BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 05/10 pp443–464
Volume 6, Number 4, December 2023
DOI: 10.26599/BDMA.2023.9020012

Towards Privacy-Aware and Trustworthy Data Sharing Using Blockchain for Edge Intelligence

Youyang Qu, Lichuan Ma*, Wenjie Ye, Xuemeng Zhai, Shui Yu, Yunfeng Li*, and David Smith

Abstract: The popularization of intelligent healthcare devices and big data analytics significantly boosts the development of Smart Healthcare Networks (SHNs). To enhance the precision of diagnosis, different participants in SHNs share health data that contain sensitive information. Therefore, the data exchange process raises privacy concerns, especially when the integration of health data from multiple sources (linkage attack) results in further leakage. Linkage attack is a type of dominant attack in the privacy domain, which can leverage various data sources for private data mining. Furthermore, adversaries launch poisoning attacks to falsify the health data, which leads to misdiagnosing or even physical damage. To protect private health data, we propose a personalized differential privacy model based on the trust levels among users. The trust is evaluated by a defined community density, while the corresponding privacy protection level is mapped to controllable randomized noise constrained by differential privacy. To avoid linkage attacks in personalized differential privacy, we design a noise correlation decoupling mechanism using a Markov stochastic process. In addition, we build the community model on a blockchain, which can mitigate the risk of poisoning attacks during differentially private data transmission over SHNs. Extensive experiments and analysis on real-world datasets have testified the proposed model, and achieved better performance compared with existing research from perspectives of privacy protection and effectiveness.

Key words: edge intelligence; blockchain; personalized privacy preservation; differential privacy; Smart Healthcare Networks (SHNs)

1 Introduction

With recent advances like machine learning and intelligent edge devices, the wide proliferation of smart healthcare systems has been enabled. Consequently, a wide range of applications has emerged and serviced our daily life. Among all of them, Smart Health Networks (SHNs) is one of the most widespread services that has been adaopted in real-world scenarios[1]. Healthcare has been a long-lasting concern of the society, and the development of advanced technologies takes it to a new stage. In this case, people rely more and more on smart

• Youyang Qu and David Smith are with Data61, Commonwealth Scientific and Industrial Research Organization (CSIRO), Sydney 2015, Australia. E-mail: {youyang.qu, david.smith}@data61.csiro.au.
• Lichuan Ma is with School of Cyber Engineering, Xidian University, Xi’an 710126, China. E-mail: lcma@xidian.edu.cn.
• Wenjie Ye is with the College of Engineering and Science, Victoria University, Melbourne 3000, Australia. E-mail: wenjie.ye@vu.edu.au.
• Xuemeng Zhai is with School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China. E-mail: zxm@uestc.edu.cn.
• Shui Yu is with School of Computer Science, University of Technology Sydney, Sydney 2007, Australia. E-mail: shui.yu@uts.edu.au.
• Yunfeng Li is with CNPIEC KEXIN LTD., Beijing 100020, China. E-mail: liyunfeng@cnpiec.com.cn.
∗ To whom correspondence should be addressed.

Manuscript received: 2022-12-06; revised: 2023-05-30; accepted: 2023-06-04

healthcare services to enhance their living quality. To make this happen, doctors or patients are willing to establish communities in SHNs with regards to a certain decease, for example, Doximity and Curofy[2,3]. It is worth mentioning that SHN users are likely to form the almost same communities in various smart healthcare networks[4,5]. On one hand, more services may be delieved. On the other hand, multiple data resources of uses are natually disclosed.

Intuitively, the key target of SHNs is to share useful information among community uses. The shared health data usually contain texts, medias, as well as spatial and temporal data[6,7]. The combination of these data can be used to re-identify a specific person and leads to further privacy disclosure. Thus, great risks are raised when sensitive health data of individuals are published without proper pre-processing. This can be even worse when different parties can access the data without proper access control[8,9].

It has been agreed on that sensitive information, especially sensitive health information raises the financial interest of diverse adversaries or attackers. New attacks are reported every several months, or even weeks. Adversaries are usually patient and smart enough to collect individual's data from various data sources, which can be used for re-identification[10]. This also causes a worse situation that linkage attacks targeting on further private data are possible[11,12]. Therefore, it is necessary to establish effective privacy protection mechanisms for SHNs.

To preserve privacy, three classic methods have been well studied, which are cryptography, anonymization and clustering, and differential privacy. Cryptography preserves privacy during the data packet transmission process but can hardly preserve privacy against data recipients[13]. Anonymization and clustering have been developing for several decades. Several benchmark methods include K-anonymity[14], L-diversity[15], and T-closeness[16]. Existing clustering methods consider record number, record type, record distribution, or a combination of them. However, they are not suitable for streaming data sharing. Differential privacy[17,18] is a powerful privacy protection tool constrained by mathematical theories. But for classic differential privacy and its variants, the privacy protection level is usually constant.

There are some pioneering works in personalized privacy protection. For instance, using virtual online distance as the penalization index is a representative work[19]. However, virtual online distance has some issues during deployment. First, the distance is not easy to define. Second, friends in the network may have the exactly same distance as attackers in some cases. Besides, some blockchain-based solutions are devised to potentially add extra protection for privacy. For example, Wang et al.[20] developed a blockchain-powered healthcare system. Blockchain-based solutions can ensure authentication and integrity, but the public accessibility of health data puts privacy at great risk.

Motivated by this, we develop a personalized differential privacy protection model, which can derive an optimized trade-off between privacy protection and health data utility. Personalization is achieved by a trust level measured by community density. By defining community density, it can be used as the measurement of the intimacy of a group of people. To avoid utility loss, we devise a novel community partition method based on the work done by Ahn et at.[21] Besides, we use a semi-sigmoid function as a mapping function, which maps the community-enabled trust to a protection level. Then, a Markov stochastic process is built to uncouple the randomized noise relationship, mitigating the linkage attacks. Moreover, we design a tailor-made blockchain structure to accommodate the community-based personalized privacy protection model while avoiding any data falsification attacks and ensuring data integrity during transmission over SHNs.

The main contributions of this work are summarized as follows.

- **Personalized and trustworthy privacy Protection.** We define community density to measure the trust within communities. Then, the trust is mapped to a privacy protection level constrained by differential privacy. In this way, we develop a novel personalized and trustworthy privacy protection model.
- **Data falsification proof.** We devise a tailor-made blockchain structure that can support the personalized and trustworthy privacy protection model. The differentially private health data are guaranteed to be authenticated provided by the features of this blockchain structure.
- **Attack-proofing and optimization trade-off.** We properly decouple the data correlation with Markov stochastic process. Therefore, the linkage attack can be eliminated. Furthermore, the proposed model achieves an optimized trade-off between personalized privacy protection and improved data utility.
- **Better performance.** Extensive results obtained

from experiments show that the proposed system can achieve a good balance between personalized privacy protection and health data utility. Besides, the system can defeat leading attacks, like linkage attacks and poisoning attacks.

2 Related Work

Research related to privacy preservation in SHNs has gained significant attention due to the sensitive and personal nature of healthcare data. Privacy preservation techniques aim to protect individuals' privacy while enabling the sharing and analysis of healthcare data. These approaches offer several advantages, but also face certain shortcomings.

One major advantage of privacy preservation techniques in SHNs is the protection of individuals' sensitive medical information. By applying privacy-preserving mechanisms, such as data anonymization, encryption, or differential privacy, personally identifiable information can be safeguarded. This ensures that unauthorized entities cannot directly link healthcare data to specific individuals, reducing the risk of privacy breaches.

Another advantage is the potential to enable secure data sharing and collaboration among healthcare stakeholders. Privacy-preserving techniques allow healthcare providers, researchers, and organizations to share data while maintaining privacy. This promotes collaborative efforts in research, clinical decision-making, and public health without compromising sensitive information.

Furthermore, privacy preservation techniques contribute to building trust and compliance with privacy regulations. Patients and individuals are more likely to participate in data sharing initiatives if they have confidence in the privacy protections implemented. Compliance with regulations, such as the Health Insurance Portability and Accountability Act (HIPAA) in the United States or the General Data Protection Regulation (GDPR) in the European Union, is crucial in maintaining ethical and legal standards.

However, privacy preservation in SHNs also has some shortcomings. One significant challenge is balancing privacy protection with data utility. Applying rigorous privacy measures may introduce noise or limitations to the data, which can potentially impact its usefulness for analysis and decision-making. Striking the right balance between privacy and utility remains an ongoing research challenge.

Another issue is the potential for re-identification attacks. Despite privacy-preserving techniques, there is still a risk of re-identification when data is combined or linked with other external information sources. Techniques and safeguards need to be continuously developed and updated to address this vulnerability.

Additionally, the complexity and diversity of healthcare data pose challenges for privacy preservation. Healthcare data encompasses a wide range of data types, including structured medical records, genomics data, and wearable sensor data. Developing privacy mechanisms that are effective across diverse data types and modalities requires ongoing research and adaptation.

Overall, research in privacy preservation for SHNs is crucial for protecting sensitive healthcare information, facilitating secure data sharing, and ensuring compliance with privacy regulations. While these techniques offer notable advantages, addressing the challenges of maintaining data utility, preventing re-identification attacks, and accommodating diverse healthcare data types will be vital for advancing privacy preservation in SHNs.

Four successful branches of privacy protection solutions include clustering, cryptography, game theory, and differential privacy[22]. The clustering-based solutions focus on how to group the datasets considering the consistency of magnitude, diversity, and distribution between each cluster and the whole datasets[14–16]. However, such a clustering approach fails to function well for large-scale and heterogeneous datasets. Cryptography-based solutions protect privacy in a point-to-point manner. However, an unknown adversary always has the potential to decrypt the data while the computational complexity is quite high[13]. Game theory is another potential optimization approach, so as to balance privacy with data utility. However, such an approach also suffers from the modelling accuracy of actions and payoffs in different contexts. It is worth noting here that the increasing complexity of the number of participants makes it hard to generalize[23]. For classic differential privacy and its variants, there are mathematical constraints to explain the privacy protection performance, but the privacy protection level is usually constant[17, 18, 24, 25].

In the privacy protection domain, linkage attacks have always been big problems, especially when we consider multiple SHNs. A lot of features have been used for accurate re-identification in this scenario, such as content matching, profile matching, unique structure matching, etc.[26,27]

By using binary classifiers, Perito et al.[28] compared the similarity of pseudo identities and achieved re-identification of users. While in Ref. [29], Zafarani and Liu defined behaviors based on pseudo identities to establish a user mapping model. With media data, Xu et al.[30] devised a high-efficiency identification model as well as a countermeasure. In addition, both static location data and trajectory data are utilized to re-identify a specific individual in Refs. [31, 32], respectively. Based on existing research, Li et al.[33] surveyed the issues and solutions of de-anonymization and aggregation of heterogeneous social networks.

To preserve the data privacy of SHNs, existing research has made great efforts, especially from the aspect of health data. Yang et al.[34] leveraged access control to preserve the privacy of health data sharing over the Internet of Things (IoTs). Another similar research was performed by Zhang et al.[35], who further introduced attribute-based access control to preserve health data privacy while considering the efficiency. In the smart wearable healthcare devices case, Liu et al.[36] designed a cooperative privacy-preserving model. The above three are cryptography methods. Despite their effectiveness, the low efficiency and granularity of privacy protection prevent their further application in SHNs.

In SHNs, an emerging technology, blockchain, is also considered by researchers[37]. To preserve big health data privacy, Xu et al.[38] designed a blockchain-based privacy protection model. Decentralized privacy protection of SHNs data is considered by Dwivedi et al.[39] using a tailor-made healthcare blockchain. Besides, Peterson et al.[40] also proposed blockchain-based private SHN data-sharing schemes. However, for most blockchain-based systems, the feature of publicly accessible private data is still a big issue and not well-addressed[41].

In healthcare application scenarios, differential privacy has been widely deployed. Nevertheless, the constant privacy protection level limits its further development. Therefore, a more flexible protection mechanism, like personalized differential privacy, is necessary. But personalized privacy suffers from security and privacy vulnerabilities as well. This is even worse in the smart healthcare context since the data are highly confidential and related to physical security. Based on our literature review, this part has not yet been well-discussed. The idea we propose in this paper is a preliminary exploration in this field that integrates personalized differential privacy protection, trust, blockchain, etc., to provide several advantageous features.

3 Personalized and Trustworthy Privacy Protection in SHNs

In this section, we present the devised personalized and trustworthy privacy-preserving model for data sharing in SHNs. We first describe our community detection algorithm and discuss the structure-based parameters. In this paper, we use a modified link community algorithm to achieve personalized privacy protection. The major modification is that we require users to belong to a single community at the final stage. The existing link community algorithm allows the user to be a part of several communities, which is not feasible in this scenario. That is because a user may access a piece of data but under different protection levels if he/she is part of several communities. As will be analyzed further, linkage attacks can be launched by a single user without collusion, which often provides incentives to malicious users. To overcome this problem, each user is only allocated to one single community with the highest trust level (highest community density and lowest protection level).

Then, to map the trust level to the privacy protection level, we introduce a semi-sigmoid function. The function value barely increases in the low range and high range but increases almost linearly in the middle range. After that, personalized and trustworthy privacy protection based on trust level is discussed and explained in detail.

3.1 Modeling of SHN's graph structure

In recent years, SHNs have been evolving at a faster pace. They are of several different forms. For example, some of them are tailor-made SHNs for doctors or patients only to share the disease information like diabetes or cancers, such as Doximity and Curofy[2]. Besides, some other social networks provide smart healthcare services, which is a sub-network functioning within the whole social network, such as Facebook. No matter the form of the SHNs, they are essentially networks with nodes as the users and edges denoting users' relationships. Therefore, it is reasonable to use the undirected graph of graph theory to model an SHN.

In this context, we use a single SHN as an example. Multiple SHN privacy issues is a simple extension of the single SHN. As mentioned above, the undirected graph is used to model the SHN, which is the foundation of the whole system. A single SHN is denoted by G, where $G = \{v_i, e_i, c_i \mid v \in V, e \in E, c \in C\}$. In this graph, $v \in V$ is the node (user in SHN), $e \in E$ is the edge (relationship between two users of SHN), and $c \in C$ is the community (formed by multiple users in SHN).

If there exists one edge (e.g., $e_{i,j}$) between two nodes (u_i, u_j), then there is relationship between these two nodes. In this context, different from traditional methods, the community is defined over a set of edges instead of nodes, which is $C = \{e_{i,j} | e \in E\}$.

In order to better clarify, we assume the modeled G to be an undirected graph. The proposed model functions well even if this assumption is removed. In addition, we assume there is no trusted central authority. The decentralized blockchain system processes the data with ϵ-differential privacy and delivers data with privacy-preserving communication. The total privacy budget of an SHN is set to be B. To allocate the budget to each user, it depends on the personzalized sensitivity value modeled in the following sections.

Privacy losses accumulate with an increasing number of data sharing. When two answers are responded to an individual, the total privacy loss increases while the privacy protection weakens. To guarantee significant privacy protection, the data curator should set a maximum privacy loss, in particular, ϵ, which is also known as the privacy budget. For instance, a privacy "cost" incurs when data are shared under a defined privacy protection level. The continuous data-sharing process results in accumulating privacy loss. It can be told that ϵ is the power of the natural logarithm. Thus, the protection level is promoted with the decrease of ϵ, data utility decreases in the meantime.

3.2 Community structure detection

To personalize the privacy protection level, we use community density to evaluate the trust within the community. The community density method is more suitable in this scenario compared with the traditional method like virtual online distance. The virtual distance only considers the relationship between two users, and may not be practical when attackers are within a relatively short distance. However, for communities, we evaluate the interaction among all members, which makes the trust more reliable.

The trust among users is evaluated by community density. If the users have dense interaction with each other, the trust in the community is high and thereby privacy protection will be released correspondingly. This is because users may share more information with people they trust, even in online scenarios like SHNs. Based on the density value, we map it to customizable privacy protection levels. To calculate community density, we first partition the whole graph into communities using Algorithm 1.

Algorithm 1 presents an algorithm for detecting communities in a smart healthcare network. Initially, each edge in the network is treated as an individual community. By examining the connectivity patterns, the algorithm calculates the similarity between pairs of edges, considering the number of neighbors of the connected nodes. These similarities are then sorted in descending order. Starting from the pair with the highest similarity, then, it progressively merges the corresponding communities, representing the merging process using a tree structure. The iteration continues until the partitioning density, which measures the density of connections within communities, reaches or exceeds a predefined threshold, then it transforms the edge pair-based tree graph into a node-based graph, where each node represents a community. Finally, it identifies the communities and any overlapped nodes within the transformed graph, which are then outputted as the detected communities in the smart healthcare network.

Algorithm 1 Edge pairs based community detection

```
Input: Smart healthcare network graph represented by $G$
Output: Set of detected communities $C = \{C_1, C_2, \ldots, C_c\}$
Initialize each edge $e_i \in E$ as a community;
Initialize amount of neighbours to each end node $V_+(i)$;
Calculate edge pair similarity $S(e_{ij}, e_{ik})$ with $V_+(j)$ and $V_+(k)$ ;
Sort all similarities $S(e_{ij}, e_{ik})$ in descending order;
Merge the communities based on the ordered edge pairs;
Represent the merging process with a tree structure;
Define a threshold for partitioning density $P_t$;
while $P < P_t$ do
    Calculate edge pairs number $M$ and communities number $|C|$;
    Calculate $m_c$ and $n_c$ as edge pairs number and nodes number in a community $C_c$, respectively;
    Derive partitioning density of $C_c$ with $m_c$ and $n_c$;
    Update partitioning density of the whole smart healthcare network as $P = \frac{1}{M} \sum_c m_c P_c$;
end while
Transform edge pair based tree graph into node-based graph;
Identify communities $C = \{C_1, C_2, \ldots, C_c\}$ and overlapped nodes $u_i \in U$;
Output the communities $C = \{C_1, C_2, \ldots, C_c\}$.
```

Algorithm 1 begins by initializing each edge in the smart healthcare network graph G as a separate community. This initialization step has a time complexity of $O(|E|)$, where $|E|$ represents the number of edges in the graph. Similarly, the initialization of the number of neighbors to each end node also takes $O(|E|)$ time as it requires iterating over all the edges in the network.

The next steps involve calculating the similarity between pairs of edges and sorting them in descending order based on these similarities. The time complexity of calculating the edge pair similarity depends on the specific method used and can range from $O(1)$ to $O(|E|^2)$. Sorting all the similarities takes $O(|E|^2 \log |E|)$ time in the worst case, assuming a comparison-based sorting algorithm like quicksort or mergesort.

Algorithm 1 then proceeds to merge the communities based on the ordered edge pairs. The time complexity of this step depends on the specific merging method used and can range from $O(1)$ to $O(|E|)$ depending on the merging strategy and any additional computations involved. Representing the merging process with a tree structure also takes $O(|E|)$ time.

Next, Algorithm 1 enters a while loop that continues until a partitioning density threshold is reached. The number of iterations in the loop can vary depending on the network's characteristics and convergence behavior. In the worst case, the loop can have a time complexity of $O(|E|)$ if each iteration involves computations that scale linearly with the number of edges.

The transformation of the edge pair-based tree graph into a node-based graph requires traversing the tree structure and constructing the new graph representation. The time complexity of this step can range from $O(|E|)$ to $O(|V|)$, where $|V|$ represents the number of nodes in the resulting node-based graph. Identifying communities and overlapped nodes depends on the specific method used and can range from $O(|V|)$ to $O(|E|)$ in time complexity.

In summary, the overall time complexity of the algorithm ranges from $O(|E|^2 \log |E|)$ to $O(|E|)$ in the worst case, depending on the specific methods used for similarity calculation, merging, and community identification. The input size and the specific characteristics of the smart healthcare network graph heavily influence the computational complexity of Algorithm 1.

Different from traditional node-based communities, we use a set of edges to represent the community as $C = \{e_{i,j} | e \in E\}$. In the set of edges, each edge should be linked to at least one other edge in this community. That means no independent edge is allowed. In this way, we can avoid a node in multiple communities at the same time. This is because of the edge-based community detection algorithm. The "overlapped nodes problem" is thereby addressed.

In Fig. 1, we show the correlation of overlapped nodes and node similarity. In node-based community detection methods, the identification of communities is typically based on the connectivity patterns among nodes in a network. These methods aim to partition the network into cohesive groups or communities, where nodes within a community exhibit strong interconnectivity while having fewer connections to nodes outside the community.

One challenge that can arise in node-based community detection is the problem of overlapped nodes. Overlapped nodes refer to nodes that belong to multiple communities simultaneously, blurring the boundaries between communities. This means that these nodes have significant connections to nodes in multiple communities, making it difficult to assign them to a single community without sacrificing the accuracy of the community detection process.

The presence of overlapped nodes can introduce complications and ambiguity in community detection results. It can lead to difficulties in accurately identifying and delineating the boundaries of the distinct communities within the network. Overlapping nodes can create overlaps between detected communities, causing them to merge or appear less cohesive than they actually are. This can impact the quality and meaningfulness of the community structure revealed by the detection method.

At first, we regard each edge as a community. After that, the edges are enrolled into different communities. The enrollment criteria are that edges share the same nodes with the first edge. The similarity of an edge pair (e_{ij}, e_{ik}) with a common node v_i is to consider the

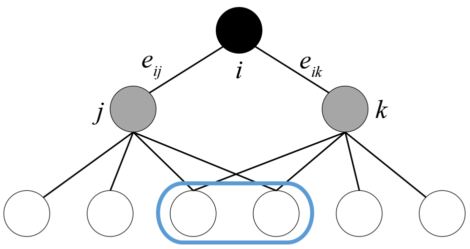


Fig. 1 Overlapped nodes and node similarity.

similarity of v_k and v_j. In the paper, we consider a concise but useful way, which is to evaluate the amounts of neighbors of u_k and u_j. Based on this methodology, we formulate the similarity of (e_{ij}, e_{ik}) as

$$S(e_{ij}, e_{ik}) = \frac{|V_+(k) \cap V_+(j)|}{|V_+(k) \cup V_+(j)|} \tag{1}$$

where $V_+(k)$ is the set of nodes V_k and all its adjacent neighbours, while $V_+(j)$ is the set of nodes V_j and all its adjacent neighbours.

Through calculating the edge pairs' similarity, it is able to detect the SHN community by clustering in a hierarchical manner. First, all similarity values of possible edge pairs are calculated. Then, the similarity values are ordered descendingly. A tree graph structure is then established to merge communities in an iterated way. In the iteration process, if there exist edge pairs that share the same similarity, they shall be merged in the same round. The convergence of community merging is controlled by a threshold. Otherwise, all the edges are merged into one single community.

To better explain, the edge pair similarity can be regarded as the strength of the merged community. This also relates to the height of a branch of the tree graph structure, as shown in Fig. 2. Therefore, to get reasonable communities, the key is to identify the best position to "cut" the tree, in particular, deriving the threshold of the community merging process. To avoid empirical errors, we establish an objective function called partitioning function, based on the density of all possible edge pairs.

Let M be the number of edge pairs inside a smart healthcare network, $|C|$ be the number of communities $\{C_1, C_2, \ldots, C_c\}$, m_c and n_c are the number of edge pairs and nodes inside community C_c, the corresponding normalization density is

$$P_c = \frac{m_c - (n_c - 1)}{n_c(n_c - 1)/2 - (n_c - 1)} \tag{2}$$

where $n_c - 1$ is the minimum number of edge pairs required to constitute a connected graph, and $n_c(n_c - 1)/2$ is the maximum number of possible edge pairs among n_c nodes. A special consideration is that $P_c = 0$ if $n_c = 2$. Thus, the partition density of the whole network is formulated as the weighted sum of P_c,

$$P = \frac{1}{M}\sum_c m_c P_c = \frac{2}{M}\sum_c \left[m_c \times \frac{m_c - (n_c - 1)}{(n_c - 2)(n_c - 1)}\right] \tag{3}$$

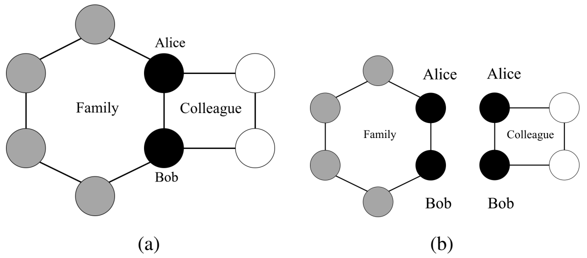


Fig. 2 Edge-based community vs. node-based community.

From Eq. (3), each term in the summation has a physical meaning within the community. Thus, the distinguishability limitation problem of modularity can be well mitigated. We can either directly optimize the partition density. The primary advantageous feature of this model is that we can flexibly partition the community based on the requirements. Another advantage of using edge pairs tree graph is to reveal the hierarchy community structure feature by non-optimization cut-off rule.

3.3 Mapping function derivation

To map the community density (trust) to a reasonable privacy protection level, a mapping function is required. The investigation shows that the Sigmoid function is a good match in this scenario. As a function that is used to evaluate the Quality of Service (QoS), we modify it to fit the proposed model. Community density is used as the input to generate personalized privacy protection levels, namely, the value of ϵ. The mapping function is referred to as QoS-based mapping function in this context.

The partitioning mechanism described above is used to generate several communities $C = \{C_1, C_2, \ldots, C_c\}$. Then we calculate the density of each community and further derive the personalized ϵ values. Apparently, the number and density of community may correspond to various users in SHN. The ϵ value should not linearly increase with the density as well. Thus, the sigmoid function works well in this scenario.

The reason why a sigmoid function is chosen is due to its unique features. To begin with, when the community density P_c is small (e.g., the most special case is a community formed by another single user), the privacy level should be high, and it increases slowly with the increase of the density. After the community has a certain scale P_c, the privacy level can be relaxed, and the relaxing rate is relatively high. However, when the community density P_c is large enough, the community is trusted in a high possibility, and the privacy level is relatively low. The impact of further relaxation is marginal, and thereby the increased scope slows down.

The modified mapping function is defined as

$$\epsilon_c = f(c) = \omega \times \frac{1}{1 + \exp(-\theta/P_c - \alpha)} \tag{4}$$

where ω is the weight parameter to adjust the amplitude of the maximum value, θ is leveraged to decide the steepness of the curve, and α denotes the location of the symmetric line.

Moving on, after Eq. (2) is substituted into Eq. (4), the sigmoid function is reshaped into

$$\epsilon_c = f(c) = \omega \times \frac{1}{1 + \exp\left(-\theta \times \frac{n_c(n_c-1)/2-(n_c-1)}{m_c-(n_c-1)} - \alpha \right)} \tag{5}$$

3.4 Community density-based personalized privacy protection in smart healthcare networks

Users who share the same interest or experience similar symptoms usually join the same community in an SHN. Besides, there is high possibility for them to form similar communities in other SHNs. Since the shared data over SHNs are highly sensitive, it is strictly necessary that other users can only access the processed data constrained by certain privacy protection methods.

The shared data are usually with various auxiliary information like location. As mentioned above, the community with higher density receives more accurate data, while the low-density community receives less accurate data, which is shown in Fig. 3. Alice belongs to two different communities. After executing Algorithm 1, personalized privacy protection levels are derived using Eq. (5). After that, the corresponding location information is shared to two communities with different accuracy based on different privacy protection levels. However, the shared data in different communities may be used to launch linkage attacks, and the linkage attack is discussed in the following subsections.

Theoretically, users that are more trustworthy will receive more accurate data. On the contract, users that are less trustworthy will receive less accurate data. To evaluate the trust among users, we use a simple and straightforward index, which is community density. Other indexes may be applicable in other scenarios. If the density is large, then people share a higher level of trust and vice versa. Then, a lower level of privacy protection will be acted on the raw data and people within this community can access reasonably accurate data.

Based on the classic differential privacy, we formulate the personalized differential privacy as follows.

Given $\epsilon \geqslant 0$, D to be the space of the sensitive data, D' to be D's adjacent dataset and the difference between them is one record, and $\mathcal{A} \subseteq D \times D$ to denote an adjacent relation. A mechanism is $\mathcal{M} \rightarrow \Delta(\mathcal{Y})$ considered to be ϵ-differentially private if

$$\Pr[\mathcal{M}(D) \in \Omega] = \exp(\epsilon) \times \Pr[\mathcal{M}(D') \in \Omega] \tag{6}$$

where $\mathcal{Y}$ is the noisy outcome, ϵ is the privacy protection level that varies with the community density, and Ω is the probability space.

The privacy protection level ϵ is defined in Eq. (5). Therefore, if we substitute it into Eq. (6), we have

$$\frac{\Pr[\mathcal{M}(D) \in \Omega]}{\Pr[\mathcal{M}(D') \in \Omega]} = \exp\left(\frac{\omega \times \exp\left(\theta \cdot \frac{n_i(n_i-1)/2-(n_i-1)}{m_i-(n_i-1)} + \alpha \right)}{1 + (1-\alpha)\exp\left(-\theta \cdot \frac{n_i(n_i-1)/2-(n_i-1)}{m_i-(n_i-1)} \right)} \right) \tag{7}$$

where the conditions of Eq. (7) inherits from Eq. (6). In the proposed personalized ϵ-differential privacy model, the privacy level ϵ is personalized by the density P_c through a sigmoid function.

In the personalized differential privacy model, ϵ is an

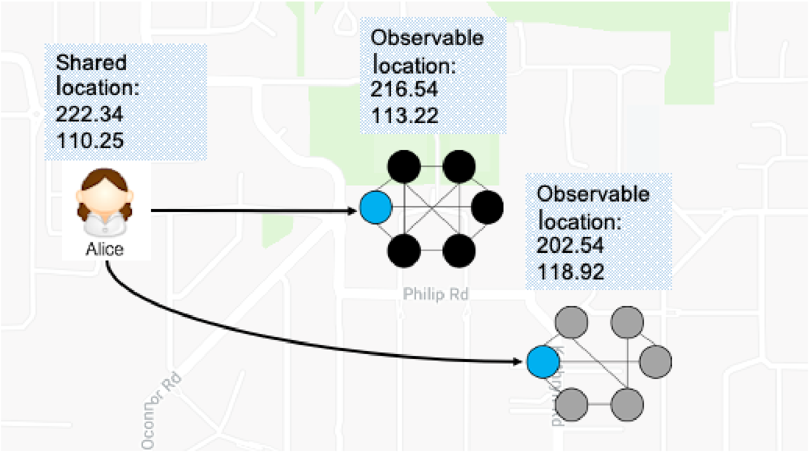

Shared location: 222.34 110.25
Observable location: 216.54 113.22
Observable location: 202.54 118.92
Alice
Philip Rd
Oconnor Rd

Fig. 3 Personalized privacy protection instance.

index to evaluate the privacy protection level and data utility since it decides the volume of the randomized noise. The overall privacy budget B is the total value of all possible ϵ. To measure the data utility, one of the most popular matrices in this scenario, Root-Mean-Square-Error (RMSE), is used to describe the balance between privacy protection and data utility[†],

$$RMSE = \sqrt{\sum_{i=1}^{n}\sum_{j\neq i}^{n} E||y_{ij} - d_{\text{raw}}||_2^2} \tag{8}$$

where d_{raw} is the raw data of the noisy outcome.

From the perspective of the optimized trade-off between privacy protection and data utility, given overall privacy budget B and minimum data utility $\min(RMSE)$, we have

$$\text{Optimize trade-off}: \max(\epsilon), \max(RMSE),$$

$$\text{s.t.,}$$

$$\epsilon_i = \frac{\omega}{1 + \exp\Big(-\theta \cdot \frac{n_i(n_i-1)/2-(n_i-1)}{m_i-(n_i-1)} - \alpha\Big)},$$

$$\sum_{i}^{n} \epsilon_i \geqslant B,$$

$$\sqrt{\sum_{i=1}^{n}\sum_{j\neq i}^{n} E||y_{ij} - d_{\text{raw}}||_2^2} \geqslant \min(RMSE) \tag{9}$$

3.5 Linkage attack model

In this subsection, we establish the adversary model and attack model. To qualitatively evaluate the adversaries and attacks, we model them using differential privacy mathematical theories.

For adversaries, we practically assume that they have a certain amount of background knowledge. The amount of background knowledge can be adjusted. The linkage attack is launched by such adversaries, who also has the access to multiple data sources. In SHNs, one of the most useful and easy-to-access background knowledge is the user connections (a sub-graph in the system model).

In the existing research, the modeling of the background knowledge of the adversary and linkage attack is barely discussed. As discussed above, the background knowledge can be regarded as raw data with noise, which is the same as differentially private noise in nature. Therefore, the linkage attack is the aggregation of mutliple pieces of raw data with different noises. Built upon this assumption, linkage attack is formulated as

Given multiple data resources $\{D_i \in D | i = 1, 2, \ldots, n\}$, a linkage attack $L(\cdot)$, and $\mathcal{M}$ being a randomized algorithm that sanitizes the dataset where $\epsilon_i = \mathcal{M}(D_i)$, the linkage attack is successfully launched if

$$\sum_{i}^{n} \mathcal{M}(D_i) \geqslant L(\cdot) \geqslant \max\{\mathcal{M}(D_i) | i = 1, 2, \ldots, n\} \tag{10}$$

Under this assumption, the success criteria of a linkage attack can be expressed as a release of differential privacy protection level. Quantitatively, the value of ϵ will increase as the attack result. To maximize the performance, we consider the worst-case linkage attack and the corresponding countermeasure, noise-decoupling mechanism is as follows.

3.6 Noise-decoupling mechanism

We start with two randomized algorithms, namely the Laplace mechanism and the exponential mechanism. Then, the noise-decoupling mechanism is given to improve the performances of the two algorithms.

3.6.1 Laplace mechanism and exponential mechanism

As mentioned above, ϵ-differential privacy is a probabilistic definition. It is necessary to design some randomized mechanisms which are differentially private. In this subsection, we will introduce two of the randomized algorithms, which are the Laplace mechanism and exponential mechanism.

In numeral scenarios, the Laplace mechanism is widely used to inject random noise under certain control. The noise generation compiles with a Laplace mechanism as

$$\text{Lap}\Big(\frac{\delta}{\epsilon}\Big) = \text{e}^{\Big(-\frac{||d_{\text{raw}}||_2 \times \epsilon}{\delta}\Big)} \tag{11}$$

We regard δ as privacy level ϵ. Although Gaussian noise can also be utilized to achieve differential privacy, it requires a slight relaxation of the definition of differential privacy. Therefore, we employ Laplace noise to deal with the numeral data published by users.

Besides the Laplace mechanism, we need to introduce an exponential mechanism to the proposed model as well. The reason is that the Laplace mechanism is limited in the scenario of numeral data. However, the exponential mechanism functions well to handle textual data.

Let K be the set of candidate items, $k_i \in K$ be a candidate item, $f(D, k)$ be the function that outputs the number of o_i inside D, a mechanism $M_f^{\epsilon}(D)$ is ϵ-differentially private if the probability of o_i being the output is proportion to $\text{e}^{\frac{\epsilon f(D,k)}{2\delta}}$,

[†] The data utility denotes how much useful information is left in the sanitized data. It is an important parameter to measure the effectiveness of privacy protection models.

$$\frac{\Pr\left[M_f^{\epsilon}(D)=k\right]}{\Pr\left[M_f^{\epsilon}(D')=k\right]} = \left(\frac{\exp\left(\frac{\epsilon f(D,k)}{2\delta}\right)}{\exp\left(\frac{\epsilon f(D',k)}{2\delta}\right)}\right)\left(\frac{\sum\limits_{o'}\exp\left(\frac{\epsilon f(D',k')}{2\delta}\right)}{\sum\limits_{o'}\exp\left(\frac{\epsilon f(D,k')}{2\delta}\right)}\right) \leqslant \exp(\epsilon) \tag{12}$$

In order to control the noise value, we introduce another term, which is global sensitivity. Assume a function $f: D \rightarrow K^d$, we have an input dataset, and the output should be a d dimensional real-valued vector. For any adjacent datasets D and D', the global sensitivity is defined as

$$G_f = \max_{D,D'} ||f(D) - f(D')|| \tag{13}$$

The global sensitivity can be applied to both the Laplace mechanism and exponential mechanism, and helps to determine privacy and accuracy.

3.6.2 Decoupling the correlation among noises

For n-tuple real-valued data d_{raw}, we aim to propose a private mechanism $\mathcal{M}$ to generate the approximation y_{ij}, in which y_{ij} is sent from u_i to u_j. As shown in Algorithm 2, two features are required for the mechanism $\mathcal{M}$. Firstly, the absolute error $|y_{ij} - d_{\text{raw}}|$ is supposed only to be determined by the density of the community. The rest of the arguments should have no impact on the absolute error. Secondly, the linkage of any series of data resources will not reveal further sensitive information about the individual. The workflow of the proposed attack-proof personalized differential privacy is shown by the pseudo-code in Algorithm 2.

Algorithm 2 Attack-proof personalized differential privacy

```
Input: Raw data $D$
Output: Sanitized dataset with personalized privacy protection
Derive communities and community densities $P_c$;
Set up proper sigmoid function $\frac{1}{e^x+1}$;
if $u_i$ belongs to a single community $C_c$ then
    Personalize privacy level $\epsilon_i$ based on $P_c$;
else
    Personalize privacy level $\epsilon_i$ based on min($P_c$);
end if
if data $D$ is numeral then
    Choose Laplace mechanism;
else
    Choose exponential mechanism;
end if
Deploy noise decouple mechanism;
Deploy data utility optimization mechanism;
Release sanitized dataset to different communities $C = \{C_1, C_2, \ldots, C_c\}$.
```

Algorithm 2 takes raw data D as input and aims to generate a sanitized dataset, with personalized privacy protection. It begins by deriving communities and computing community densities D_c from the raw data. A proper sigmoid function is set up to calculate personalized privacy levels for individual data points. Algorithm 2 then determines the privacy level ϵ_i based on whether a data point u_i belongs to a single community or multiple communities. For numerical data, the Laplace mechanism is chosen to add privacy-preserving noise, while for non-numerical data, the exponential mechanism is used. Algorithm 2 employs noise decoupling and data utility optimization mechanisms to enhance privacy and maintain data quality. Finally, the sanitized data is released to different communities, $C = C_1, C_2, \ldots, C_c$, considering the personalized privacy levels and community memberships of the data points. The output is a privacy-protected and community-tailored sanitized data set.

Algorithm 2 starts by deriving communities and community densities from the raw data D. The time complexity of this step depends on the specific community detection method employed and can range from $O(e^2)$ to $O(e^3)$, where e is the number of edges in the raw dataset D. Community detection often involves repetitive processes, navigating through graphs, or utilizing optimization algorithms, all of which add to the complexity of the undertaking.

Next, Algorithm 2 sets up a sigmoid function, which has a constant time complexity of $O(1)$ as it involves defining a mathematical function. Then it checks whether a data point e_i belongs to a single community or multiple communities. This step has a constant time complexity of $O(1)$ as it involves a simple conditional check.

Based on the membership status, Algorithm 2 personalizes the privacy level ϵ for the data point. The time complexity of this step depends on the computation involved in assigning a personalized privacy level based on the community densities D_c. It can range from $O(1)$ to $O(|C|)$.

Algorithm 2 proceeds to choose either the Laplace or exponential mechanism for privacy preservation. This step has a constant time complexity of $O(1)$ as it involves selecting between two predefined mechanisms.

Next, Algorithm 2 deploys the noise decouple mechanism to protect privacy. The time complexity of this step depends on the specific method used for noise decoupling and can range from $O(e)$ to $O(e^2)$. Complex computations or iterative processes may be involved in this step.

Similarly, Algorithm 2 deploys the data utility optimization mechanism, which aims to balance privacy and data quality. The time complexity of this step depends on the specific optimization methods used and can range from $O(e)$ to $O(e^2)$, depending on the size of the data and the complexity of the optimization process.

Finally, Algorithm 2 releases the sanitized and privacy-protected data to different communities. This step has a constant time complexity of $O(1)$, as it involves the release of data to predefined communities.

In summary, the overall time complexity of the algorithm depends on the specific methods used for community detection, privacy level personalization, noise decoupling, and data utility optimization. The time complexity ranges from $O(e^2)$ to $O(e^3)$ for community detection, and the other steps generally have time complexities ranging from $O(1)$ to $O(e^2)$, depending on the size of the data and the specific computations involved in each step.

In Algorithm 2, to achieve the targets, we generate the noises defined on a private stochastic process, which is designed and discussed in detail as below.

Let ϵ_i, ϵ_{i+1}, ϵ_{i+2} be three instances of privacy protection level. Given $\epsilon_i < \epsilon_{i+1} < \epsilon_{i+2}$, the following properties are required for the private stochastic process.

- The noise complies with Laplacian mechanism: $\forall \epsilon > 0, \mathrm{d}[\Pr\left(V_\epsilon = v\right)] \propto \mathrm{e}^{(-\epsilon||v||_2)}$;
- The noise generation process complies with the private stochastic process: $\epsilon_i < \epsilon_{i+1} < \epsilon_{i+2}$, $V_{\epsilon_i} | V_{\epsilon_{i+1}}$, $V_{\epsilon_i} \perp V_{\epsilon_{i+2}}$;
- The transfer probability in the Markov process is

$$\mathrm{d}\left[\Pr\left(V_{\epsilon_i} = v_i | V_{\epsilon_{i+1}} = v_{i+1}\right)\right] \propto \delta(v_i - v_{i+1}) + \frac{(n+1)\epsilon_i^{1+\frac{n}{2}} ||v_i - v_{i+1}||_2^{1-\frac{n}{2}}}{(2\pi)^{\frac{n}{2}}} \times \mathrm{Bessel}_{\frac{n}{2}-1}(\epsilon_i ||v_i - v_{i+1}||_2)\tau + O(\tau^2),$$

s.t.,

$$\tau = \frac{\epsilon_i}{\epsilon_{i+1}} - 1 \tag{14}$$

where Bessel ( ) is a Bessel function.

4 Blockchain-Enhanced Mechanism Against Data Falsification

In this part, we present how the tailor-made blockchain structure can guarantee the integrity of differentially private health data and prevent data falsification operations.

4.1 Consortium blockchain-based smart healthcare network

The smart healthcare network has been modeled as a graph structure as above. For each node in the graph, it is also a node in the proposed consortium blockchain system as well, as shown in Fig. 4, from which we can tell there are different entities in SHNs, including but not limited to patients, hospitals, health bureaus, etc., different entities have different accesses. For example, the health bureau has the most access due to its supervision role, but it does not have the right to revise the data stored on-chain. Hospitals have access to the health history of patients and are allowed to add new items to the history. Patients can only access their own data with the rights of adding, deleting, or revising[42]. Except for this, all parties conduct data sharing which dynamically changes the access during the operation of the system

As mentioned earlier, doctors and patients can manage the data and thereby will generate blocks of the consortium blockchain. Usually, the block will include some sensitive information, like diagnosis notes, identity information, time stamp, location, etc. The privacy protection techniques of the data stored on the consortium blockchain are a necessity and will be discussed in the subsequent subsections.

To make it more secure and robust, the Proof-of-Work (PoW) consensus algorithm is deployed. This requires miners to mine for a nonce value and get the block generation chance. The data are firstly broadcast to all eligible parties and all parties start to calculate the

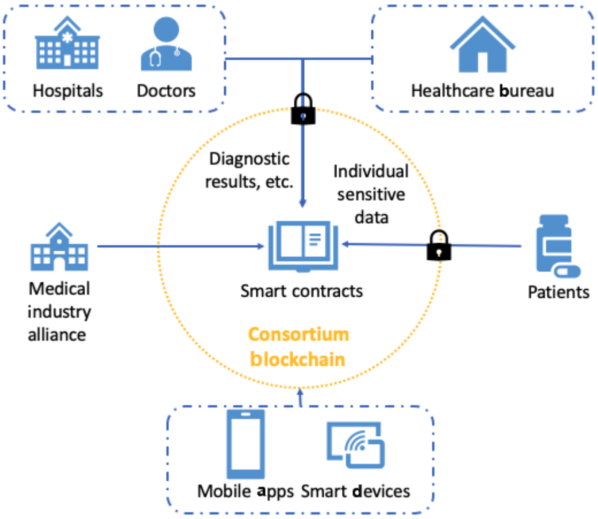


Fig. 4 Consortium blockchain based smart healthcare network.

nonce value. The party that first finds the nonce will generate a candidate block, after which, the block is broadcast to all eligible parties again. The parties who receive a candidate block of this round stop mining and validate the data in the block. If the data are authentic, the block will be appended to the local chain. As a consortium blockchain, the health bureaus will serve as the leader of the chain to help with the consensus process. We can always add more layers for more functionalities. For example, an analytic layer could be used for disease surveillance[20]. In such a consortium of blockchain-based smart healthcare networks, several privacy requirements are to be met, including data integrity and interoperability, specifically for healthcare research facilitation.

4.2 Blockchain brief overview

We show the basic PoW-based blockchain structure in this subsection, which is the foundation of the proposed structure.

In Fig. 5, a generalized structure of blockchain systems for medical data sharing in smart healthcare networks is presented. This structure demonstrates how blocks are appended to each other, ensuring a sequential order of transactions. Each block includes the hash value of the previous block, creating a chain-like structure that ensures the integrity and immutability of the data.

Within each block, transactions are stored in a Merkle tree structure. The Merkle tree allows for efficient summarization and verification of the transactions contained within the block. By hashing the individual transactions and then combining them in pairs until a final hash value, known as the Merkle tree root, is computed, the block can represent a condensed representation of all the transactions it contains.

In this particular context, a transaction refers to a piece of sanitized medical data that adhere to personalized differential privacy. Sanitization techniques are applied to the medical data to remove personally identifiable information and ensure individual privacy. Personalized differential privacy ensures that the level of privacy protection is tailored to the specific requirements and preferences of each patient or data subject.

By utilizing blockchain technology in the sharing of medical data, the structure depicted in Fig. 5 offers several benefits. It enhances data security, as the hash values and the chaining mechanism make it extremely difficult for unauthorized parties to tamper with or modify the stored data. The Merkle tree structure facilitates efficient and secure verification of the integrity of transactions within a block. Additionally, the use of personalized differential privacy techniques helps protect the privacy of individuals while enabling the sharing and analysis of aggregated and anonymized medical data.

Overall, the blockchain system presented in Fig. 5 provides a robust framework for secure and privacy-preserving medical data sharing in smart healthcare networks, ensuring data integrity, tamper resistance, and individual privacy protection.

In Table 1, an instance of a generalized block header is shown with two key components. The block header usually contains a version of the block, parent block

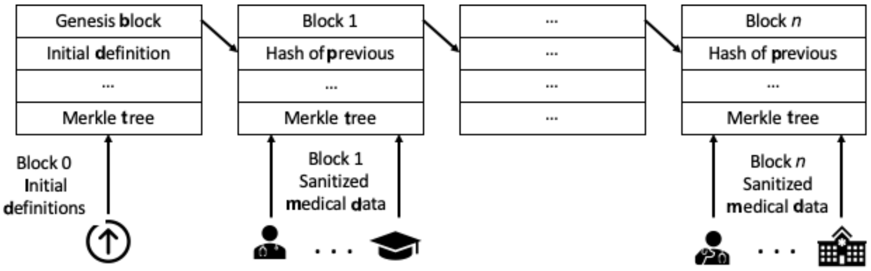


Fig. 5 Blockchain architecture for sanitized sensitive medical data sharing.

Table 1 Instance of block header in the blockchain system.

Element	Example value
Block version	03000000
Parent block hash	c5ee0a2b1480a2852a30d5ffe364d98e10d9334beb48ca0d000000000000000
Merkle tree root	8c2de567cad23df7992e030b44af454d70add80201edcd21cbb940ab88da45c
Timestamp	29d5a5a4
nBits	34cd1b29
Nonce	fd8e2664

hash, Merkle tree, timestamp, nBits, and Nonce (for Proof-of-Work). The size of the block determines the maximum number of transactions (shared data). The shared data are usually protected and validated by asymmetric cryptography in a trustworthy case. But if there is an untrustworthy environment, other techniques may be deployed, such as a digital signature. In the block body, it contains sanitized heath data or any other data types regarding the application scenarios.

- **Version of block**: The block version is a numeric value that represents the version of the blockchain protocol being used. It helps ensure compatibility between different versions of the blockchain software. The version number may be updated over time as new features or improvements are introduced to the protocol.
- **Parent block hash**: The parent block hash is a unique identifier for the previous block in the blockchain. It is the hash value of the header of the previous block. By including the parent block hash in the current block header, the blockchain maintains a chronological order and creates a chain of blocks. This chaining mechanism ensures the immutability and integrity of the blockchain.
- **Merkle tree**: The Merkle tree root is a hash value that represents a condensed summary of all the transactions within the block. In a blockchain, transactions are grouped together in a Merkle tree structure. The Merkle tree allows for efficient verification of the integrity of all the transactions in the block. The root hash is calculated by hashing the concatenated hash values of the individual transactions in a specific order until a single hash value remains.
- **Timestamp**: The timestamp indicates the time when the block was created or mined. It is typically represented as a Unix timestamp, which is a numerical value that represents the number of seconds elapsed since January 1, 1970. The timestamp helps establish the order of blocks in the blockchain and ensures that blocks are added at regular intervals.
- **nBits**: The nBits value represents the target difficulty for mining the block. Mining is the process of finding a nonce value that, when combined with other block data, produces a hash value below a certain target difficulty. The nBits value encodes the target difficulty, which determines the computational effort required to mine a block. Miners adjust the nBits value periodically to maintain a consistent block generation rate.
- **Nonce**: The nonce is a random value that miners modify during the mining process in order to find a suitable hash value that satisfies the target difficulty. Miners repeatedly change the nonce and recompute the block hash until they find a nonce that, when combined with other block data, produces a hash value that is below the target difficulty. The nonce is a crucial component of the proof-of-work consensus algorithm, which ensures that mining requires computational effort and contributes to the security of the blockchain.

These elements, combined together, form the block header. The block header is hashed to produce the block's unique identifier, which is used in the blockchain's consensus algorithm to validate and add the block to the blockchain.

The connection among blocks ensures the integrity and immutability of the blockchain. Any change to a block's data would alter its hash, making it inconsistent with the stored parent block hash in the subsequent block. This would break the chain and invalidate the affected block, alerting the network to potential tampering attempts. Therefore, altering the data in one block would require recalculating the hash for that block and all subsequent blocks, which becomes increasingly computationally expensive and practically infeasible as the blockchain grows longer.

This linking mechanism provides several important benefits. **Security**: The chain of blocks ensures the security of the blockchain by preventing unauthorized modifications. Once a block is added to the blockchain, it becomes extremely difficult to alter any past blocks without the consensus of the majority of network participants. **Immutability**: The chaining of blocks creates an immutable record of transactions. Once a block is added to the blockchain, its contents are effectively set in stone. This feature is valuable in applications where data integrity and auditability are critical. **Consensus**: The connection between blocks plays a crucial role in achieving consensus among network participants. By following the longest valid chain of blocks, participants can agree on the order of transactions and determine the valid state of the blockchain.

4.3 Importance of defense against data falsification

In all data-sharing scenarios or services, the data falsification issue is an inescapable topic due to its significant negative impact. For instance, one of the primary attacks in this domain, poisoning attacks, is usually launched by adversaries. To poison the raw data, adversaries may inject or replace the data with misleading features. This can bring catastrophe to smart healthcare applications since the data are directly related to human health. To mitigate the risks, we explain the devised approach in the following paragraphs.

On the tailor-made blockchain, the differentially private data (raw data + differentially private noise) are saved with certain access control. When an individual queries the blockchain, the differentially private data will directly used as the response to the querier. However, since access control is deployed, an individual can only access the data within the community but an individual can belongs to multiple communities as described above. Within a community, all members share a same privacy protection level and can access the same data. Different from traditional blockchain systems, it is partitioned into many consortium sub-chains, with which community members only need to maintain their corresponding sub-chain. This brings more efficiency and flexibility for smart healthcare network users.

Despite this, adversaries will still try to falsify the differentially private data stored on the sub-chains. The adversary may inject or revise the data to gain financial benefits. However, all behaviors will be cross-validated by community members of each sub-chain. In this case, if there is a malicious operation on data, most trustworthy members will choose not to act on it and thereby the operation cannot be performed. The raw data will remain unchanged in this case. At the same time, smart healthcare data sharing will not be as frequent as transaction systems, which will not bring in too much computation burden or processing delay. The sub-chain structure also can improve the efficiency.

5 Performance Evaluation

This section will illustrate the experimental settings and results to validate the effectiveness of the proposed solution. The proposed model is Community-based Differential Privacy (C-DP), while the two benchmark models are classic Differential Privacy (DP) and Personalized Differential Privacy (P-DP). P-DP uses the virtual online distance as the personalization index. To evaluate the whole system, we evaluate privacy protection performance, data utility degree, community distribution similarity, and blockchain performance.

We use two real-world datasets for the experiments, which are Doximity dataset and HealthTap dataset[43, 44]. Doximity dataset is collected from its developer API following a uniform distribution[43]. Doximity is a popular health social network that offers abundant functions, like befriending, news publishing, data sharing, etc. The details of the obtained data are as follows. Doximity is one of the very few popular social networks for doctors that provides developer API, but it does not allow broad querying of its database. Therefore, we randomly obtained 2122 nodes and 14 389 edges with an average degree of 4.39. The collected data has been anonymized by Doximity, but the nodes and edges relationship are stored for research purposes. In addition, we also build an anonymous doctor social graph via the API of HealthTap[44]. Health information of the HealthTap is provided interactively by a network of nearly over 150 000 licensed doctors. It also provides the functionality of peer review, which includes doctors rating each other and self-identifying specializations. From the HealthTap, we obtain a total of 1325 nodes consisting of a network of 5231 edges.

In this work, when we talk about DP, it is the classic differential privacy that provides uniform privacy protection to all users. In the case of P-DP, it is personalized privacy that leverages virtual online distance to personalize privacy levels. For C-DP, which is the proposed model, it maps community density to personalized privacy protection levels.

5.1 Community density similarity

In order to utilize community density as the index, we firstly use evaluation results to demonstrate the similar distribution of the two datasets. The outcome shows that both the community number distribution and the overlapped nodes distribution are quite close, which verifies our idea of mapping density to privacy level.

In Fig. 6a, we compare the density with the node number. The distributions are quite similar to each other. Of all the records, there is a noisy point where node number equals 1. When node number is 1, we can easily tell that they have no mutual trust people in this smart healthcare network. The amount of this node is normally large, which captures the real-world features.

In Fig. 6b, we illustrate the distributions of overlapped nodes. Both of the numbers are decreasing with the number of overlapped communities increasing. They share the same trends and similar numbers while the maximum difference is no larger than 2. The x-axis starts from 2 since one node in one community is not defined as overlapping. In Fig. 6b, we have tried to discuss one advanced feature with these two datasets, which shows the total amount of each group regarding the inclusion of overlapping nodes. Intuitively, any node

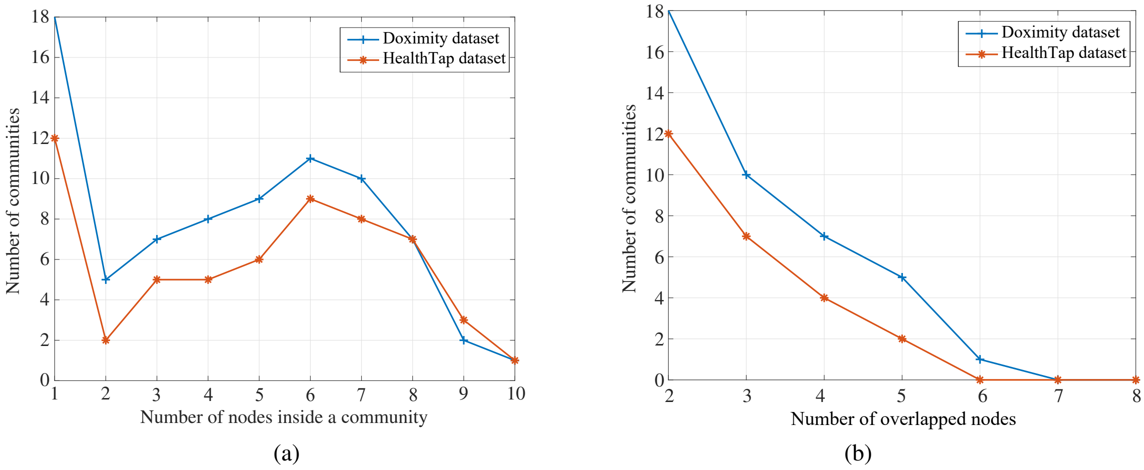


Fig. 6 Community similarity distribution in the smart healthcare network.

can be assigned to many groups. Therefore, we show that a negative correlation between these two parameters. Besides, for the given datasets, we can observe that the existence of the community is questionable when the number of nodes is than more 6.

The comparison of density with the node number (representing the number of nodes in a community) is shown in Fig. 6. The distributions of density and node number are quite similar, indicating that communities in the smart healthcare network tend to have consistent densities regardless of their size. However, there is a notable noisy point where the node number equals 1. This implies that there are instances where a community consists of only one node, indicating a lack of mutual trust connections within that community. The occurrence of such single-node communities is relatively common, which aligns with real-world characteristics of smart healthcare networks.

The numbers of overlapped nodes decrease as the number of overlapping communities increases. This trend suggests that as nodes are assigned to more communities, the likelihood of them belonging to multiple communities decreases. The distributions of the two numbers (overlapped nodes and the number of overlapping communities) show similar patterns, with a maximum difference of no more than 2. This indicates a consistent relationship between the two parameters. Additionally, the x-axis starts from 2 in Fig. 6b because one node in one community is not considered overlapping. The analysis also delves into an advanced feature, discussing the total number of groups considering the inclusion of overlapping nodes. It is observed that there is a negative correlation between the total number of groups and the presence of overlapping nodes. This implies that when nodes are assigned to multiple groups, the existence or validity of the community structure becomes questionable, particularly when the number of nodes in a community exceeds 6.

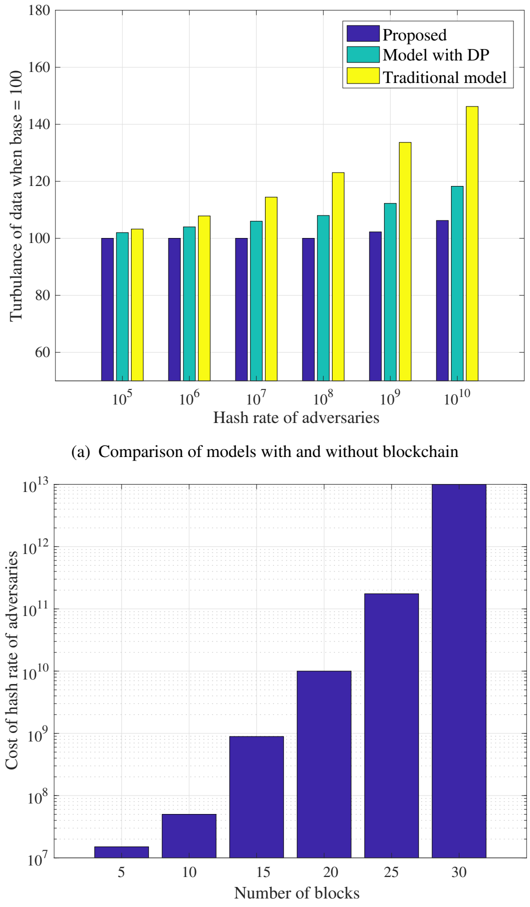
Overall, the analysis of the results highlights the characteristics of the smart healthcare network. It shows the distribution patterns of density, node numbers, and overlapped nodes, shedding light on the structure and trust relationships within the network. The presence of single-node communities suggests a lack of connections and trust, while the decreasing number of overlapped nodes indicates less overlap as nodes are assigned to more communities. These insights provide valuable information for understanding the community structure and dynamics within the smart healthcare network.

5.2 Evaluations on the security level of blockchain

In this section, we discuss how the aforementioned attacks are resistant to a satisfyingly high degree by using the modified blockchain structure. To better compare the performance, we establish two personalized privacy protection models, one with the blockchain while the other without. The proposed blockchain structure is based on a PoW consensus algorithm. Thus, a large number of blocks means it is almost impossible to launch attacks because it costs too much hash rate. In the following context, we consider a start-up blockchain with a limited amount of blocks. In this case, the adversaries may have the incentive to mount relevant attacks.

To compare the performances of the three models, we establish a coordinate system where the semi-logarithmic x-axis denotes the required hash rate and the linear y-axis is the turbulence of data when the exponential base is 100. As shown in Fig. 7a, with the increase of the hash rate of the adversary, the turbulence grows correspondingly for all three cases. To successfully launch attacks, the hash rate of adversaries should pass a specific threshold. Usually, in PoW-based consensus blockchain systems, the threshold is believed to be 50%. If the adversary's hash rate exceeds more than half of the total hash rates, it results in the successful execution of the attack and grants the adversary control over the blockchain system. As indicated in our simulation, there are only 10 blocks in total. Apparently, in practice, there are hundreds of, or even thousands of blocks, appending one after another, and it needs unimaginable amount of computing power to make the attacks happen.

In order to show how much hash rate is required for launching the attack, we establish a coordinate system where the semi-logarithmic *y*-axis denotes the required hash rate and the linear *x*-axis is the number of blocks. From Fig. 7b, it is intuitive that the required hash rate grows in an exponential manner with the increase of block number (shows linearly because of the semi-logarithmic *y*-axis). It is worth mentioning that the hash rate demand breaks through 10^{13} when the block number is 30. Usually, the high hash rate will demotivate most adversaries. Although there are adversaries with extremely high hash rates, they may not benefit from attacking such a blockchain. Moreover, in PoW-based consensus blockchains, the mining difficulty lifts with rounds, and thereby the protection is enhanced with more blocks appending to the blockchain.



(a) Comparison of models with and without blockchain

(b) Successful attack requirements with respect to the number of blocks

Fig. 7 Performances comparison with blockchain.

The performance of three models is compared using a coordinate system where the *x*-axis represents the required hash rate (semi-logarithmic scale) and the *y*-axis represents the turbulence of data when the base is set to 100. The results depicted in Fig. 7a show that as the hash rate of the adversary increases, the turbulence of the data grows accordingly for all three cases. This indicates that to successfully launch attacks, the hash rate of the adversary must surpass a specific threshold. In most PoW based consensus blockchain systems, this threshold is commonly believed to be 50%. This means that if the adversary's hash rate exceeds half of the total hash rates in the network, they can successfully attack and take control of the blockchain system. However, in the simulation presented, there are only 10 blocks in total, which is significantly smaller than the typical number of blocks in practical scenarios. It is important to note that in real-world scenarios with hundreds or even thousands of blocks appending one after another, launching successful attacks would require an unimaginable amount of computing power.

A coordinate system is established to demonstrate the required hash rate for launching an attack. The *y*-axis represents the required hash rate (semi-logarithmic scale) and the *x*-axis represents the number of blocks. From Fig. 7b, it is evident that the required hash rate increases exponentially as the number of blocks increases (appearing linearly due to the semi-logarithmic *y*-axis). It is worth mentioning that the hash rate demand surpasses 10^{13} when the number of blocks reaches 30. Typically, a high hash rate requirement discourages most adversaries. Even if there are adversaries with extremely high hash rates, they may not find it beneficial to attack such a blockchain. Additionally, in PoW-based consensus blockchains, the mining difficulty increases with each round, resulting in enhanced protection as more blocks are appended to the blockchain.

Overall, the analysis of the results highlights the relationship between the required hash rate, number of blocks, and the security of the blockchain system. Increasing the hash rate of the adversary leads to higher turbulence in the data and increases the risk of successful attacks. However, the practical feasibility of launching such attacks is severely constrained by the vast computing power required, especially in scenarios with a large number of blocks. The increasing hash rate demands with the number of blocks also act as a deterrent for adversaries, while the mining difficulty mechanism in PoW-based blockchains adds an additional layer of protection as more blocks are added.

5.3 Evaluations on the blockchain against poisoning attacks

To validate the performance of poisoning attack proof, we compare the three models, in particular, classic differential privacy, personalized differential privacy, and blockchain-assisted personalized differential privacy. With 50 times experiments, the comparison of actual results and poisoned results are evaluated by Average Absolute Error (AAE). In each round, we practically assume that an attacker poisons 20% of the whole dataset. Besides, the user number is used as the x-axis. From Fig. 8, if there are over 30 users in a community, the AAE value of the blockchain-assisted model converges to 0. This indicates the poisoning attack is totally mitigated.

This is because of a practical assumption of blockchain, in particular, a sufficiently large community. If an adversary hopes to poison the data on-chain, he/she needs over 50% of the nodes to agree on through consensus. In most public blockchains running nowadays, there usually have a sufficiently large community to maintain the chain, for example, Ethereum. Usually, the assumption holds since the majority of the members agree on the same benefits. In the experiments, the nodes may become malicious with a chance of 50% or 0% before or after receiving any rewards. Then, we have the observation that the poisoning attacks are mitigated after the user number in a community passes 30.

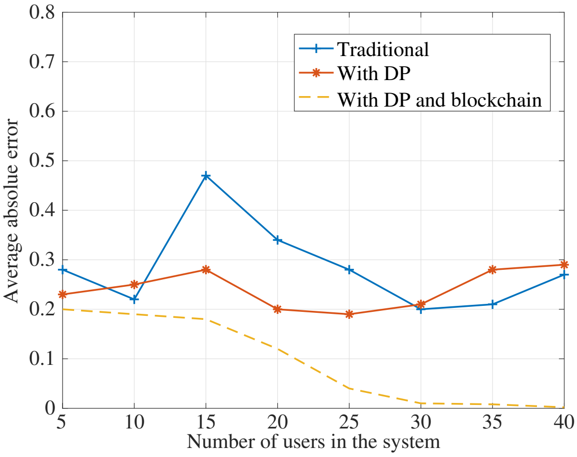
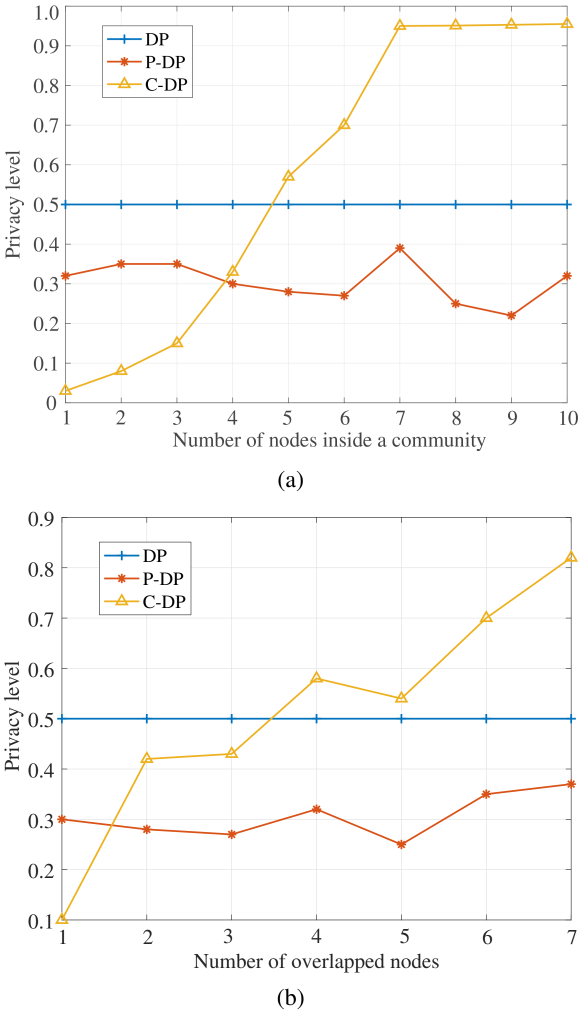


Fig. 8 Performance comparison with DP and blockchain.

5.4 Privacy protection measurement

The privacy protection level is the most important index for privacy protection models. The most primary characteristic of personalized privacy protection is flexible and directional.

In Fig. 9a, we can conclude that for DP, the privacy level maintains the same all the time. For P-DP, although the privacy level fluctuates a little bit, it is not practical because it takes all the users into consideration, including people who are not direct trustees. In C-DP, we only consider the users who are direct trustees. The direct



(a)

(b)

Fig. 9 Privacy protection measurement from perspectives of personalized privacy and overlapped nodes.

friends are divided by the community density, and the privacy level complies with sigmoid function trends.

In Fig. 9b, the overlapped nodes play an important role as we have to decide the privacy level of overlapped nodes when they are in different groups with various densities. We provide the lowest level based on the communities it involves. Therefore, we can observe a fluctuation in the privacy protection level. If the node is inside five communities, but all the communities have a relatively low density, it may have a relatively high privacy level and vice versa.

The results depicted in Fig. 9a demonstrate the behavior of different privacy preservation approaches: DP, P-DP, and C-DP. In the case of DP, the privacy level remains the same throughout the entire duration. This indicates a consistent and fixed level of privacy protection, which may not be tailored to individual users' specific requirements. For P-DP, although the privacy level fluctuates slightly, it takes into account all users, including those who are not direct trustees. This approach may not be practical as it lacks precision in privacy level assignment. On the other hand, C-DP focuses only on users who are direct trustees. The privacy level is determined based on the community density, and it follows the trends of a sigmoid function. This approach provides a more tailored and fine-grained privacy protection level, aligning with the characteristics of the communities and ensuring more personalized privacy preservation.

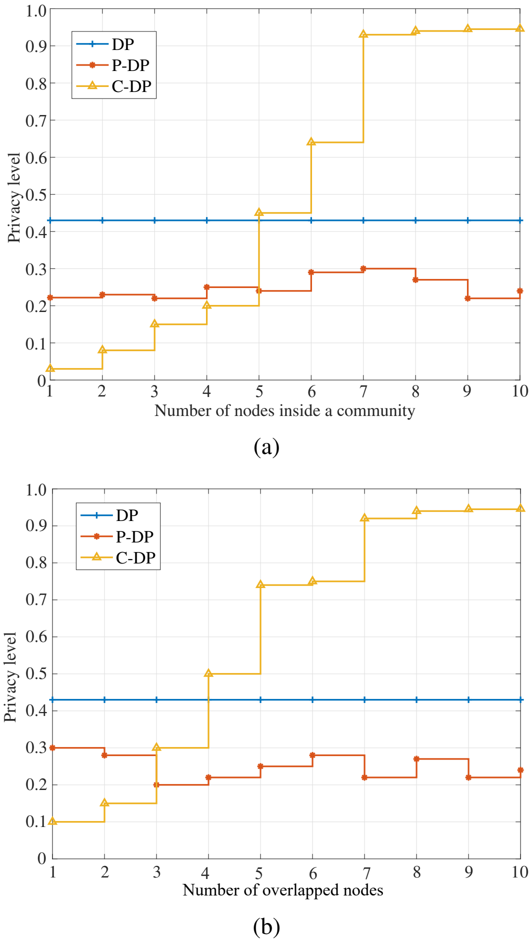
The analysis is centered around the role of overlapped nodes and their impact on privacy levels when they belong to different groups with varying densities. Figure 9b presents the results. It is observed that the privacy protection level exhibits fluctuations due to the presence of overlapped nodes. The privacy level determination takes into account the communities in which the node is involved. If a node is part of multiple communities but all those communities have relatively low densities, it may have a higher privacy level. Conversely, if the node is involved in several communities with higher densities, it may have a lower privacy level. This fluctuation reflects the consideration of community characteristics and the varying influence of overlapping memberships on privacy preservation.

Overall, the analysis of the results highlights the differences and implications of various privacy preservation approaches. DP provides a fixed level of privacy, P-DP lacks precision, and C-DP offers personalized privacy levels based on community densities. Additionally, the presence of overlapped nodes introduces variations in privacy levels depending on the densities of the communities they belong to. These insights emphasize the importance of tailoring privacy preservation to individual users' needs and considering the dynamics of community structures in determining privacy levels for optimal privacy protection.

5.5 Data utility comparison

Data utility directly affects the quality of service of users. Therefore, we can not sacrifice too much data utility to achieve additional privacy load. Personalized privacy protection provides flexible data utility to different users, which can provide high-quality service to specific users.

Figure 10a leverages a stairs chart to show the data utility increases with the increase of the ϵ, which is also known as privacy level. In terms of the overlapped nodes data utility in Fig. 10b, we can see that the fluctuating trend is similar to the privacy protection level as well.



(a)

(b)

Fig. 10 Data utility measurement from perspectives of personalized privacy and overlapped nodes.

Thus, the trade-off between privacy protection level and data utility is nicely derived.

Figure 10a presents a stair chart illustrating the relationship between data utility and the privacy level (represented by ϵ). The chart demonstrates that as the privacy level (i.e., ϵ) increases, the data utility also increases. This indicates that a higher level of privacy protection (achieved by increasing ϵ) is associated with a higher level of data utility. The stair-like pattern suggests that there are distinct increments in data utility as the privacy level is adjusted, rather than a continuous and smooth progression. This implies that privacy preservation mechanisms are capable of striking a balance between protecting privacy and maintaining useful data.

Figure 10b depicts the data utility of overlapped nodes. The fluctuating trend observed in the data utility aligns with the privacy protection level. This implies that as the privacy level varies (e.g., influenced by community densities or specific privacy mechanisms), the data utility of overlapped nodes follows a similar fluctuating pattern. The trade-off between privacy protection level and data utility is clearly demonstrated, suggesting that increasing privacy protection measures may come at the cost of some loss in data utility. However, the fluctuating trend indicates that there might be opportunities to optimize the privacy-utility trade-off by fine-tuning the privacy mechanisms or adjusting community characteristics.

Overall, the analysis of the results highlights the relationship between privacy protection level, data utility, and the influence of overlapped nodes. It reveals that increasing the privacy level tends to improve data utility, indicating that privacy preservation measures can be designed to strike a balance between privacy protection and maintaining valuable data. The fluctuating trend in data utility and its similarity to the privacy protection level emphasizes the trade-off between privacy and utility, presenting opportunities for further optimization in privacy mechanisms. These findings contribute to understanding the interplay between privacy and data utility in the context of the depicted scenarios.

5.6 Evaluation on the computation overhead of optimization for trade-off

In Fig. 11, we have demonstrated the computation overhead of the optimization for the trade-off. We evaluated the change of computation overhead regarding the number of communities inside a healthcare network. Since there is a personalized mapping function, the increment of a number of communities costs less and less computation power. If the number of communities is great enough, the computation overhead will converge to a specific value, which testifies the scalability of the proposed model in this big data era.

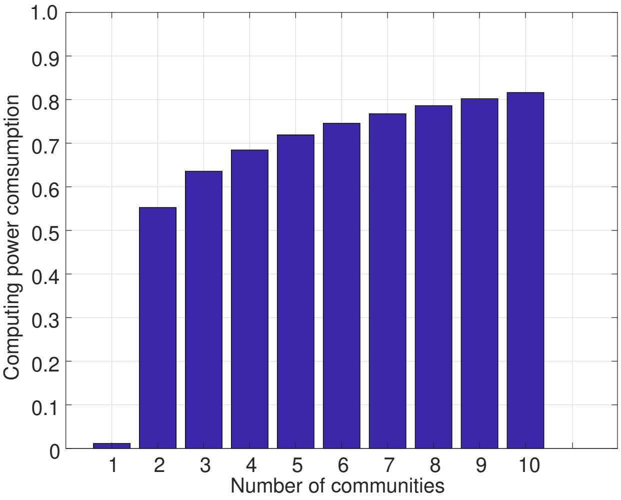


Fig. 11 Evaluation on the computation overhead of optimization for trade-off.

6 Conclusion

In this article, we start by describing the vulnerabilities of SHNs, including privacy leakages concerns, trade-off issues, as well as linkage and poisoning attacks. To solve the problems, we devise a personalized and trustworthy privacy protection model considering the trust measured by community density. By clearly defining the community, we then personalize the privacy protection levels constrained by differential privacy. To mitigate linkage attacks, a noise correlation uncoupling mechanism is proposed. In this case, even conducting linkage attacks, no more sensitive information can be obtained, which wipes out the incentive of attackers. Meanwhile, the integration of blockchain can defeat data falsification attacks. We perform corresponding experiments and the results confirm its effectiveness.

It is worth noting that in the proposed structure, the community detection is an important procedure but not easy to control the granularity. In addition, the theoretical foundation of the optimized trade-off should be further clarified.

Future work in progress includes establishing the model using game theory, which can better describe the confrontation of data holders and adversaries. This will help with deriving a better balance between privacy and data utility. Besides, the integration of federated learning to enhance privacy protection in this scenario is ongoing.

Acknowledgment

This paper was supported by the National Key Research and Development Program of China (No. 2021YFF0900400).

References

[1] D. Garcia, Leaking privacy and shadow profiles in online social networks, *Sci. Adv.*, vol. 3, no. 8, p. e1701172, 2017.

[2] M. S. Hossain, C. Xu, Y. Li, J. Bilbao, and A. El-Saddik, Advances in next-generation networking technologies for smart healthcare, *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 14–15, 2018.

[3] Y. Wang, A. Zhang, P. Zhang, Y. Qu, and S. Yu, Security-aware and privacy-preserving personal health record sharing using consortium blockchain, *IEEE Internet Things J.*, vol. 9, no. 14, pp. 12014–12028, 2022.

[4] X. Zhou, X. Liang, H. Zhang, and Y. Ma, Cross-platform identification of anonymous identical users in multiple social media networks, *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 2, pp. 411–424, 2016.

[5] L. Catarinucci, D. De Donno, L. Mainetti, L. Palano, L. Patrono, M. L. Stefanizzi, and L. Tarricone, An IoT-aware architecture for smart healthcare systems, *IEEE Internet Things J.*, vol. 2, no. 6, pp. 515–526, 2015.

[6] S. Yu, Big privacy: Challenges and opportunities of privacy study in the age of big data, *IEEE Access*, vol. 4, pp. 2751–2763, 2016.

[7] Y. Qu, S. Yu, W. Zhou, S. Peng, G. Wang, and K. Xiao, Privacy of things: Emerging challenges and opportunities in wireless internet of things, *IEEE Wirel. Commun.*, vol. 25, no. 6, pp. 91–97, 2018.

[8] D. He, R. Ye, S. Chan, M. Guizani, and Y. Xu, Privacy in the internet of things for smart healthcare, *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 38–44, 2018.

[9] X. Nie, A. Zhang, J. Chen, Y. Qu, and S. Yu, Blockchain-empowered secure and privacy-preserving health data sharing in edge-based IoMT, *Security and Communication Networks*, vol. 2022, p. 8293716, 2022.

[10] Y. Zhang, J. Tang, Z. Yang, J. Pei, and P. S. Yu, COSNET: Connecting heterogeneous social networks with local and global consistency, in *Proc. 21th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Sydney, Australia, 2015, pp. 1485–1494.

[11] J. Zhang, Q. Zhan, and P. S. Yu, Concurrent alignment of multiple anonymized social networks with generic stable matching, in *Theoretical Information Reuse and Integration*, T. Bouabana-Tebibel and S. H. Rubin, eds. Switzerland: Springer, 2016, pp. 173–196.

[12] M. M. Merener, Theoretical results on de-anonymization via linkage attacks, *Trans. Data Privacy*, vol. 5, no. 2, pp. 377–402, 2012.

[13] Y. Gong, C. Zhang, Y. Fang, and J. Sun, Protecting location privacy for task allocation in ad hoc mobile cloud computing, *IEEE Trans. Emerg. Topics Comput.*, vol. 6, no. 1, pp. 110–121, 2018.

[14] P. Samarati and L. Sweeney, Protecting privacy when disclosing information: K-anonymity and its enforcement through generalization and suppression, in *Proceedings of the IEEE Symposium on Research in Security and Privacy*, Oakland, CA, USA, 1998, pp. 1–19.

[15] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, L-diversity: Privacy beyond k-anonymity, *ACM Trans. Knowl. Discov. Data*, vol. 1, no. 1, pp. 3–55, 2007.

[16] N. Li, T. Li, and S. Venkatasubramanian, Closeness: A new privacy measure for data publishing, *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 7, pp. 943–956, 2010.

[17] C. Dwork, Differential privacy, in *Proc. 33rd Int. Conf. Automata, Languages and Programming – Volume Part II*, Venice, Italy, 2006, pp. 1–12.

[18] C. Dwork, Differential privacy, in *Encyclopedia of Cryptography and Security*, 2nd ed., H. C. A. van Tilborg and S. Jajodia, eds. New York, NY, USA: Springer, 2011, pp. 338–340.

[19] F. Koufogiannis and G. J. Pappas, Diffusing private data over networks, *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 1027–1037, 2018.

[20] S. Wang, J. Wang, X. Wang, T. Qiu, Y. Yuan, L. Ouyang, Y. Guo, and F. Wang, Blockchain-powered parallel healthcare systems based on the ACP approach, *IEEE Trans. Comput. Soc. Syst.*, vol. 5, no. 4, pp. 942–950, 2018.

[21] Y. Y. Ahn, J. P. Bagrow, and S. Lehmann, Link communities reveal multiscale complexity in networks, *Nature*, vol. 466, no. 7307, pp. 761–764, 2010.

[22] Y. Qu, L. Gao, Y. Xiang, S. Shen, and S. Yu, Fedtwin: Blockchain-enabled adaptive asynchronous federated learning for digital twin networks, *IEEE Netw.*, vol. 36, no. 6, pp. 183–190, 2022.

[23] W. Wang and Q. Zhang, Privacy preservation for context sensing on smartphone, *IEEE/ACM Trans. Netw.*, vol. 24, no. 6, pp. 3235–3247, 2016.

[24] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, Our data, ourselves: Privacy via distributed noise generation, in *Proc. 24th Annu. Int. Conf. Theory and Applications of Cryptographic Techniques*, St. Petersburg, Russia, 2006, pp. 486–503.

[25] S. R. Pokhrel, Y. Qu, and L. Gao, QoS-aware personalized privacy with multipath TCP for industrial IoT: Analysis and design, *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4849–4861, 2020.

[26] J. Ma, Y. Qiao, G. Hu, Y. Huang, M. Wang, A. K. Sangaiah, C. Zhang, and Y. Wang, Balancing user profile and social network structure for anchor link inferring across multiple online social networks, *IEEE Access*, vol. 5, pp. 12031–12040, 2017.

[27] S. R. Pokhrel, Y. Qu, S. Nepal, and S. Singh, Privacy-aware autonomous valet parking: Towards experience driven approach, *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5352–5363, 2021.

[28] D. Perito, C. Castelluccia, M. A. Kaafar, and P. Manils, How unique and traceable are usernames? in *Proc. 11th* Int. *Symp. Privacy Enhancing Technologies*, Waterloo, Canada, 2011, pp. 1–17.

[29] R. Zafarani and H. Liu, Connecting users across social media sites: A behavioral-modeling approach, in *Proc. 19th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Chicago, IL, USA, 2013, pp. 41–49.

[30] K. Xu, Y. Guo, L. Guo, Y. Fang, and X. Li, My privacy my decision: Control of photo sharing on online social networks, *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 2, pp. 199–210, 2017.

[31] R. Schlegel, C. Chow, Q. Huang, and D. S. Wong, Privacy-preserving location sharing services for social networks, *IEEE Trans. Serv. Comput.*, vol. 10, no. 5, pp. 811–825, 2017.

[32] J. A. Miller and B. Hoover, An exploratory analysis of the effects of spatial and temporal scale and transportation mode on anonymity in human mobility trajectories, in *Human Dynamics Research in Smart and Connected Communities*, S. L. Shaw and D. Sui, eds. New York, NY, USA: Springer, 2018, pp. 149–162.

[33] H. Li, Q. Chen, H. Zhu, D. Ma, H. Wen, and X. Shen, Privacy leakage via de-anonymization and aggregation in heterogeneous social networks, *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 2, pp. 350–362, 2017.

[34] Y. Yang, X. Zheng, W. Guo, X. Liu, and V. Chang, Privacy-preserving smart IoT-based healthcare big data storage and self-adaptive access control system, *Inf. Sci.*, vol. 479, pp. 567–592, 2019.

[35] Y. Zhang, D. Zheng, and R. H. Deng, Security and privacy in smart health: Efficient policy-hiding attribute based access control, *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2130–2145, 2018.

[36] H. Liu, X. Yao, T. Yang, and H. Ning, Cooperative privacy preservation for wearable devices in hybrid computing-based smart health, *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1352–1362, 2019.

[37] Y. Qu, M. P. Uddin, C. Gan, Y. Xiang, L. Gao, and J. Yearwood, Blockchain-enabled federated learning: A survey, *ACM Comput. Surv.*, vol. 55, no. 4, p. 70, 2022.

[38] J. Xu, K. Xue, S. Li, H. Tian, J. Hong, P. Hong, and N. Yu, Healthchain: A blockchain-based privacy preserving scheme for large-scale health data, *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8770–8781, 2019.

[39] A. D. Dwivedi, G. Srivastava, S. Dhar, and R. Singh, A decentralized privacy-preserving healthcare blockchain for IoT, *Sensors*, vol. 19, no. 2, p. 326, 2019.

[40] K. J. Peterson, R. Deeduvanu, P. Kanjamala, and K. Boles, A blockchain-based approach to health information exchange networks, in *Proc. NIST Workshop Blockchain Healthcare*, Miami, FL, USA, 2016, pp. 1–10.

[41] H. D. Zubaydi, Y. W. Chong, K. Ko, S. M. Hanshi, and S. Karuppayah, A review on the role of blockchain technology in the healthcare domain, *Electronics*, vol. 8, no. 6, p. 679, 2019.

[42] Y. Qu, X. Yuan, M. Ding, W. Ni, T. Rakotoarivelo, and D. Smith, Learn to unlearn: A survey on machine unlearning, arXiv preprint arXiv: 2305.07512, 2023.

[43] Doximity, Doximity developer API, https://www.doximity.com/, 2022.

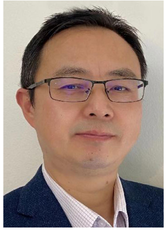
[44] HealthTap, Healthtap developer API, https://www.healthtap.com/, 2022.

Youyang Qu received the BEng degree in mechanical automation and the MEng degree in software engineering from Beijing Institute of Technology, China in 2012 and 2015, respectively, and the PhD degree from Deakin University, Australia in 2019. He is currently a research scientist at Data61, Commonwealth Scientific and Industrial Research Organization (CSIRO), Australia. Before joining CSIRO, he served as a research fellow at Deakin University. His research interests focus on machine learning, big data, IoT, blockchain, and corresponding security and customizable privacy issues.

Lichuan Ma received the BEng degree in information security from Shandong University, China in 2012, and the PhD degree in information security from Xidian University, China in 2018. Now, he is working at the School of Cyber Engineering, Xidian University, and a member of Shaanxi Key Laboratory of Blockchain and Secure Computing. His research interests focus on trust management and privacy-preserving techniques for intelligent systems.

Wenjie Ye received the BEng degree in computer engineering from Nanyang Technological University, Singapore in 2007, and the PhD degree in electrical engineering from Swinburne University of Technology, Australia in 2017. He is a lecturer in the information technology discipline at the College of Engineering and Science, Victoria University, Australia. His research interests are in artificial intelligence, blockchain, computer vision, fog computing, robotics, sliding mode control, system automation, vehicle systems, and vehicle control.

Xuemeng Zhai received the BEng and PhD degrees from University of Electronic Science and Technology of China (UESTC), China in 2014 and 2020, respectively. He is currently an assistant professor at the School of Information and Communication Engineering, UESTC. His research interests include network science, complex networks, knowledge graphs, and sparse representation.



Shui Yu received the PhD degree from Deakin University, Australia in 2004. He is currently a professor at the School of Computer Science, University of Technology Sydney, Australia. His research interests include network science, security and privacy, big data, and mathematical modelling.

Yunfeng Li received the PhD degree in computer simulation and chemical engineering from Monash University, Australia. He is currently the CEO of CNPIEC KEXIN LTD. He has won a scholarship from the University of New South Wales in Australia and a scholarship from Monash University in Australia. He is mainly engaged in the research of big data, artificial intelligence, computer simulation, and other directions, and leads the industrial application based on artificial intelligence technology.

David Smith received the BEng degree in electrical engineering from the University of New South Wales, Australia in 1997, and the MEng and PhD degrees in telecommunications engineering from the University of Technology Sydney, Australia in 2001 and 2005, respectively. He is currently a principal research scientist at Data61, CSIRO, leading the distributed privacy and security team in the information security and privacy group of the software and computational systems program. His research interests are in data privacy, distributed systems privacy and edge computing data privacy, distributed machine learning, data privacy for supply chains, wireless body area networks, game theory for distributed networks, 5G networks, disaster tolerant networks, and distributed optimization for smart grid.