J Inf Process Syst, Vol.20, No.4, pp.514~523, August 2024
https://doi.org/10.3745/JIPS.03.0200
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

# Collaborative Secure Decision Tree Training for Heart Disease Diagnosis in Internet of Medical Things

Gang Cheng[1], Hanlin Zhang[1,*], Jie Lin[2], Fanyu Kong[3], and Leyun Yu[4]

**Abstract**

In the Internet of Medical Things, due to the sensitivity of medical information, data typically need to be retained locally. The training model of heart disease data can predict patients' physical health status effectively, thereby providing reliable disease information. It is crucial to make full use of multiple data sources in the Internet of Medical Things applications to improve model accuracy. As network communication speeds and computational capabilities continue to evolve, parties are storing data locally, and using privacy protection technology to exchange data in the communication process to construct models is receiving increasing attention. This shift toward secure and efficient data collaboration is expected to revolutionize computer modeling in the healthcare field by ensuring accuracy and privacy in the analysis of critical medical information. In this paper, we train and test a multiparty decision tree model for the Internet of Medical Things on a heart disease dataset to address the challenges associated with developing a practical and usable model while ensuring the protection of heart disease data. Experimental results demonstrate that the accuracy of our privacy protection method is as high as 93.24%, representing a difference of only 0.3% compared with a conventional plaintext algorithm.

**Keywords**

Decision Tree, Heart Disease Diagnosis, Secure Multi-Party Computation

## 1. Introduction

Heart disease data help us understand the pathogenesis and influencing factors of heart disease, and they provide a basis for clinicians to perform effective diagnosis and treatment. By analyzing heart disease data, we can define the potential laws of the disease and predict the future health status of patients; thus, we can develop personalized treatment plans. In addition, heart disease data can be used to evaluate the effectiveness of preventive measures and provide scientific support for the formulation of public health policies. Thus, heart disease data play a crucial role in the development of heart disease prevention and treatment.

Applying machine learning algorithms in cardiac disease research [1,2] requires many high-quality datasets; however, it is frequently difficult for a single research institution or organization to satisfy this requirement. Thus, multiparty cooperation has become a key factor in solving this problem.



Manuscript received February 29, 2024; first revision May 8, 2024; second revision May 30, 2024; accepted June 7, 2024.

* **Corresponding Author:** Hanlin Zhang (hanlin@qdu.edu.cn)
[1] College of Computer Science & Technology, Qingdao University, Qingdao, China (sdgangcheng@qq.com, hanlin@qdu.edu.cn)
[2] School of Electronic and Information Engineering, Xian Jiaotong University, Xian, China (jielin@mail.xjtu.edu.cn)
[3] School of Software, Shandong University, Jinan, China (fanyukong@sdu.edu.cn)
[4] JIC IOT Co. Ltd., Jiangxi, China (yuleyun@jiciot.com)

However, in multiparty cooperation, we must maintain awareness of the sensitivity of heart disease data, which include patient privacy and health status. If such data are leaked or abused, the rights and interests of patients will be damaged. Thus, to ensure data privacy and security, advanced privacy protection technologies should be adopted [3,4].

The secure multiparty computing (MPC) method solves the problem of privacy-protecting collaborative computing among a group of untrusting parties. MPC ensures the independence of the input, the correctness of the calculation, decentralization, and other characteristics, while not disclosing the input values to other members involved in the calculation. There are many mature MPC frameworks, e.g., the SecureML, ABY3, SecureNN, and Falcon frameworks [5]. Thus, this paper uses secure MPC technology to address the problem of developing a multiparty heart disease diagnosis model.

### 1.1 Contributions

The primary contributions of this study are summarized as follows.

- We designed several MPC protocols to protect the raw heart disease data and established a decision tree (DT) model for heart disease diagnosis using data from various parties. Compared with traditional DT models trained for privacy protection of heart disease data, we have reduced the training complexity to a linear correlation with the depth of the tree.
- To avoid computing resource limitations, we propose an outsourcing scheme that allows participants to upload their heart disease data without having to participate in the online calculation.
- To prove the accuracy of the DT model for heart disease diagnosis, we implemented the protocol based on the MP-SPDZ framework (a secure program compiler capable of converting Python-based language programs to bytecode) and conducted benchmark experiments on the Cleveland database.

## 2. Problem Statement

### 2.1 Notations

In this paper, $[x]$ denotes the secret shared $x$, and the $i$-th component of a vector $v$ is denoted $v[i]$. The dataset at the root node, which comprises all data provided by all parties, is denoted $D$, and the $i$-th component of $D$ is denoted $D[i]$. The DT model produced through training is denoted $T$, where $T[i]$ represents the $i$-th node in the DT. The statistical security parameter is denoted $\lambda$.

### 2.2 System Model

Our goal is to develop a secure DT training protocol for joint cardiac data involving multiple parties. The proposed system architecture uses lightweight replicated secret sharing technology to send the parties' data to three trusted servers for secure DT training and saves the trained model between the three servers. Here, each participant, e.g., patient or a medical institution, can act as a data provider, with data from any of the parties. Cardiac data are highly private; thus, the data cannot be disclosed to others.

Note that our scheme is not restricted to the partitioning scenario of the cardiac dataset. In the vertical partitioning scenario, participants use a privacy-preserving set intersection to identify and align common samples. Once the dataset is determined, the participants can safely share their data, with each participant receiving a secret share of the complete dataset. Thus, the data provided by parties are transmitted to

three servers for secure DT training using replicated secret sharing, which ensures that the data are "available but not visible." Once the DT training is completed, the model makes new instance predictions in the form of secret sharing, and they can jointly restore the model for DT prediction.

### 2.3 Replicated Secret Sharing Scheme

Here, we present a brief overview of the replicated secret sharing protocol for three parties. This protocol is designed for arithmetic circuits modulo $2^n$. The protocol generates three random values $x_1, x_2, x_3$ such that $x = x_1 + x_2 + x_3$ for a secret value $x$. These shares are divided into three parts: $\{(x_1, x_2), (x_2, x_3), (x_3, x_1)\}$. Note that each participant owns a portion. For example, party 1 holds $(x_1, x_2)$. We outline the addition and multiplication operations as follows.

To add two secret values $x + y$, the parties involved do not need to communicate. Instead, they can compute $x + y := (x_1 + y_1, x_2 + y_2), (x_2 + y_2, x_3 + y_3), (x_3 + y_3, x_1 + y_1)$ locally.

In contrast, the parties must interact to multiply two secret values $x * y$. We define $z = x * y$, and the parties $P_1, P_2, P_3$ hold correlated randomness $\alpha,\ \beta,\ \gamma$ respectively, where $\alpha + \beta + \gamma = 0$. $P_1$ computes $z_1 = x_1y_1 + x_1y_2 + x_2y_1 + \alpha$ and sends $z_1$ to $P_2$ \$, $P_2$ computes $z_2 = x_2y_2 + x_2y_3 + x_3y_2 + \beta$ and sends $z_2$ to $P_3$, and $P_3$ computes $z_3 = x_3y_3 + x_3y_1 + x_1y_3 + \gamma$and sends $z_3$ to $P_1$.

### 2.4 Privacy-Preserving DT Training

As a core machine learning algorithm, DTs have been employed extensively for various classification and regression tasks, with representative examples including the ID3, C4.5, and CART algorithms. When training DTs, the input data are presented in tabular format, and the data feature two key values, i.e., the number of instances in the dataset $\boldsymbol{n}$ and the number of input attributes for each instance $\boldsymbol{m}$. In the DT inference process, the attributes of the input heart disease data are designated as heart disease attributes, and the attributes of the DT's inference results are designated as the classification.

Note that MPC cannot disclose the individual data assessments based on node information. To adapt to MPC, the child nodes must have the same data size as the parent node. The DT is constructed recursively from the root node downward; thus, in the MPC scheme, each node's data size is equivalent to that of the root node, thereby maintaining a consistent computational cost. As the tree depth increases linearly, the number of nodes and the computational cost increase exponentially, while plaintext algorithms exhibit linearly increasing computational cost. The innovative data structure reported by Hamada et al. [6] deviates from conventional algorithms and facilitates secure computation of the sum, prefix sum, and maximum value for each group by leveraging these modules to structure the training algorithm, and this design linearly scales the computational cost with the tree depth.

## 3. Building Blocks for DT Training

The data structure we utilize primarily comprises three protocols, i.e., the grouped prefix summation protocol, the grouped summation protocol, and the compressed grouping vector protocol.

### 3.1 Grouped Prefix Summation Protocol

The grouped prefix summation protocol computes the sum of each group in the heart disease attribute

vector in index order. Assuming the attribute vector $x = [7,4,0,8,4,5]$ and the auxiliary vector $a = [1,0,0,1,1,0]$ of the dataset, the protocol is calculated as $[7,11,11,8,4,9]$. To calculate the sum of $n$ tuples, if the last tuple has $a_n = 1$, the result will be $(x_n, 1)$. This means that $(x_1, a_1) + (x_2, a_2) + \cdots + (x_n, 1) = (x_n, 1)$. Note that the sum of previously calculated data can be reset when the first element of the grouping occurs. By utilizing this property, we can ensure that the data between different groups impact each other when performing the grouping prefix sum.

### 3.2 Grouped Summation Protocol

The design of the grouped summation protocol is based on the grouped prefix summation protocol, with the goal of computing the sum of the data in each group. For example, given the attribute vector $x = [5,7,2,1,3,4]$ and auxiliary vector $a = [1,0,0,1,1,0]$ from the dataset, the result of applying the grouping protocol is $sx = [14,14,14,1,7,7]$. Note that the grouping prefix summation value of the last element in each group corresponds to the grouping summation result.

### 3.3 Compressed Grouping Vector Protocol

The compressed grouping vector protocol condenses a grouping vector into a vector of a specified size. The elements of the newly generated vector are all assigned the last element of each group in the initial vector. Initially, we generate an indicator vector $[la]$ to pinpoint the last position of each group in the grouping vector and set the other positions in vector $x$ to 0. Then, the last element of each group is sorted in ascending order based on their group indices, and we extract the last $t$ elements of the sorted vector to form a new vector $[cx]$.

## 4. Heart Disease Diagnosis DT

The Gini index is frequently employed for classification tasks; thus, it is an excellent fit for our target application, i.e., diagnosing heart disease. Thus, we utilize the Gini index as the evaluation function in the DT for heart disease diagnosis. The Gini value, which serves as a measure of the purity of the heart disease dataset $D$, is calculated as follows:

$$Gini(D) = 1 - \frac{\left|D_{y_i=0}\right|^2 + \left|D_{y_i=1}\right|^2}{|D^2|}, \tag{1}$$

For dataset $D$ and a heart disease attribute $x$, the Gini index is defined as follows:

$$Gini_{index} = \frac{\left|D_{x_i \geq t}\right|}{|D|} Gini\left(D_{x_i \geq t}\right) + \frac{\left|D_{x_i<t}\right|}{|D|} Gini\left(D_{x_i<t}\right), \tag{2}$$

Abspoel et al. [7] simplified the calculation of the Gini index, as shown in Eq. (3), where the maximum value of the modified Gini index is equivalent to the minimum value of Eq. (2):

$$MGini_{index} = \frac{\sum_{b=0}^{1}\left|D_{x_i \geq t, y_i=b}\right|^2}{\left|D_{x_i \geq t}\right|} + \frac{\sum_{b=0}^{1}\left|D_{x_i<t, y_i=b}\right|^2}{\left|D_{x_i<t}\right|}. \tag{3}$$

First, we construct a subprotocol using the modified Gini index as the foundation. The purpose of this protocol is to determine the optimal attribute threshold for each heart disease attribute. Initially, we employ secure sorting to arrange the classification attributes and auxiliary vectors based on the heart disease attribute. Then, the classification attributes and heart disease attributes are sorted using the previously sorted auxiliary vector as the guiding factor. After two sorting rounds, the heart disease attribute assumes its place in the group. We calculate the modified Gini index using the median value of the adjacent heart disease attribute values within the group as the attribute threshold. Finally, we compare the optimal threshold of the maximum Gini index for the heart disease attributes.

For the heart disease dataset, the Gini index is calculated for each attribute, the attribute with the highest Gini index is identified, and its corresponding attribute threshold is recorded.

We present a batch internal node training protocol that is used to train nodes in the $k$-th layer (where $k$ belongs to the range $[0, h]$). This protocol receives private packet datasets from the $k$-th layer and transforms them into information for the output layer. In addition, the protocol computes the test results for all data within the nodes.

In the following, we discuss the specifics of the internal node training protocol. Initially, the protocol computes the heart disease attribute selection protocol to acquire the optimal heart disease attribute and attribute threshold for each group. Then, we assess whether the amount of data and the sum of the classifications in each group are equivalent. This is equivalent to determining whether the heart disease data within a node belongs to the same class. If they belong to the same class, the attribute threshold is set to the global minimum. Ultimately, the protocol acquires $[A]$, $[T]$, $[N]$, which are redundant because they are stored in an element-by-element manner. Thus, we employ the compressed grouping vector protocol to convert them into node-by-node storage formats $[AID^k]$, $[Threshold^k]$, and $[NID^k]$. Finally, to train the next layer of nodes, we analyze the data partitioning within each node based on the selected heart disease attribute.

Here, we present a batch processing protocol for training leaf nodes, which is utilized to train the layer h nodes. Through our group-by-group operation, this protocol calculates the most frequent values in each group's heart disease classification and directly labels them as leaf labels.

This protocol employs the group summation protocol to determine the maximum number of types of heart disease and health data in each group as the predicted labels for the leaf nodes. It also outputs $[NID^h]$ and $[Lable^h]$.

Finally, we present our DT training algorithm, which utilizes the developed protocols as building blocks. We construct a DT by conducting batch training for nodes of the same height layer by layer, which ensures that both the input and output remain confidential.

The protocol accepts two inputs, i.e., the publicly agreed tree depth $h$ and the heart disease dataset shared confidentially among all parties after secret sharing. The output is a DT that includes the index vectors for $h$ internal nodes, the heart disease attribute index vectors, and the attribute threshold vectors, as well as the index vectors for the leaf nodes and two vectors to predict the heart disease classification of the leaf nodes.

During training, we begin by initializing the auxiliary vector $a_0$ and the index vector $N^{(0)}$ for the first layer of nodes. Note that the first layer only has one node, i.e., the root node; thus, all data belong to the same group. Therefore, we set $a_0[1] = 1$ and set the remaining elements to $0$, while initializing the index vector to $1$.

The protocol trains the DT in a layer-by-layer manner, beginning from the root node (layer 0) and terminating at the leaf node (layer $h$). During each iteration, the $k$-th layer employs an internal node training protocol to generate a new layer of internal nodes and derive the partitioning results for the subsequent layer of data. Then, stable sorting of the current data is performed based on the obtained partitioning results. Due to the stability of this sorting process, elements that were originally in the same group and obtained identical test outcomes will appear sequentially after sorting, which indicates that they remain in the same group within our data structure. Finally, a leaf node training protocol is utilized to generate a layer of leaf nodes, thereby completing the DT training process.

## 5. Dataset

We conducted a set of benchmark experiments on the Cleveland heart disease database, which comprises 76 attributes; however, all published experiments have only utilized a subset of 14 attributes. Specifically, the Cleveland database is the sole database utilized by machine learning researchers to date. Table 1 provides details of the attribute content of the heart disease dataset.

Here, the "goal" field represents the presence of heart disease in a patient. It is assigned an integer value ranging from 0 (indicating absence) to 4. Researchers conducting experiments with the Cleveland database have primarily attempted to distinguish between the presence (values 1, 2, 3, 4) and absence (value 0) of heart disease. In our experiments, we assigned all values 1, 2, 3, and 4 to value 1 to determine whether the patient has heart disease, thereby disregarding the severity of the condition.

**Table 1.** Properties of heart disease dataset

| Variable name | Type | Descriptions |
|---|---|---|
| age | age | Record the personnel's age, ranging from 0 to 100 years |
| sex | Categorical | Binary gender represented by 0 and 1, respectively |
| cp | Categorical | Various forms of chest discomfort |
| trestbps | Integer | Resting blood pressure (on admission to the hospital) |
| chol | Integer | Serum cholesterol |
| fbs | Categorical | Fasting blood sugar >120 mg/dL |
| restecg | Categorical | Resting electrocardiogram results |
| thalach | Integer | Maximum heart rate achieved |
| exang | Categorical | Exercise induced angina |
| oldpeak | Integer | ST depression induced by exercise relative to rest |
| slope | Categorical | Slope of the highest motion ST segment |
| ca | Integer | Number of major vessels (0–3) colored by fluoroscopy |
| thal | Categorical | Thalassemia |
| target | Integer | Presence of heart disease |

## 6. Experimental Setup

In this study, we leveraged the MP-SPDZ framework [8] to implement our secure training protocol for DTs. The MP-SPDZ framework offers a compiler that converts secure programs written in a Python-based language into bytecode. Then, the bytecode can be executed by various C++-based engines, each

implementing the generic MPC protocol. The MP-SPDZ framework is designed for efficient and scalable secure computation in multiparty settings, and it provides a high-level programming language to express secure computations and optimizes the computation automatically to minimize communication and computation costs. In addition, it supports various types of secure computation, including arithmetic and Boolean circuits, and it allows for flexible deployment on various hardware platforms, including CPUs, GPUs, and FPGAs. By leveraging the MP-SPDZ framework, we can implement our secure training protocol for DTs efficiently and effectively, thereby making it suitable for real-world applications.

We conducted our benchmark experiments on a standard desktop computer with an Intel Core i7-11700K CPU (3.60 GHz) running Ubuntu 20.04 via VMware Workstation with 50 GB of memory. In addition, we utilized a script to execute all parties in non-interactive mode. This script, provided by the MP-SPDZ framework, simulates the runtime and communication costs of all parties accurately. The difference in protocol runtime between using this script and three computers connected via LAN (1 Gbps with 0.1 ms latency) was negligible. In this experiment, we set the statistical security parameter to 40 bits.

## 7. Performance Evaluation

To ensure the security of our approach, we employed the replicated secret sharing technique within the ring $Z_{2^{64}}$ because, in a three-party setting, replicated secret sharing offers a superior solution for semi-honest and honest-majority scenarios compared to other secret sharing methods. In addition, we evaluated the runtime and communication costs of each protocol under passive security conditions. The passive security protocol ensures that no party can access any information about the input data or the computations performed by the other parties, except for what can be inferred from the output of the computation.

Table 2 shows a comprehensive overview of the performance of our privacy-preserving heart disease diagnosis DT model compared with the plaintext algorithm model. For this evaluation, we utilized 75% of the heart disease dataset for training, and the remaining 25% was used for testing. We evaluated the accuracy of both sets across various tree depth settings. The protocol training process required only 17.779 seconds, and the communication cost for each central server was 825.143 MB, for a total of 2475.36 MB across all three servers. On a DT with a depth of five, the prediction accuracy on the training set reached 93.24%, and at a depth of three, the prediction accuracy on the test set reached 81.33%.

**Table 2.** Accuracy of training and testing sets with different tree depths

| Tree depth | Accuracy (%) | | |
|---|---|---|---|
| | Training set | Testing set | Training set on sklearn |
| 1 | 78.37 | 66.67 | 79.52 |
| 2 | 79.28 | 70.67 | 82.28 |
| 3 | 85.14 | 81.33 | 87.20 |
| 4 | 89.19 | 74.67 | 90.38 |
| 5 | 93.24 | 80.00 | 93.54 |

Fig. 1 shows the trend in accuracy changes obtained on the heart disease training and testing sets. DTs have unique characteristics; thus, the accuracy of the training set increases gradually as the tree depth increases. However, deeper trees frequently carry the risk of overfitting, which leads to reduced accuracy

on the test set. Thus, it is imperative to adjust the tree depth continuously in practical applications to achieve optimal results on the test set. Fortunately, the tree depth does not serve as a privacy protection value and does not pose a privacy threat to heart disease data.

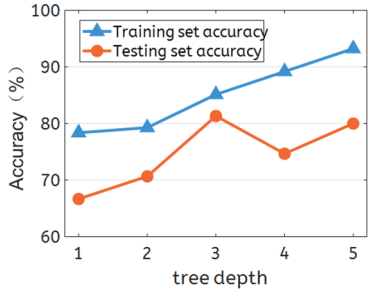


**Fig. 1.** Trend of accuracy variation with tree depth.

## 8. Conclusion

This paper has proposed a refined heart disease diagnosis DT model that safeguards privacy by utilizing lightweight replication secret sharing technology. This innovative method preserves sensitive patient data and utilizes a special data structure combined with MPC technology. Compared with traditional privacy protection algorithms with exponential complexity, the complexity of the proposed heart disease diagnosis DT algorithm is linearly proportional to the depth of the tree. This system enables participants to remain offline after uploading heart disease data, thereby offloading the computational burden to powerful servers. Currently, our system can only be applied to categorical datasets, and research on regression tasks will be the focus of future work.

## Conflict of Interest

The authors declare that they have no competing interests.

## Funding

This research is supported by National Natural Science Foundation of China (No. 62102212), Shandong Province Youth Innovation and Technology Program Innovation Team (No. 2022KJ296), Natural Science Foundation of Shandong (No. ZR202102190210) and Nanchang Major Science and Technology Project (No. 2023137).

## References

[1] S. S. Alotaibi, H. A. Mengash, S. Dhahbi, S. Alazwari, R. Marzouk, M. A. Alkhonaini, A. Mohamed, and A. M. Hilal, “Quantum-enhanced machine learning algorithms for heart disease prediction,” *Human-centric Computing and Information Sciences*, vol. 13, article no. 41, 2023. https://doi.org/10.22967/HCIS.2023.13.041

[2] D. Maheshwari, U. Ullah, P. A. O. Marulanda, A. G. O. Jurado, I. D. Gonzalez, J. M. O. Merodio, and B. Garcia-Zapirain, "Quantum machine learning applied to electronic healthcare records for ischemic heart disease classification," *Human-centric Computing and Information Sciences*, vol. 13, article no. 6, 2023. https://doi.org/10.22967/HCIS.2023.13.006

[3] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, D. Papadopoulos, and Q. Yang, "SecureBoost: a lossless federated learning framework," *IEEE Intelligent Systems*, vol. 36, no. 6, pp. 87-98, 2021. https://doi.org/10.1109/MIS.2021.3082561

[4] W. J. Lu, Z. Huang, Q. Zhang, Y. Wang, and C. Hong, "Squirrel: a scalable secure two-party computation framework for training gradient boosting decision tree," in *Proceedings of the 32nd USENIX Security Symposium (USENIX Security 23)*, Anaheim, CA, USA, 2023, pp. 6435-6451.

[5] S. Wagh, S. Tople, F. Benhamouda, E. Kushilevitz, P. Mittal, and T. Rabin, "Falcon: honest-majority maliciously secure framework for private deep learning," *Proceedings on Privacy Enhancing Technologies*, vol. 2021, no. 1, pp. 188-208, 2021. https://doi.org/10.2478/popets-2021-0011

[6] K. Hamada, D. Ikarashi, R. Kikuchi, and K. Chida, "Efficient decision tree training with new data structure for secure multi-party computation," *Proceedings on Privacy Enhancing Technologies*, vol. 2023, no. 1, pp. 343-364, 2023. https://doi.org/10.56553/popets-2023-0021

[7] M. Abspoel, D. Escudero, and N. Volgushev, "Secure training of decision trees with continuous attributes," *Proceedings on Privacy Enhancing Technologies*, vol. 2021, no. 1, pp. 167-187, 2021. https://doi.org/10.2478/popets-2021-0010

[8] M. Keller, "MP-SPDZ: a versatile framework for multi-party computation," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, Virtual Event, USA, 2020, pp. 1575-1590. https://doi.org/10.1145/3372297.3417872

**Gang Cheng** **https://orcid.org/0009-0001-8894-8204**

He received a B.S. degree in Information Security from Qingdao University in 2021. He is pursuing a master's degree in cyberspace security at Qingdao University. His research interests include secure multiparty computation and machine learning technologies.

**Hanlin Zhang** **https://orcid.org/0000-0001-8869-6863**

He received a B.S. degree in Software Engineering from Qingdao University in 2010. He received a M.S. degree in Applied Information Technology and a Ph.D. degree in Information Technology from Towson University, MD, USA, in 2011 and 2016, respectively. He is currently working at Qingdao University as an Associate Professor in the School of Computer Science and Technology. His research interests include privacy-enhanced computation, blockchain, and machine learning security.

**Jie Lin** **https://orcid.org/0000-0003-3476-110X**

He received B.S. and Ph.D. degrees from the Department of Computer Science and Technology at Xi'an Jiaotong University in 2009 and 2013, respectively. He is currently an associate professor in the Department of Computer Science and Technology at Xi'an Jiaotong University. His research interests include the Internet of Things, cyberspace security, and edge computing.

**Fanyu Kong** **https://orcid.org/0000-0003-1369-6855**

He received M.S. and B.S. degrees from the School of Computer Science and Technology at Shandong University, China, in 2003 and 2000, respectively. He received a Ph.D. degree from the Institute of Network Security at Shandong University, China, in 2006. He is currently an associate professor at the Institute of Network Security at Shandong University, China. His research interests include cryptanalysis, digital signature, and network security.

**Leyun Yu** **https://orcid.org/0009-0008-5296-7666**

He graduated in 2008 with a bachelor's degree in Information Management and Information Systems from Nanchang University. He completed his graduate studies while working and obtained a master's degree in Electrical Engineering from Nanchang University in 2017. He currently serves as the technical director at JIC IOT Co. Ltd., where he is responsible for the daily development and management of the company's research and development center.