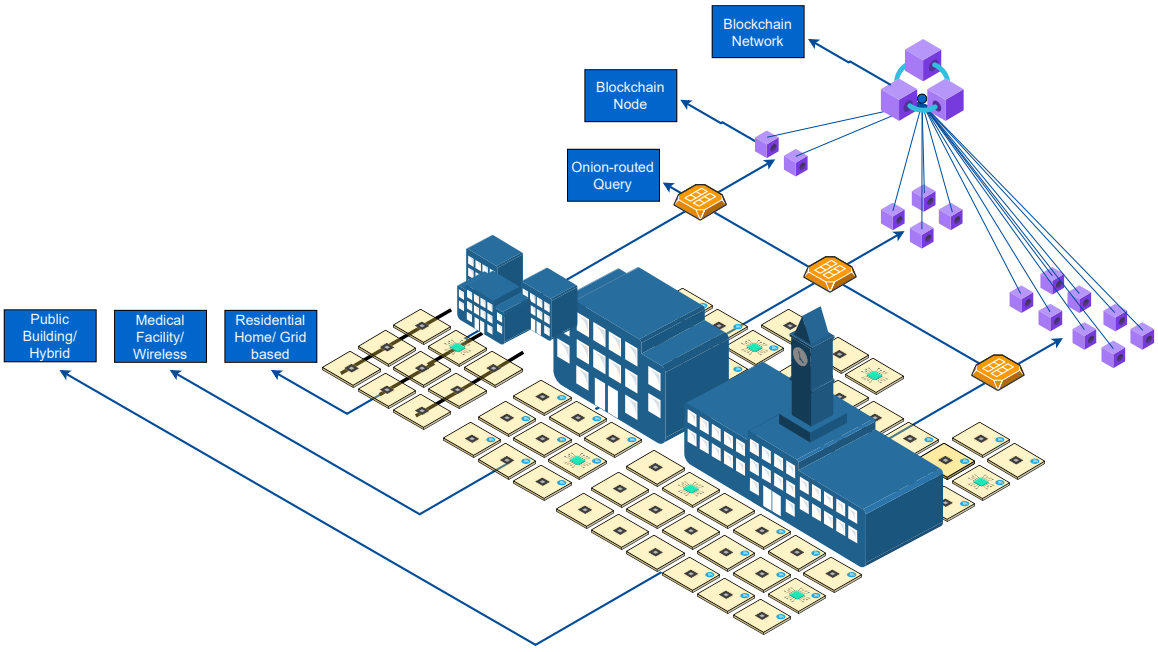

Figure 1. High level view of the presented architecture.

3.1. Cost Aspects of the Proposed Solution

The proposed solution was implemented with cost effectiveness as one of the most important factors. The retail value of the ICT hardware employed in the solution should not exceed USD 100 per square meter (10.76 square feet). One square meter would occupy nine tiles, around USD 20 for the controller and less than USD 80 for the nine pressure sensors. The solution scales linearly with no additional cost.

3.2. Non-Intrusive, Privacy-Preserving Indoor Location

Indoor location has many applications for structural health monitoring, studying the behavioral patterns of a building's occupants and health-related issues such as locating lost patients with memory and orientation disorders, healthy activities, etc. Most existing solutions for indoor location rely on wearable devices (i.e., location-aware bracelets), which require frequent charging and can generate invalid data in case the device is forgotten. Our approach is a passive system that does not require any maintenance or wearable device. We used off-the-shelf force resistors (FSR model 406), which are emended and centered inside a 30 $\times$ 30 cm tile of foam. Once force is applied, the foam and FSR deform, which can be measured as a voltage drop by the controller.

In a wired setting, each tile of foam is shaped like a puzzle piece, which ensures easy assembly. Each tile has two connectors on each face of the square to seamlessly connect to a neighbouring tile. It also includes a small chip for converting the analog signal to a digital that finally allows the collection of sensor readings over a one-wire type protocol. Every three-by-three grid of tiles contains one compute unit, which serves as a controller for the underlying sensors, and a WSN/blockchain node, as depicted in Figures 2–4. Figure 5 illustrates an assembled module in grid mode.

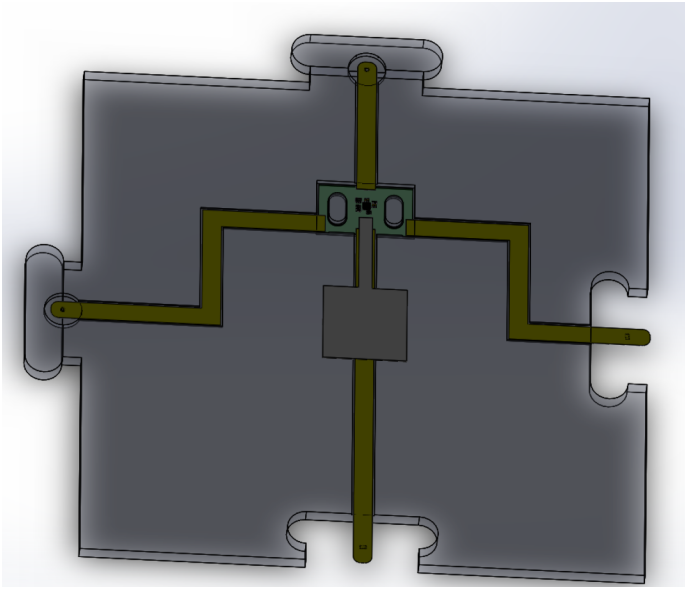

Figure 2. Bottom side of the foam tile.

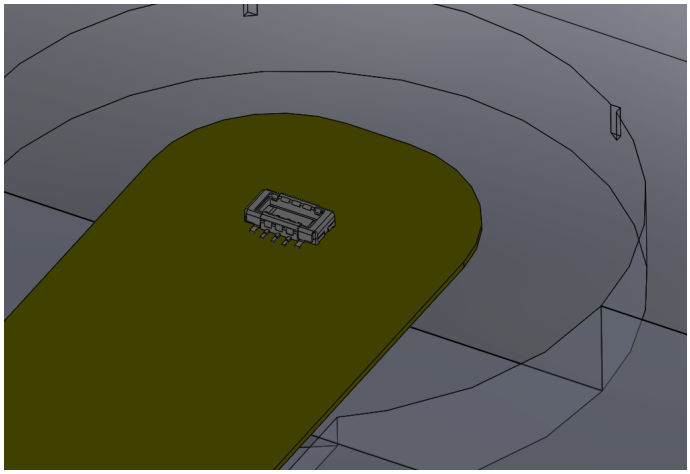

Figure 3. A detailed view of the male connector.

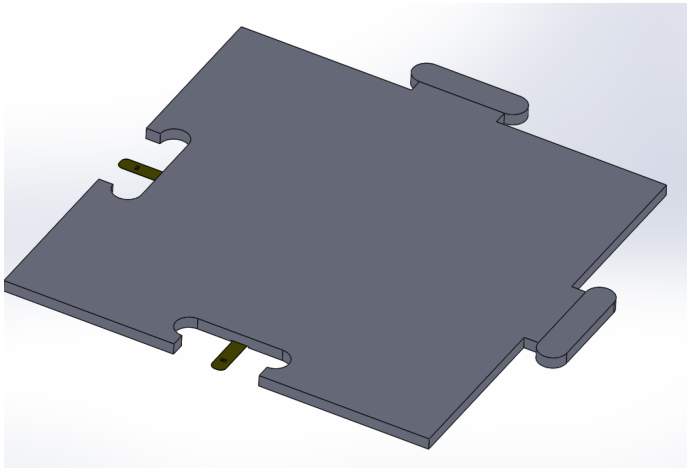

Figure 4. Upper side of the foam tile.

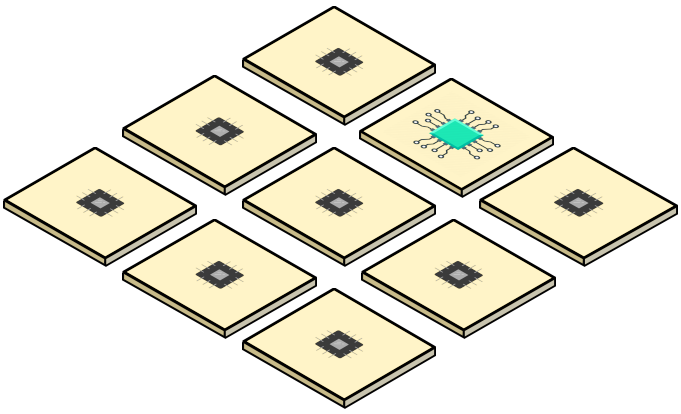

Figure 5. Grid-based connection of individual force sensors.

If physical connections are not suitable, a completely wireless configuration is possible but less cost effective. Figure 6 illustrates a module in full WSN mode.

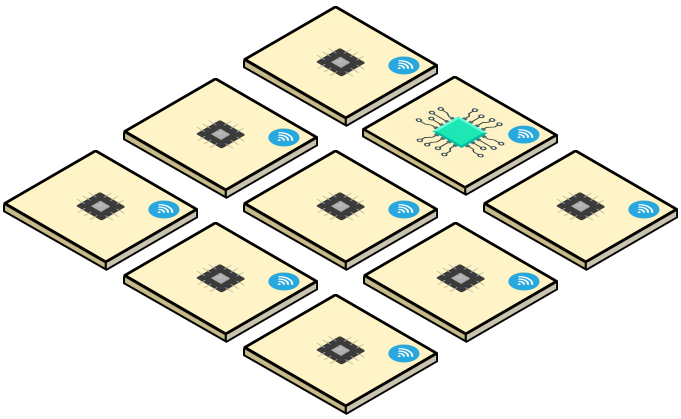

Figure 6. Wireless-enabled force sensors.

4. Secure, and Private Data Filtering, and Aggregation

In this section, we present the data acquisition layer, consisting of the General Purpose Data and Query Privacy Preserving Protocol described in [13]. The communication protocol presented in [13] is characterized by messages containing a layered object made of several encryption layers similar to the one employed in the onion routing [46]. Each layer of the layered object contains the IP address of the next receiver of the message, and since layers are produced using public key cryptography [47], the message must travel through the exact sequence of nodes defined at message construction. The technique of encoding the message path in the message is commonly known as source routing [48]. Path information is carried in encryption layers to restrict the knowledge obtained by nodes processing the message, which only learn about the sender and the next receiver of the message. Therefore, the whole message path is not revealed to any node receiving the message.

In addition to the layered object, messages specified by the communication protocol in [13] include a payload. The payload consists of computer code specified in a general-purpose programming language and a binary string that stores an aggregate. Therefore, it includes instructions specifying the data to retrieve and the aggregated data of nodes in the message path. In the following, we will refer to the onion message (OM) as a structure consisting of the layered object and the aforementioned payload. The OM payload is secured by symmetric key encryption to prevent malicious actors from tracking the OM and obtain values added by sensor nodes by comparing the aggregate pre and post OM processing. Moreover, encryption keys required to decipher the OM payload are delivered only to specific nodes in the OM path by enclosing symmetric encryption keys in the layered object. Nodes in the OM path are either: (a) processing the OM or (b) emulating OM processing.

(a) Nodes processing the OM obtain two symmetric encryption keys and the next-hop IP address from layer decryption of the layered object. The first symmetric encryption key is used to access the content of the OM payload. Next, the node executes the computer code and embeds results in the binary string. The OM payload is then encrypted using the second symmetric encryption key, and after a time-span affected by randomness, the OM is forwarded to the next-hop node.

(b) Nodes emulating OM processing only obtain the next-hop IP address from layer decryption of the layered object. These nodes retain the OM without accessing the payload for a time-span similar to nodes processing the OM, and then the message is forwarded to the next-hop node.

Therefore, external actors observing network communications are not able to identify nodes contributing to the aggregated result; consequently, they cannot associate activities occurring in the monitored environment with messages transiting network nodes.

4.1. Data Filtering and Aggregation

The framework for privacy-aware indoor location makes use of the privacy-preserving communication protocol described in [13] to securely convey to sensor nodes information related to data filtering and aggregation while maintaining the identity of interested nodes hidden from other entities except the message's origin.

The information related to data filtering and aggregation is delivered to sensor nodes in the form of computer code included in the payload of the previously described OM. Sensor nodes processing the OM execute the delivered computer code in a secure execution environment. The execution environment provides restricted access to the underlying sensor node system, allowing the executing computer code to access sensor readings recorded in the last h hours (h a fixed network parameter).

Since the described technique conveys general-purpose computer code to sensor nodes, it is possible to compute virtually any operation on the data of sensor nodes. Therefore, the presented technique can be used to count the number of persons in an environment, identify when and where the social distancing is violated, determine if a room was over-utilized and needs cleaning to prevent the spreading of the virus, etc.

In the following, we show how to verify if the social distancing is violated in a specific area of the monitored environment. The processing of a similar request begins as described in [13]. The sink node receives the request expressing the operation and the target location and starts constructing the OM to answer the request. First, the required operation is converted into a task specified in a general-purpose programming language. The task pseudo-code for addressing the verification of social distancing is shown in Algorithm 1. Then the set of nodes target of the request is selected and the sink node starts constructing OM. Since the communication protocol [13] relies on messages uniform size, the request will be resolved by issuing multiple OM.

An OM including the task given in Algorithm 1 being processed on a node of the smart floor sensor network described in Section 3.2 will perform the following: The data of the sensor network node is first filtered to a narrow time interval ($time_{start}$ and $time_{end}$); the narrower the time interval is, the more accurate the data acquired. All objects detected are filtered from the data by observing the data variance. Then, the data are filtered using the function FILTERSTATIONARY($data$,$time$) to remove all non-stationary activities, and the $time$ argument is used to determine when an activity is considered non-stationary. The observed phenomenon is considered non-stationary when it leaves the sensor in an amount of time lower than $time$ milliseconds. The threshold value must be of $time > (time_{end} - time_{start}) * \frac{1}{2}$; otherwise, repeat event detection may occur. The filtered data are discretized into a value array of underlying sensors, each value describing the number of observed events. The array of values is then stored in w, the data carrying binary string at the position determined by the two symmetric encryption keys and the linear probing technique. Since both symmetric encryption keys are known only to the current node and to the message's origin, other nodes processing the OM cannot identify which node contributed to which value in the data-carrying binary string. The OM is then reassembled and sent to the next-node IP address.

When the OM ends its path at the issuer sink node, the sink node uses the symmetric encryption key obtained from layered object decryption to decipher the OM payload and access the data carrying string. Moreover, the symmetric encryption key acts as the OM identifier. Thus, the sink node can uniquely identify the OM and use information about symmetric encryption keys and the OM path maintained from OM construction to associate the data in the data-carrying string to nodes in the OM path.

Therefore, the sink node gathers the results of all OM issued to resolve a request and uses the collected data to reconstruct the environment representation as shown in Figure 7 to detect where and when the social distancing was violated.

Algorithm 1: Data filtering to supervise social distancing violations

Input: D sensor node data
w binary string
S_1, S_2 symmetric encryption keys

Output: w' modified binary string

Function *Main (args)*:

t_{start},t_{end} ; // Time interval
$t_{contact}$; // Time in milliseconds

// Filter the interval of data between $time_{start}$ and $time_{end}$
D = FilterTimeInterval (D,t_{start},t_{end});
// Exclude objects from the data
D = FilterObjects (D);
// Remove all activities that are not stationary for $t_{contact}$ milliseconds
D = FilterStationary (D,$t_{contact}$);
// Discrertize the data into an value array of underlying sensors, each value describing the number of observed events
$tile_{status}$ = DiscretizeData (D);
// Use encryption keys to find the position in w where insert the data
$$pos = (S_1 + S_2)\% \frac{size(w)}{size(tile_{status})};$$

// Use linear probing to insert data in w
while $w[pos * size(tile_{status})]! = null$ **do**
  $pos+ = 1$;
end
$w[pos * size(tile_{status})] = tile_{status}$;
return w ;
end

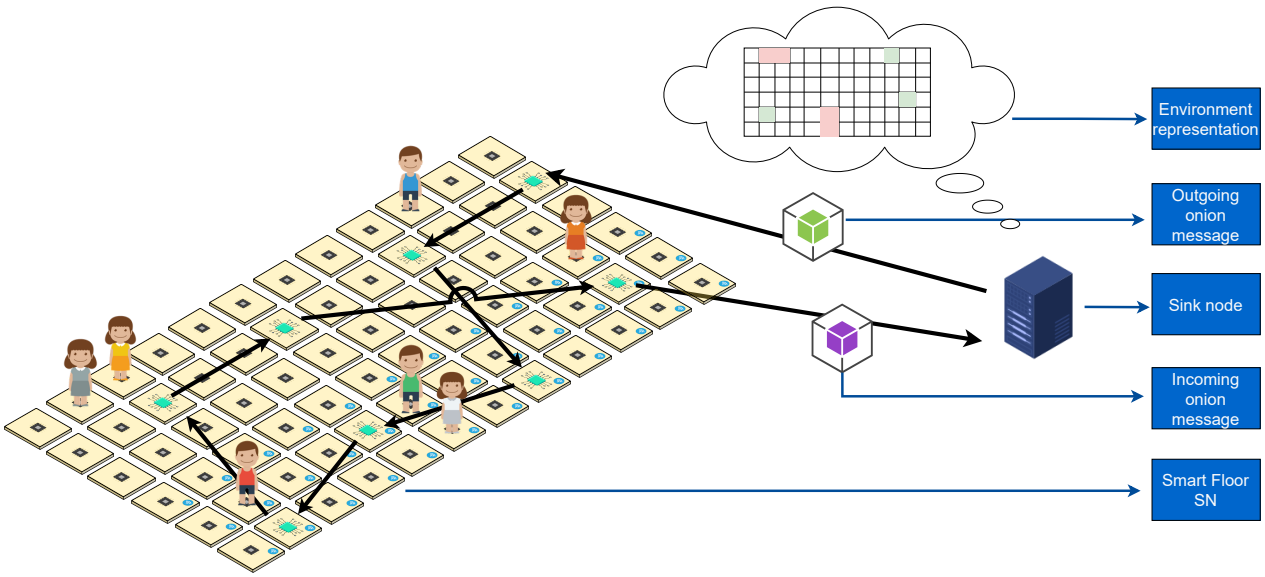

Figure 7. The figure displays the data acquisition layer relying on the privacy-preserving communication protocol in [13]. The environment representation highlights where the social distancing was violated (red-colored squares).

5. Blockchain for Secure Storage and Computation

Blockchain provides a secure, decentralized, transparent and immutable record that has gained a lot of attention. The unique set of properties it provides have directed researchers to seek other uses besides cryptocurrency. The first practical implementation

was Bitcoin, which uses Proof of Work to secure the blockchain coupled with the unspent transaction output (UTXO) transaction model. Even with the limited expressing power of Bitcoin's UTXO model, researchers have demonstrated that an access control can be built [40]. Microsoft implemented a decentralized identity solution running on the Bitcoin network [49], and Factom protocol, which uses the Bitcoin network as a decentralized notary service [50]. Smart contract platforms such as Ethereum use a state-based model in which state transitions are recorded in blocks. This paved the way for the development of smart contracts, Turing complete programs that are recorded on-chain. With smart contracts, more complex applications can be built. Our framework uses the OpenEthereum [51] private network as a smart contract platform that facilitates two main modules, namely Role-based access control (RBAC) and decentralized sink node for the underlying WSNs. In a permissioned setting, Ethereum is configured to run a proof of authority (PoA), in which only a selected group of nodes are configured as validators. In our use case, each building with an indoor location system operates at least one OpenEthereum node. However, preferably, most compute units that serve as sinks should run a light client.

5.1. WSN Sink

In order to perform queries and computation, WSNs are usually deployed with a sink node. Sensors in the network collect information from the environment and ultimately transfer the data to the sink node. In practical implementations, sink nodes usually reside in the cloud and seldom on-site. Whatever the case, sink nodes arguably present a single point of failure (SPOF) of the entire system [52]. Moreover, sink nodes are easier to identify as a target due to their fixed network identity and recognizable traffic patterns. In our solution, we achieve complete decentralization by replacing sink nodes with smart contracts. The sink contract keeps a record of public keys of all nodes in the network. The publicly exposed function *sendQuery()* enables users to initiate a query on a set of tiles and retrieve the result once submitted on-chain. The contract keeps a registry of all the computing nodes, their public keys, and references to which building/area they belong to. A query consists of a set of compute units and a function. To obtain the set of compute units, the sender can call the function *getComputingNodes()*, which checks the senders public key against RBAC and filters the set accordingly. The result is a subset of units, the sender has access to. Upon calling *sendQuery()* the contract creates an onion. The subset of computing nodes should be randomized to avoid using on-chain randomness when creating the onion.

Computing nodes in the WSN run a light client of OpenEthereum and are able to synchronize blocks with reasonable storage and resource requirements. Upon receiving a new block, each node checks the list of added onions to determine the starting node on the route. This is made possible by keeping encryption integrity checks on the first layer. The node whose key passes the integrity check is able to decrypt the first layer and initiate the query. Note that even if the onion is publicly available, no third party can decrypt it or determine the route the query will take. From a network point of view, every query has a sink node, which is pseudo-randomly selected amongst the set of nodes in the underlying WSN, as detailed in Section 4.

The route ends at the starting node, which submits a transaction to the contract storing the result of the computation encrypted with the public key of the original sender. This protects the results on the public ledger so that only the owner of the corresponding private key can view them.

5.2. Role-Based Access Control

RBAC is a smart contract deployed on the blockchain that allows the creation, removal, revocation, and transfer of roles to actors that interact with the sink node contract and underlying WSNs that are queried. Upon adding a new building, the transaction signer is automatically given the role of admin. We divide assets into buildings, areas, and sensors. Initially, each sensor must be registered using the public key and assigned to a area within a building. After the configuration, new roles can be assigned to each of the resources by

protecting their getter methods. This enables administrators to limit access to queries on individual area; i.e., an open space in a public building can be queried by anyone to learn how crowded it is. However, the offices of the public building can only be queried by the manager and occupants. Each of the public functions exposed by the sink contract is first filtered by the RBAC to determine if access is granted.

6. Validation

To validate the proposed privacy-aware framework, we designed an experiment to assess the average response time. We define the response time as the elapsed time between the execution of the sink smart contract and the subsequent transaction storing the result of the data filtering and aggregation. To conduct this investigation, we individually considered the latency introduced by blockchain operations (sink contract execution and subsequent result transaction) and the technique presented in Section 4 for data filtering and aggregation. Specifically, we validated the privacy-aware framework for the wireless configuration of the floor location system. We considered only the wireless configuration since the latency introduced by messages moving in the wireless multi-hop network is inherently higher than in wired settings.

6.1. Data Filtering and Aggregation

To evaluate the data filtering and aggregation duration, we used the simulator PPWSim [53]. PPWSim is based on the NS3 discrete-event simulation environment for Internet systems [54] and is designed to simulate the General Purpose Data and Query Privacy Preserving Protocol described in [13] and estimate network delays. We refer to the network delay as the latency for an OM (onion message) to travel from one node to the node at the next-hop address. To obtain valuable results to validate the proposed framework, we further extended PPWSim to estimate the delay of OM processing.

Experimental Setup

Since the detailed simulation description can be found in [53], in the following, we will outline the simulator parameters selected to obtain network delay results.

The simulator was set up to construct an ad hoc wireless network of 200 nodes. Nodes were deployed according to a grid structure; each node was equidistant from the closest nodes in cardinal directions. The simulated wireless communication conforms to the IEEE 802.11n standard operating at 2.4 GHz at the data rate of 13 Mbps (Modulation Coding Scheme index 1), abd the wireless communication range was set up to allow direct communication only between neighbouring nodes. The maximum transmission unit and maximum segment size were set to the ns-3 default value, 2296 bytes and 536 bytes, respectively.

OMs were transmitted over the TCP protocol, and the routing of packets in the multi-hop network was handled using the Optimized Link State Routing Protocol (OLSR) [55].

As described in [53], the simulator operates by issuing OMs from a node in the center of the network. OMs are issued sequentially; after an OM returns back to the issuer node, the following OM is issued. The central node was set up to issue 30 OM for each value of $n = \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$, the OM path length. OMs are constructed by randomly selecting n nodes to include in the OM path. The OM path is encoded in the layered object. Encryption layers of the layered object are produced using an ECC-based [56] public-key cipher of 256 b key length implemented in the Libsodium library [57]. Each encryption layer includes a shared secret, the next-hop IP address, two 32b symmetric encryption keys, and the inner encryption layer. To replicate the transfer of computer code, OMs are including a payload consisting of padding p = 2.5 k bytes. The OM size at n path length is given in Table 1.

To assess the OM processing delay using PPWSim, we had to first estimate Δ_{om} the maximum execution time of an OM. As described in [13], the Δ_{om} is a fixed network parameter depending on implementation specifics. The Δ_{om} is used to bound the OM

sojourn time on nodes to only a specific amount in order to achieve privacy preservation, as discussed in Section 4. To estimate the Δ_{om} specific to the privacy-aware framework, we measured delays introduced at each step of the OM execution on a node of the floor location system described in Section 3.2. Sixteen FSR sensors characterize each node of the floor location system, and one compute unit, in our implementation the ESP32-DevKitC V4. Table 2 presents the OM execution broken in individual operations, and the delays of operations are reported. We emphasize the fact that in the privacy-aware framework, the nodes of the floor system are executing only operations presented in Table 2. The OM construction involving the computation of many public-key encryption layers is achieved by the smart contract; therefore, this was executed on validator nodes of the blockchain as described in Section 5 and discussed in Section 6.2.

Table 1. Size of the layered object at the selected OM path lengths n. The row total gives the OM total size, including the payload of 2.5 kB.

n	10	20	30	40	50	60	70	80	90	100
Layered object (bytes)	840	1680	2520	3360	4200	5040	5880	6720	7560	8400
Total (bytes)	3340	4180	5020	5860	6700	7540	8380	9220	10,060	10,900

Table 2. OM execution broken in individual operations; the operation execution time was measured on the ESP32-DevKitC V4. Cryptographic operations were carried out using the Libsodium library [57]. The public-key cipher is ECC based using Curve25519 [58] and the symmetric key cipher is ChaCha20.

Operation	ECC Decryption	ECC Decryption	ChaCha20 Encryption	ChaCha20 Decryption	Data Processing
Data	1 B	1 kB	2.5 kB	2.5 kB	15 kB
Execution time	18.4 ms	18.9 ms	1.2 ms	1.1 ms	9.8 ms

Based on the data in Table 2, we estimated that the Δ_{om} appropriate to our system specifics is 35 ms. This value was derived for the OM size at the path length $n = 100$. As reported in Table 2, the ECC decryption is computation intensive only in deriving the shared secret. The ECC decryption of the layered object of 8400 bytes requires 22.2 ms, payload decryption and encryption require 2.3 ms, and payload content execution requires 9.8 ms. Therefore, we obtained a rough estimate of $\Delta_{om} = 35$ ms.

The Δ_{om} estimate was included in the PPWSim following the guidelines defined in [13] specifying that the technique ensures privacy preservation if the OM sojourn time on WSN nodes corresponds to $\Delta_{om} \times r$. r is a randomly chosen float bounded by $1 \leq r \leq 5$.

Therefore, in the extended version of PPWSim, nodes receiving the OM decipher the outer encryption layer of the layered object to reveal the next-hop IP address and the inner encryption layer. The layered object size is uniform by adding the padding of the same number of bytes as the removed layer. The payload is of uniform size, and after the sojourn time of $\Delta_{om} \times r$, the OM is forwarded to the next-hop node. Measurements are taken separately for network delays and OM processing, and the results are presented, respectively, in Figure 8 and Table 3.

Table 3. Delay introduced by OM processing at n nodes. Average and standard deviation are computed for 30 OMs at each value of n.

n	10	20	30	40	50	60	70	80	90	100
mean (seconds)	1.12	2.28	3.23	4.24	5.35	6.38	7.39	8.60	9.55	10.61
std	0.018	0.022	0.046	0.078	0.080	0.092	0.121	0.153	0.097	0.110

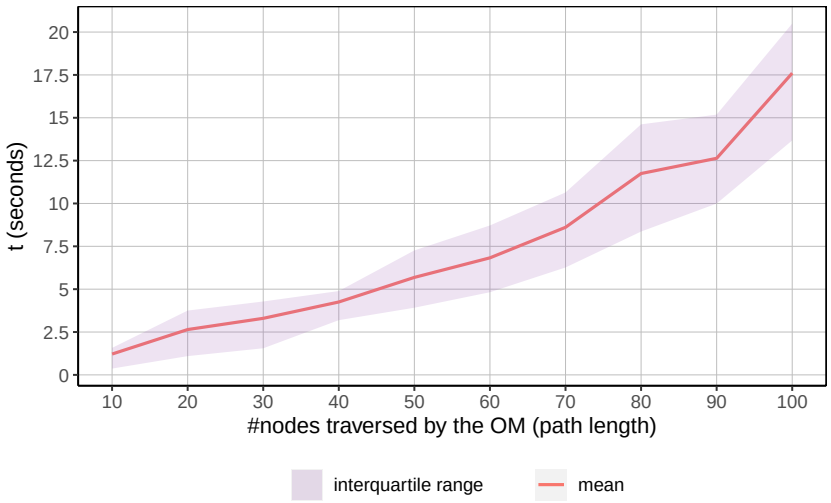

Figure 8. Required time for an OM to travel the selected path length. Measurements do not include the OM processing delay. Statistics are computed for 30 OMs at each OM path length.

6.2. Blockchain

As described in Section 5, the privacy-preserving framework relies on a PoA Ethereum blockchain maintained by a selected group of validator nodes operating in server farm-like settings. Therefore, the smart contracts responsible for RBAC and OM creation are executed on high-performance machines. Several studies [44,59] provide evidence that RBAC could function in similar settings, and reported results show that RBAC operations require low resource consumption. On the other hand, OM creation requires several public-key cryptography operations. We measured the time to create an OM of 100 encryption layers using Curve25519 [58] on a standard laptop (CPU: Intel i5, RAM: 16 GB). The OM construction took 19 ms of CPU time.

However, the time required to execute the mentioned contracts is negligible in the assessment of the framework response time since the blockchain state is propagated only at new block creation. Therefore, the nodes of the floor system running the light client can detect a new OM only after a new block is added to the blockchain. The PoA Ethereum block period is usually in the range from 2 to 15 s [60].

6.3. Discussion

We provided the validation of the WSN framework for privacy-aware indoor location by assessing its response time. The reported results show that applying the PoA Ethereum on the floor system does not introduce significant latency in response times. Nonetheless, it binds the detection of new OMs and the result transaction to a discrete basis imposed by the block period.

Moreover, the results show the applicability of the General Purpose Data and Query Privacy Preserving Protocol [13] to the indoor location floor system. The data in Figure 8 and Table 3 show that in the extreme scenario of OM path length $n = 100$, the OM Round-Trip-Time is generally less than 30 s. However, in practical implementations, the system will rather rely on multiple smaller OMs executed in parallel than one large OM. Therefore, the framework response time is reduced to approximately 10 s + two block periods if parallel OM execution is applied at $n = 50$.

7. Conclusions and Future Work

In this paper, we present a system for privacy-preserving, non-intrusive, and secure indoor location monitoring. We specifically design the system to not allow identification through data filtering. We present an innovative way of passively approximating location by measuring the force applied to the floor. We are able to distinguish objects from persons by observing the activity at the local level. The sensitized floor forms a WSN that is secure

from both external and internal adversaries. By designing a unique onion routing-based protocol, we were able to conceal the network identity of nodes in the WSN. Moreover, our onion-based approach allows a general-purpose computing model for distributed algorithms, and to the best of our knowledge, no comparable solution exists. To address the issue of SPOF on sink nodes, we used blockchain-based smart contracts that replace the onion creation and storage of query results. The blockchain operates in permissioned mode in which sink nodes are registered, and their public keys stored on the blockchain. We also show how using a blockchain-based RBAC is possible to further protect the query and data access. We validate our solution on our use case of tracking violations of indoor physical distancing restrictions to avoid the spread of COVID-19.

The presented solution aims at an implementation of a self-managing system to control the compliance to a set of predefined rules, such as the COVID-19 pandemic rules issued by local governments. The set of rules can be arbitrarily defined and modified without requiring updates of sensor nodes.

A typical use-case for the presented system would be the installation in a nursing home. The occupants are automatically pseudo-identified by the system in bedrooms and later tracked along the corridors of the building, ensuring an overview of the number of occupants in specific areas, triggering temporary blocks and sanitizing actions.

The system cannot be used as a critical contact signalling system (such as the exposure notification system by Google and Apple) as it is lacking a backward loop that would enable the information about an infection or critical contact to be attributed to a specific person.

Future work should explore more sophisticated algorithms for the detection and tracking of users. Data should be analyzed to advance our understanding of behavior in an effort to improve future building designs. Implementations that aim to identify occupants (i.e., elderly homes) should explore a key management scheme and extend the smart contracts to include the ability for users to grant the system permissions to use their data.

Author Contributions: Conceptualization, A.T.; methodology: A.T., N.H. and J.V.; software, A.T. and N.H.; validation, A.T., J.V. and N.H.; formal analysis, A.T., J.V. and N.H.; investigation, A.T., J.V. and N.H.; funding acquisition and resources, J.V.; data curation, N.H.; writing—original draft preparation, A.T., N.H.; writing—review and editing, J.V.; visualization, A.T. and N.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by H2020 grant number 739574 and 857188 by the Slovenian Research Agency (ARRS) grant number J2-2504.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: The authors gratefully acknowledge the European Commission for funding the InnoRenew CoE project (H2020 Grant Agreement #739574) as well as the Slovenian Research Agency (ARRS) for supporting project number J2-2504.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Riffe, T.; Acosta, E. Data Resource Profile: COVerAGE-DB: A global demographic database of COVID-19 cases and deaths. *Int. J. Epidemiol.* **2021**, *50*, 390–390f. [CrossRef]
2. Bale, R.; Li, C.G.; Yamakawa, M.; Iida, A.; Kurose, R.; Tsubokura, M. Simulation of droplet dispersion in COVID-19 type pandemics on Fugaku. In Proceedings of the Platform for Advanced Scientific Computing Conference, Geneva, Switzerland, 5–9 July 2021; pp. 1–11.
3. Ooi, C.C.; Suwardi, A.; Ouyang, Z.L.; Xu, G.; Tan, C.K.I.; Daniel, D.; Li, H.; Ge, Z.; Leong, F.Y.; Marimuthu, K.; et al. Risk assessment of airborne COVID-19 exposure in social settings. *Phys. Fluids* **2021**, *33*, 087118. [CrossRef]
4. Sun, C.; Zhai, Z. The efficacy of social distance and ventilation effectiveness in preventing COVID-19 transmission. *Sustain. Cities Soc.* **2020**, *62*, 102390. [CrossRef] [PubMed]

5. Singh, R.P.; Javaid, M.; Haleem, A.; Suman, R. Internet of things (IoT) applications to fight against COVID-19 pandemic. *Diabetes Metab. Syndr. Clin. Res. Rev.* **2020**, *14*, 521–524. [CrossRef] [PubMed]
6. Singh, P.K.; Nandi, S.; Ghafoor, K.Z.; Ghosh, U.; Rawat, D.B. Preventing covid-19 spread using information and communication technology. *IEEE Consum. Electron. Mag.* **2020**, *10*, 18–27. [CrossRef]
7. Kumar, K.; Kumar, N.; Shah, R. Role of IoT to avoid spreading of COVID-19. *Int. J. Intell. Netw.* **2020**, *1*, 32–35. [CrossRef]
8. Dong, Y.; Yao, Y.D. IoT platform for COVID-19 prevention and control: A survey. *IEEE Access* **2021**, *9*, 49929–49941. [CrossRef]
9. Garg, L.; Chukwu, E.; Nasser, N.; Chakraborty, C.; Garg, G. Anonymity preserving IoT-based COVID-19 and other infectious disease contact tracing model. *IEEE Access* **2020**, *8*, 159402–159414. [CrossRef]
10. Chan, H.; Perrig, A. Security and privacy in sensor networks. *Computer* **2003**, *36*, 103–105. [CrossRef]
11. Gao, Y.; Ao, H.; Feng, Z.; Zhou, W.; Hu, S.; Tang, W. Mobile network security and privacy in WSN. *Procedia Comput. Sci.* **2018**, *129*, 324–330. [CrossRef]
12. Shi, Q.; Zhang, Z.; He, T.; Sun, Z.; Wang, B.; Feng, Y.; Shan, X.; Salam, B.; Lee, C. Deep learning enabled smart mats as a scalable floor monitoring system. *Nat. Commun.* **2020**, *11*, 4609. [CrossRef] [PubMed]
13. Hrovatin, N.; Tošić, A.; Mrissa, M.; Vičič, J. A General Purpose Data and Query Privacy Preserving Protocol for Wireless Sensor Networks. *arXiv* **2021**, arXiv:2111.14994.
14. Goldschlag, D.M.; Reed, M.G.; Syverson, P.F. Hiding routing information. In *International Workshop on Information Hiding*; Springer: Berlin/Heidelberg, Germany, 1996; pp. 137–150.
15. Thulasiraman, P.; Haakensen, T.; Callanan, A. Countering passive cyber attacks against sink nodes in tactical sensor networks using reactive route obfuscation. *J. Netw. Comput. Appl.* **2019**, *132*, 10–21. [CrossRef]
16. Brena, R.F.; García-Vázquez, J.P.; Galván-Tejada, C.E.; Muñoz-Rodriguez, D.; Vargas-Rosales, C.; Fangmeyer, J. Evolution of indoor positioning technologies: A survey. *J. Sens.* **2017**, *2017*, 2630413. [CrossRef]
17. Farahsari, P.S.; Farahzadi, A.; Rezazadeh, J.; Bagheri, A. A Survey on Indoor Positioning Systems for IoT-based Applications. *IEEE Internet Things J.* **2022**, *early access*. [CrossRef]
18. Want, R.; Hopper, A.; Falcao, V.; Gibbons, J. The active badge location system. *ACM Trans. Inf. Syst. (TOIS)* **1992**, *10*, 91–102. [CrossRef]
19. Forestiero, A. Metaheuristic algorithm for anomaly detection in Internet of Things leveraging on a neural-driven multiagent system. *Knowl.-Based Syst.* **2021**, *228*, 107241. [CrossRef]
20. Forestiero, A. Self-organizing anomaly detection in data streams. *Inf. Sci.* **2016**, *373*, 321–336. [CrossRef]
21. Orr, R.J.; Abowd, G.D. The smart floor: A mechanism for natural user identification and tracking. In Proceedings of the CHI'00 Extended Abstracts on Human Factors in Computing Systems, The Hague, The Netherlands, 1–6 April 2000; pp. 275–276.
22. He, C.; Zhu, W.; Chen, B.; Xu, L.; Jiang, T.; Han, C.B.; Gu, G.Q.; Li, D.; Wang, Z.L. Smart floor with integrated triboelectric nanogenerator as energy harvester and motion sensor. *ACS Appl. Mater. Interfaces* **2017**, *9*, 26126–26133. [CrossRef]
23. Kaddoura, Y.; King, J.; Helal, A. Cost-precision tradeoffs in unencumbered floor-based indoor location tracking. In Proceedings of the Third International Conference On Smart Homes and Health Telematic (ICOST), Sherbrooke, QC, Canada, 4–6 July 2005.
24. Shen, Y.L.; Shin, C.S. Distributed sensing floor for an intelligent environment. *IEEE Sens. J.* **2009**, *9*, 1673–1678. [CrossRef]
25. Yang, X.; Gao, L.; Zheng, J.; Wei, W. Location privacy preservation mechanism for location-based service with incomplete location data. *IEEE Access* **2020**, *8*, 95843–95854. [CrossRef]
26. Shen, H.; Zhou, J.; Cao, Z.; Dong, X.; Choo, K.K.R. Blockchain-based lightweight certificate authority for efficient privacy-preserving location-based service in vehicular social networks. *IEEE Internet Things J.* **2020**, *7*, 6610–6622. [CrossRef]
27. Selamneni, V.; Dave, A.; Mondal, S.; Mihailovic, P.; Sahatiya, P. Large Area Pressure Sensor for Smart Floor Sensor Applications—An Occupancy Limiting Technology to Combat Social Distancing. *IEEE Consum. Electron. Mag.* **2021**, *10*, 98–103. [CrossRef]
28. Tošić, A.; Hrovatin, N.; Vičič, J. Data about fall events and ordinary daily activities from a sensorized smart floor. *Data Brief* **2021**, *37*, 107253. [CrossRef] [PubMed]
29. Gu, T.; Fang, Z.; Abhishek, A.; Mohapatra, P. IoTSpy: Uncovering Human Privacy Leakage in IoT Networks via Mining Wireless Context. In Proceedings of the 2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications, London, UK, 31 August–3 September 2020; pp. 1–7.
30. Zhang, F.; He, W.; Liu, X. Defending against traffic analysis in wireless networks through traffic reshaping. In Proceedings of the 2011 31st International Conference on Distributed Computing Systems, Minneapolis, MN, USA, 20–24 June 2011; pp. 593–602.
31. Saltaformaggio, B.; Choi, H.; Johnson, K.; Kwon, Y.; Zhang, Q.; Zhang, X.; Xu, D.; Qian, J. Eavesdropping on fine-grained user activities within smartphone apps over encrypted network traffic. In Proceedings of the 10th {USENIX} Workshop on Offensive Technologies ({WOOT} 16), Austin, TX, USA, 8–9 August 2016.
32. Middya, R.; Chakravarty, N.; Naskar, M.K. Compressive sensing in wireless sensor networks—A survey. *IETE Tech. Rev.* **2017**, *34*, 642–654. [CrossRef]
33. Zheng, H.; Yang, F.; Tian, X.; Gan, X.; Wang, X.; Xiao, S. Data gathering with compressive sensing in wireless sensor networks: A random walk based approach. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *26*, 35–44. [CrossRef]
34. Hu, P.; Xing, K.; Cheng, X.; Wei, H.; Zhu, H. Information leaks out: Attacks and countermeasures on compressive data gathering in wireless sensor networks. In Proceedings of the IEEE INFOCOM 2014-IEEE Conference on Computer Communications, Toronto, ON, Canada, 27 April–2 May 2014; pp. 1258–1266.

35. Xie, K.; Ning, X.; Wang, X.; He, S.; Ning, Z.; Liu, X.; Wen, J.; Qin, Z. An efficient privacy-preserving compressive data gathering scheme in WSNs. *Inf. Sci.* **2017**, *390*, 82–94. [CrossRef]
36. Paillier, P. Public-key cryptosystems based on composite degree residuosity classes. In *International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 223–238.
37. Xu, J.; Yang, G.; Chen, Z.; Wang, Q. A survey on the privacy-preserving data aggregation in wireless sensor networks. *China Commun.* **2015**, *12*, 162–180. [CrossRef]
38. Bista, R.; Chang, J.W. Privacy-preserving data aggregation protocols for wireless sensor networks: A survey. *Sensors* **2010**, *10*, 4577–4601. [CrossRef]
39. Sandhu, R.S. Role-based access control. In *Advances in Computers*; Elsevier: Amsterdam, The Netherlands, 1998; Volume 46, pp. 237–286.
40. Di Francesco Maesa, D.; Mori, P.; Ricci, L. Blockchain Based Access Control. In *IFIP International Conference on Distributed Applications and Interoperable Systems*; Chen, L.Y., Reiser, H.P., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 206–220.
41. Hu, V.C.; Kuhn, D.R.; Ferraiolo, D.F.; Voas, J. Attribute-based access control. *Computer* **2015**, *48*, 85–88. [CrossRef]
42. Sandhu, R.S.; Samarati, P. Access control: Principle and practice. *IEEE Commun. Mag.* **1994**, *32*, 40–48. [CrossRef]
43. Achour, I.; Ayed, S.; Idoudi, H. On the Implementation of Access Control in Ethereum Blockchain. In Proceedings of the 2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Virtual, 29–30 September 2021; pp. 483–487.
44. Cruz, J.P.; Kaji, Y.; Yanai, N. RBAC-SC: Role-based access control using smart contract. *IEEE Access* **2018**, *6*, 12240–12251. [CrossRef]
45. Wadaa, A.; Olariu, S.; Wilson, L.; Eltoweissy, M.; Jones, K. On providing anonymity in wireless sensor networks. In Proceedings of the Tenth International Conference on Parallel and Distributed Systems, Istanbul, Turkey, 15–20 July 2004; pp. 411–418.
46. Syverson, P.F.; Goldschlag, D.M.; Reed, M.G. Anonymous connections and onion routing. In Proceedings of the 1997 IEEE Symposium on Security and Privacy (Cat. No. 97CB36097), Oakland, CA, USA, 4–7 May 1997; pp. 44–54.
47. Rivest, R.L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]
48. Sunshine, C.A. Source routing in computer networks. *ACM SIGCOMM Comput. Commun. Rev.* **1977**, *7*, 29–33. [CrossRef]
49. Microsoft. Decentralized Identity. 2018. Available online: https://query.prod.cms.rt.microsoft.com/cms/api/am/binary/RE2DjfY (accessed on 19 December 2021).
50. Snow, P.; Deery, B.; Kirby, P.; Johnston, D. Factom Ledger by Consensus. 2015. Available online: https://cryptochainuni.com/wp-content/uploads/Factom-Ledger-by-Consensus.pdf (accessed on 19 December 2021).
51. Buterin, V. Ethereum white paper. *GitHub Repos.* **2013**, *1*, 22–23.
52. Kohno, E.; Ohta, T.; Kakuda, Y. Secure decentralized data transfer against node capture attacks for wireless sensor networks. In Proceedings of the 2009 International Symposium on Autonomous Decentralized Systems, Athens, Greece, 23–25 March 2009; pp. 1–6.
53. Hrovatin, N.; Tošić, A.; Vičič, J. Ppwsim: Privacy Preserving Wireless Sensor Network Simulator. *SSRN* **2021**. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3978796 (accessed on 19 December 2021).
54. Henderson, T.R.; Lacage, M.; Riley, G.F.; Dowell, C.; Kopena, J. Network simulations with the ns-3 simulator. *SIGCOMM Demonstr.* **2008**, *14*, 527.
55. Clausen, T.; Jacquet, P.; Adjih, C.; Laouiti, A.; Minet, P.; Muhlethaler, P.; Qayyum, A.; Viennot, L. *Optimized Link State Routing Protocol (OLSR)*. RFC; INRIA. 2003. Available online: https://hal.inria.fr/inria-00471712/ (accessed on 19 December 2021).
56. Miller, V.S. Use of elliptic curves in cryptography. In *Conference on the Theory and Application of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 1985; pp. 417–426.
57. Libsodium. The Sodium Crypto Library. Available online: https://libsodium.gitbook.io/doc/ (accessed on 28 May 2021).
58. Bernstein, D.J. Curve25519: New Diffie-Hellman speed records. In *International Workshop on Public Key Cryptography*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 207–228.
59. Rahman, M.U.; Baiardi, F.; Guidi, B.; Ricci, L. Protecting personal data using smart contracts. In *International Conference on Internet and Distributed Computing Systems*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 21–32.
60. Schäffer, M.; Angelo, M.D.; Salzer, G. Performance and scalability of private Ethereum blockchains. In *International Conference on Business Process Management*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 103–118.