Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 7463722, 16 pages
https://doi.org/10.1155/2023/7463722

Research Article

A Trusted Remote Data Trading Scheme in Hybrid SDN for Intelligent Internet of Things

Yu Zhang,[1,2] **Bei Gong**,[1] **Yong Wu**,[1] **Guiping Zheng**,[1] **and Zipeng Diao**[1]

[1]*Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China*
[2]*School of Information Science and Technology, Zhengzhou Normal University, Henan 450044, China*

Correspondence should be addressed to Bei Gong; gongbei@bjut.edu.cn

Received 3 August 2022; Revised 7 September 2022; Accepted 24 November 2022; Published 9 February 2023

Academic Editor: Zhuojun Duan

Intelligent internet of things (IIoTs) have these features: heterogeneous network patterns, significant differences in devices, dynamic variability of network topologies, etc. In the complex security situation, it is necessary to reject data from untrusted devices to guarantee the security data trading of IIoTs. In this paper, we focus on the trustworthiness and authentication in a hybrid SDN scenario of IIoTs. For the process of the trusted judgment, we firstly implement the standardized model for the nodes with the device attributes, network states, and operation behaviors. Based on the standardized model, we propose feature evaluation functions in SDN and IP domains, respectively, to calculate the intradomain node trust values to achieve the trusted judgment. To consider the demand for secure data trading for cross-domain devices, we propose a remote data trading scheme in which the data transmitter signs the node identity and its trust value by a group signature and the data receiver verifies the signature. The group signature is not only to protect the privacy of the group members but also to support the dynamic accession and revocation of group members, so it is more suitable for IIoTs where the nodes frequently access/exit. The security is proved under the standard model. We conduct the simulation experiments to evaluate the correctness of the trusted judgment mechanism. The evaluation shows that the scheme has lower computational cost and the higher efficiency of the group signature scheme.

1. Introduction

According to IoT analytics, the number of connected IoT devices worldwide is expected to reach 14.5 billion by the end of 2022. Up to 2025, there will probably be more than 27 billion IoT connections [1]. As the explosive growth of network scale and business traffic, the drawbacks of the traditional switch-based network architecture are obvious, and the increasingly complex network protocols have made network operation and maintenance more complicated and more challenging in management and configuration. The emergence of software-defined networking (SDN) has changed the existing network infrastructure, shielding the underlying physical network differences and meeting the needs of IIoT security, management, applications and other requirements, thus becoming an innovative network architecture suitable for IIoT heterogeneous scenarios.

Nowadays, it is very costly to complete the networkwide deployment of the SDN in a short period, so there will be the coexistence of the SDN network and the current mainstream IP network. Accordingly, the hybrid SDN model indicates the coexisting network architecture [2]. The hybrid SDN model inherits SDN security flaws such as topology poisoning attacks [3], new-flow attacks [4], blackhole attacks, grayhole attacks, Sybil attacks, and other security threats. In addition to internal and external attacks, devices in the hybrid SDN require the interaction of secure data trading. Therefore, we provide solid solutions for hybrid SDN to provide trusted judgment for intradomain nodes and remote attestation for cross-domain nodes, to ensure the security of the dynamic operating environment of IIoTs and the security of the data trading source.

The legacy IP and SDN networks interconnect, and the data trading demands of intradomain or cross-domain

devices exist in both network architectures. The data trading with untrusted devices is unsecured, so it is necessary to perform trust judgment for the data trading source, as illustrated in Figure 1. Because of the heterogeneity of device nodes, the trusted judgment methods for devices are hard to unify [5]. The SDN control node as the manager of the SDN domain builds the trust chain for the trust measurement of the devices, as seen in Figure 1(a). However, there are no management nodes in the IP domain, and the nodes make trusted judgments by each other, as seen in Figure 1(b). Due to the SDN and IP domains cannot communicate directly, cross-domain devices must exchange their identity information and trust values to judge whether the other party is trusted or not, as seen in Figure 1(c). The trusted judgment of cross-domain devices exploits the remote attestation mechanism proposed by TCG [6]. In hybrid SDN, we adopt the remote attestation mechanism to prove the trusted transmitter by sending relevant information such as node identity information and trust value to the remote node for verification. It is necessary to propose a signature scheme to prevent falsifying the message.

In hybrid SDN, IP and SDN networks can be treated as two separate groups, so the remote attestation scheme is suited to exploit the group signature. The group signature scheme can only indicate that the signer is from the group, which does not expose the node privacy and trace the signer when it is questioned. Since IIoT is a dynamic changeable network, the node trust value is closely related to the changes in the surrounding environment, such as the business execution and the states of the neighboring nodes. If the node is untrusted, it will not be able to participate in the signature process as a group member. Nevertheless, most group signature models do not consider the member revocation operation [7]. As a result, the group signature scheme which supports dynamic joining and revocation of group members is more suitable for the dynamic operating environment of IIoT nodes.

To ensure the security of data trading, it is necessary to confirm that the data trading source node is trusted. In this paper, we build the standardized description models of the intradomain nodes at first for the SDN and IP domains in the hybrid SDN architecture. Based on the standardized model, we calculate the trust value of the intradomain nodes by their respective feature evaluation functions. The cross-domain devices realize the remote attestation by a group signature to judge whether the data trading source node is trusted or not to enhance the security of IIoT data trading. Our paper makes the following contributions:

(i) By analyzing the operating environment of devices, tasks, and devices in hybrid SDN architecture, we build the standardized models using multidimensional attributes, network states, and interaction behaviors of the IIoT nodes in the SDN and IP domains

(ii) By setting the feature evaluation function, we propose the trusted judgment models of the intradomain nodes, complete the calculation of the node trust values, and verify the correctness of the trusted judgment process in simulation experiments

(iii) By using the remote attestation mechanism for cross-domain nodes, we propose a group signature scheme that supports dynamic joining and revocation of group members and has less computation and higher efficiency; meanwhile, the security of the scheme is proved under the standardized model

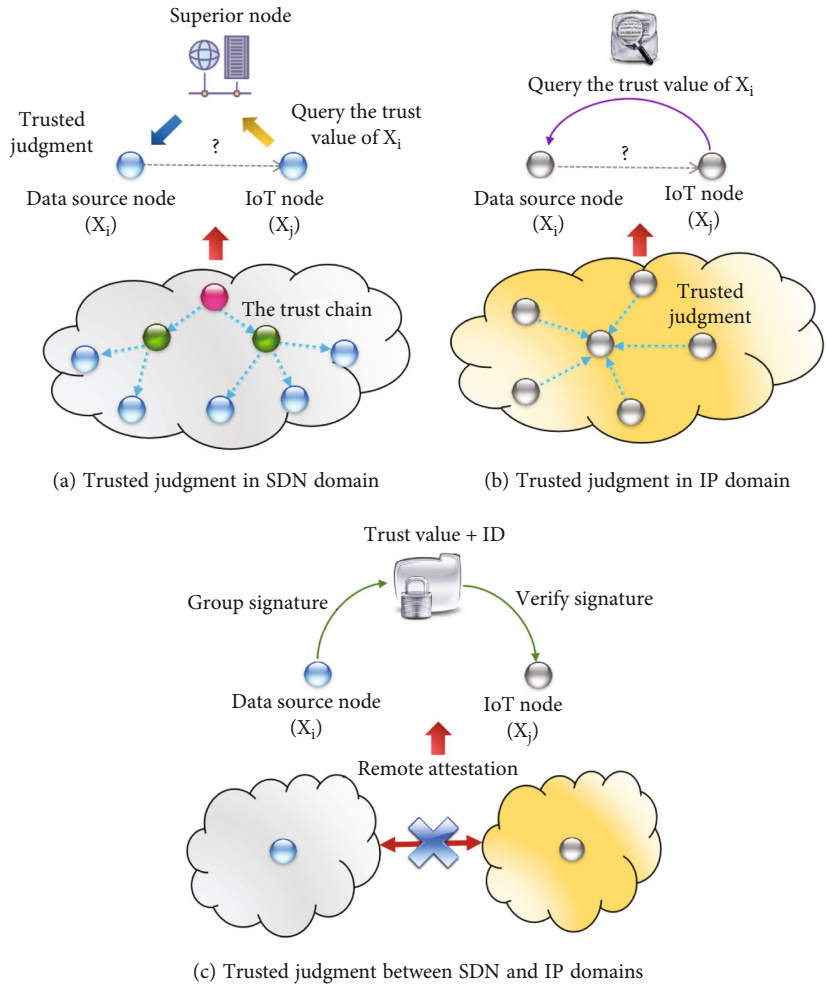
In Section 2, we describe the related work. Section 3 describes the hybrid SDN architecture and the standardized models in SDN domain and IP domain. We detail the implementation of the intradomain trusted judgment in Section 4. Section 5 realizes a group signature scheme and the security analysis. Section 6 is the simulation experiments for the trusted judgment and the group signature scheme. We conclude our work in Section 7.

2. Related Work

SDN is a novel network model; although SDN can be applied for IIoTs and dynamically perform different IIoT tasks in heterogeneous network scenarios, the openness of SDN leads to the possibility of serious security threats [8]. Liu et al. [9] proposed an SDN-based secure connectivity model for IIoTs, to safeguard the network by controlling the data flow and using a combination of channel and tag protocols to solve the routing security problem. But the security mechanism of the model is too complex, with poor real-time response and high energy consumption. Zhou et al. [10] combined trusted computing technology with the SDN network architecture to ensure the security and trust of the control domain in the SDN by using the SDN controller as the trusted root and measuring the device hardware, boot sequence, controller operating system, communication module, controller policy application, and other modules and network devices as the trusted chain transfer rules. The model relies on a security-trusted hardware platform and requires a costly reconstruction of the SDN controller.

The abstract description of devices in IIoTs is the key to building a secure and trusted IIoTs. There are several standardized description models for IIoT scenarios. Chen et al. [11] proposed multidimensional attributes of the IIoT nodes in edge computing, and a comprehensive trust aggregation algorithm is implemented by the subject node for the trusted judgment using the unified quantification. The model realizes trust attributes in the edge computing environment as a domain and lacks the security for cross-domain nodes. Zheng et al. [12] proposed a trust management mechanism for the wireless sensor network by calculating the trust values to achieve dynamic adaptive adjustment. The model uses a distributed networking structure to realize local trust measurement and global trust measurement for the selection and update of management nodes. The model is applicable to wireless sensor network architectures and only considers the distributed IIoT scenario.

The trusted IIoT data trading is built on the basis of the trusted data source. Yu et al. [13] discussed three IIoT data



(a) Trusted judgment in SDN domain

(b) Trusted judgment in IP domain

(c) Trusted judgment between SDN and IP domains

Figure 1: The process of trusted judgment for the intradomain or cross-domain nodes.

source security models and defined the security of each model, discussed the security challenges faced by different applications, and proposed security strategies for different attacks. However, the network models are from specific security threats and are less capable of dealing with unknown security threats. Gong et al. [14] proposed a trusted authentication scheme for IIoT data sources based on a trusted hierarchy, which uses a threshold group signature scheme to achieve trusted authentication of data sources. The group signature scheme lacks a revocation mechanism and is not applicable to the dynamic trusted judgment process. Liu et al. [15] proposed a distributed IIoT systems for smart cities, which builds a subdomain network by blockchain real-time reviewing of transaction nodes and adopts ring signature to ensure the privacy and security of data signature.

In general, the existing trust models and remote attestation schemes are not suitable for hybrid SDN. Legacy IP and SDN are different in networking models, devices, and protocol configurations. Therefore, it is necessary to redefine the node attributes and operational behaviors for hybrid SDN and study the intradomain and cross-domain trusted judgment mechanism which is applicable to hybrid SDN.

3. The Standardized Models for Hybrid SDN

3.1. Hybrid SDN Architecture. SDN cannot connect directly with legacy IP network due to the difference in message exchange mode; the hybrid SDN architecture is to be formed. Hybrid SDN networks are divided into the SDN autonomous system (SAS) and IP autonomous system (IPAS). The SAS is a centralized framework by the SDN controller as the manager, while IPAS is a distributed framework, with SAS and IPAS bridged by a gateway, as shown in Figure 2. The IP and SDN is hybrid in topology and divided into two domains, which face compatibility problems and security challenges in both domains [16].

The SAS contains an SDN controller, OpenFlow switches, legacy routers, legacy switches, middleware, agent, and IIoT devices. In Figure 2, the yellow table is the flow table which indicates that the device is under SDN control, and the green table is the legacy routing table. There are several networking models for SDN and IP networks to coexist: (1) a networkwide SDN in which SDN controllers manage OpenFlow switches and IIoT nodes: the mode is simple to deploy as the SDN controller performs all the management roles in the network, but not all legacy devices support

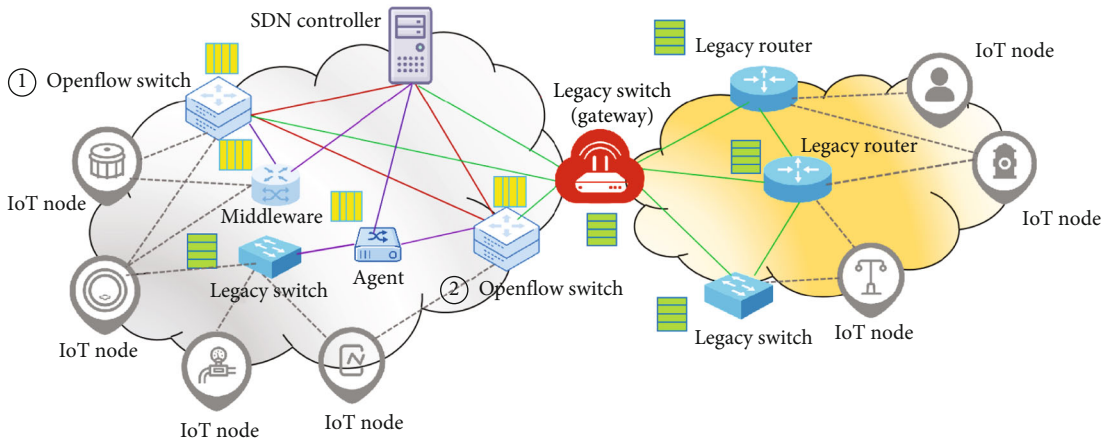


Figure 2: Hybrid SDN Architecture (left is the SAS domain, right is the IPAS domain, and the gateway connects two domains).

centralized management protocols. The SDN controller can execute the communication protocol by itself without protocol translation. A networkwide SDN mode is costly in the process of upgrading legacy network scenarios [17]. In terms of security, the problem of a single-point failure in the networkwide SDN controller is fatal, as illustrated by the red line in Figure 2. (2) Incremental deployment of SDN in legacy networks: the SDN controller uses the middleware to change the configuration of legacy devices by parsing the original IP legacy protocol to interact with the SDN switch in standard mode. The SDN switch does not need to support all OpenFlow features in the middleware mode. The middleware node is illustrated in the left diagram of Figure 2. (3) Legacy router/switch deployment in the SDN domain: an agent to implement the OpenFlow protocol is added to the legacy device to communicate with the controller without any changes applied to the controller. The bottleneck of the agent limits the scalability of the network, as illustrated by the agent node in the left diagram of Figure 2. A comparison of hybrid SDN networking models is shown in Table 1.

3.2. The SAS Intradomain Standardized Model. The SDN architecture defined by the ONF organization is divided into three layers [18], as illustrated in Figure 3. The SDN network infrastructure consists of switches, routers, middleware, and SDN network protocols. The SDN control plane executes the protocols and software, including the southbound protocol, network operating system, northbound open interface, and application layer software. OpenFlow is the most widely used southbound interface standard in SDN, connecting the controller and forwarding devices to achieve separation of the control plane and data plane. The northbound interface is an open interface to business applications that connects the control plane and application plane. The SDN controller (SDN network operating system) is a centralized scheduler of various resources in the network to provide services for traffic engineering, mobile and wireless networks, network measurement and monitoring, network security, and data center networks. The SAS intradomain standardized model is proposed as follows.

3.2.1. SAS Device Attribute Description Vector. The core functions of the SDN controller include forwarding device management, forwarding rule calculation, and resource management, which realize the centralized management of SDN. A single-node controller architecture is illustrated in the middle of Figure 3. SDN controller attributes are denoted by the 7-tuple $DC = (id, ds, im, cm, CS, CM, CA)$, where id indicates the controller identity (including network identification number, authentication key, and additional access information); ds indicates the digest value of the stored information component in the controller, which is used to store and manage all SDN information; and im is the digest value of the information processing component, which is used to configure various rules of the forwarding device, $im = (ft, gt, mt, pp)$. OpenFlow switches consist of ports and flow tables, group tables, and meter tables. The set of ports is denoted by pp. The flow table ft is the forwarding table digest value for data flow. A data flow corresponds with a flow table entry, and the mapping of the source and destination flow tables is configured by the specific forwarding device. The group table gt is defined as a set of action buckets that can be used by multiple flow table entries to achieve multicast, load balancing, disaster tolerant backup, and aggregation functions. The meter table mt provides QoS for OpenFlow switches by metering flows and setting speed limit rules. cm is the control management component, which represents the communication overhead of the controller. The component is responsible for connecting various forwarding devices of SDN networks and managing the flow table states. CS denotes the mapping relationship of the ith controller instance X_i and the w switches attached to it. If CS contains SDN middleware or agent in the hybrid SDN structure (see Figure 2), CM denotes the mapping relationship of X_i and the middleware which belongs to it, and CA denotes the mapping relationship of X_i and the agent which belongs to it.

The other nodes must register their device attributes information to the SDN controller when they are first accessing to network. The network nodes and IIoT nodes consist of device identification id, device type dt, device hardware information ho, basis software, and configuration protocol

Table 1: Comparison of hybrid SDN networking models.

Networking model	SDN components	Protocol conversion	Communication scope	Scalability & robustness
Networkwide SDN	Controller/switches/nodes	N/A	SDN internal	Impacted by SDN controller performance
SDN middleware	Controller/middleware/nodes	Middleware parsing protocol	At least one SDN device	Restricted by protocol parsing
SDN agent	Controller/legacy switches & agents/nodes	Agent parsing protocol	All nodes limited by agents	Limited by controller loads
Hybrid SDN	Controller/switches/gateways/legacy switches & routers/nodes	VLAN gateway	Networkwide	Dependent on network architecture

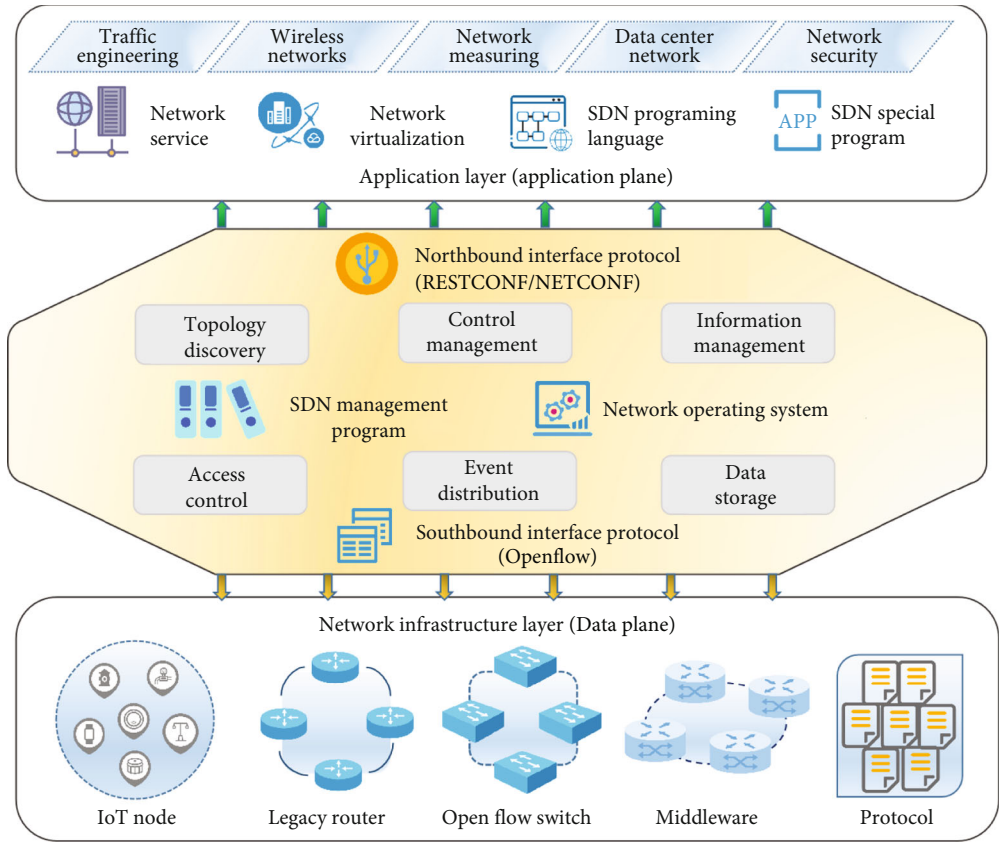


Figure 3: The SDN network architecture (the ONF organization defines three layers, including control plane, data plane, and application plane).

information hs, which are described by the 4-tuple WNN = (id, dt, ho, hs).

3.2.2. SAS Network Attribute Description Vector. In SDN networks, the bandwidth is bw, the trading delay is td, the flow table lifetime is tt, and the time delay is tl. Therefore, the following 4-tuple is defined to describe the network attributes NS = (bw, td, tt, tl), where the trading delay td represents the time gap from the entry of the packet into processing to the end of being processed, assuming that the beginning time is t_1 and the end time is t_2, then td = $t_2 - t_1$. The flow table lifetime tt indicates the flow table existence time, and the value tt impacts the switch forwarding speed; the larger means the longer occupied the switch time, the more load for the flow table to process. The time delay tl indicates the time that a packet is sent from the source to the destination and consists of the link trading delay, signal propagation time on the link, node queuing, and processing time, $\mathrm{tl} = \sum_{i=1}^{m}(p/\mathrm{bw}_i + \mathrm{td}_i) + \sum_{j=1}^{n}(t_{q_j} + t_{s_j})$, where p is the packet group size, bw_i is the bandwidth, td_i is the signal propagation delay, t_{q_j} is the node queuing time, and t_{s_j} is the node processing time.

3.2.3. SAS Link State Description Vector. The SDN link states include the node addition and deletion, link alteration, and the topology states. The SDN network topology is represented by a 3-tuple Topo = (N, P, L),where N denote the set of nodes, P denotes the set of ports, and L denotes the set of links in topology. $L(p_1, p_2)$ denotes the data flow from

port p_1 to port p_2, $p_1, p_2 \in P$. When p_1, p_2 belong to the same node, L indicates a data link from port p_1 in to port p_2 out in a node; when p_1, p_2 belong to different nodes, L indicates a data link through port p_1 to port p_2. Then, $\text{Topo}_i = (N_i, P_i, L_i)$ and $\text{Topo}_j = (N_j, P_j, L_j)$ denote the network topology states at the ith moment and the jth moment. $\Delta\text{Topo}_{i \longrightarrow j}$ denotes the topology changes, $\Delta\text{Topo}_j = (\Delta N_{i \longrightarrow j}, \Delta P_{i \longrightarrow j}, \Delta L_{i \longrightarrow j})$, where $\Delta N_{i \longrightarrow j}$ denotes the set of node changes from the ith to the jth moment, $\Delta N_{i \longrightarrow j} = \{\sum_{k \in 1,2 \cdots} \pm n_v{}^k\}$, where $n_v \in N$ means nodes, $\pm$ means add/delete n_v nodes by device management module in controller, if $\Delta N^{i \longrightarrow j} = NULL$ means no change in device nodes. $\Delta P_{i \longrightarrow j}$ denotes the set of port changes from the ith to the jth moment, $\Delta P_{i \longrightarrow j} = \{\sum \pm p(n_v)^u\}$, where $p(n_v)^u$ denotes the uth port of node n_v, $\pm$ denotes port enabled/disabled, and the port addition/deletion depends on the data forwarding rule $\Delta L_{i \longrightarrow j}$. $\Delta L_{i \longrightarrow j}$ denotes the changes of data links from the ith to the jth moment, $\Delta L_{i \longrightarrow j} = \{\sum \pm L(p^{u_1}(n_a), p^{u_2}(n_b))\}$, where $\pm L(p^{u_1}(n_a), p^{u_2}(n_b))$ indicates the link change from the port u_1 of node n_a into the port u_2 output of node n_b, $n_a, n_b \in N$, $p^{u_1}(n_a), p^{u_2}(n_b) \in P$. If there is no change in the device node, topology switching is performed based on the data of $\Delta L_{i \longrightarrow j}$ in $\Delta\text{Topo}_{i \longrightarrow j}$.

3.2.4. SAS Application Task Description Vector. Given that the set of k tenants in SAS is $O = \{o_1, o_2, \cdots, o_k\}$, n applications represented the set $A = \{a_1, a_2, \cdots, a_n\}$; each application runs m tasks represented by the set $S = \{s_1, s_2, \cdots, s_m\}$; the mapping relationship between controller and switches is CS; the $w \times n$ mapping relationship is constructed between w switches and n applications, represented by SA; the $k \times n$ mapping relationship is constructed between k tenants and n applications as TA; and the $n \times m$ mapping relationship is constructed between n applications and m tasks as OA; then, the SDN application tasks can be described by the 4-tuple $\text{AT} = (\text{CS}, \text{SA}, \text{TA}, \text{OA})$.

The SAS standardized model can be composed of device attributes description vector DC, network attributes description vector NS, link states description vector Topo, and application tasks description vector AT; thus, the SDN single-controller mapping model is represented by the 5-tuple $\text{NSC} = (\text{DC}, \text{WNN}, \text{NS}, \text{Topo}, \text{AT})$.

3.3. The IPAS Intradomain Standardized Model. There are some heterogeneous networks in IIoT scenarios, with the difference in hardware and software attributes, deployment locations, task states, computing capabilities, and network data trading capabilities of the nodes. Building a mapping model of IIoT nodes is critical to complete IIoT security and trust. A unified mapping model for IIoT devices in IPAS is shown as follows.

3.3.1. IIoT Device Raw Attribute Description Vector. The IIoT devices are identified by attributes solidified in devices, such as the device profile information about the device hardware type, vendor, product name, version number, and device verification information. The software attributes consist of firmware, communication protocols, third-party libraries, and operating system information. Firmware is a software module to accomplish the communication between various types of devices in order to overcome the problem of heterogeneous communication protocols. The device identification is id, the hardware device information digest value is denoted by hn, and the basic software digest value is denoted by hs; then, the 3-tuple $\text{NN} = (\text{id}, \text{hn}, \text{hs})$ is used as the identification information of the IIoT heterogeneous node entity.

3.3.2. IIoT Network Attribute Description Vector. A set of network addresses is present by the node source address, destination address, and MAC address as na, np is present as the set of ports, network bandwidth as nb, the selected channel as nc, the requested data as nq, the actual sending data as nd, the response time as nt, the IIoT heterogeneous network environment can be represented by the 7-tuple $\text{NS} = (\text{na}, \text{np}, \text{nb}, \text{nc}, \text{nq}, \text{nd}, \text{nt})$.

3.3.3. IIoT Link State Description Vector. The IP network routers are interconnected through links, usually indicated by the unweighted undirected connectivity diagram [17]. $\text{Topo} = (V, E)$ is present the state of links, where V denotes the set of routers, switches, and IIoT nodes and E denotes the set of links. There are two link instances Topo_i and Topo_j, and the change of nodes $\Delta V_{i \longrightarrow j}$ using the intersection of two-node sets is presented by $\Delta V_{i \longrightarrow j} = |V_i \bigcap V_j|$, and the change of links is presented by $\Delta E_{i \longrightarrow j} = |E_i \bigcap E_j|$.

3.3.4. IIoT Application Task Description Vector. IIoT nodes collect data, forward the data, and even provide services. The composite tasks execute in multiapplication multitasking scenarios. In Δt, $\text{NA} = \{\text{na}_1, \text{na}_2, \cdots, \text{na}_n\}$ denotes the set of all applications running at a node instance, and the subtasks set running at an application instance is presented by $\text{NT} = \{\text{nt}_1, \text{nt}_2, \cdots, \text{nt}_m\}$. Each subtask can only connect one channel in a time period, and the channel set is presented by $\text{NC} = \{\text{nc}_1, \text{nc}_2, \cdots, \text{nc}_k\}$. The application tasks of the node are presented as the 3-tuple $\text{NAT} = (\text{NA}, \text{NT}, \text{NC})$.

The IIoT devices in IPAS are interconnected using IP protocols. The IPAS node is mapped by building multidimensional attributes such as raw attributes NN, network attributes NS, link states Topo, and application tasks NAT. Thus, the IPAS node can be represented by the 4-tuple $\text{NIP} = (\text{NN}, \text{NS}, \text{Topo}, \text{NAT})$.

4. A Trusted Judgment Model for Hybrid SDN

4.1. The Definition of the Trusted Judgment Model. In the hybrid SDN scenario, the trusted judgment methods are different based on the networking models of SAS and IPAS. Because of the centralized networking in SAS, the controller as the manager evaluates other nodes according to the feature evaluation function to judge whether they are trusted. In IPAS, there is no management node; hence, the nodes are equal to each other; thus, the feature evaluation function needs to be developed by the collaboration of the nodes and then judge which nodes are trusted by the feature evaluation function. All operations of the abnormal nodes in domains

will be strictly restricted. The definition of trusted judgment is shown as follows.

Proposition 1 (SAS feature evaluation function). *Suppose there exists a group Gs, the mapping model of nodes in SAS is NSC, the SDN controller is the management node, the node rule calculation function is Gs is FV, and the trust value of node X_i is calculated by the feature evaluation function $M_{SAS}(\cdot)$, then the node trust value is calculate as $Trust_{Xi} = M_{SAS}(NSC, FV)$.*

Proposition 2 (IPAS feature evaluation function). *Suppose there exists a group Gs, the mapping of nodes in IPAS is NIP, the node rule calculation function is FV', and the trust value of node X_i is calculated by the feature evaluation function $M_{IP}(\cdot)$, then the node trust value is calculate as $Trust_{Xi} = M_{IP}(NIP, FV')$.*

Proposition 3 (Trusted judgment). *Set a trust value boundary threshold H_t at each autonomous system; if the node trust value $Trust_{Xi} < H_t$, the node is judged to be abnormal and all the operations of the node are restricted.*

4.2. The SAS Intradomain Trusted Judgment Model. The trusted judgment model is realized based on the standardized model in Section 3, and the following logically describes the process of trusted judgment for nodes in SAS and IPAS. Assuming that the SDN controller as the management node in SAS is trusted, it is necessary to perform the trusted judgment for the connected devices, as described in Section 3, the control node measures the network nodes such as router, switch, middleware and the common IIoT nodes to ensure the global trust value in the SAS domain, as demonstrated in Figure 4.

4.2.1. Node Device Metric. In SAS, the IIoT devices request to access the network for the first time or the global trust value is lower than the threshold that needs to evaluate, based on the device attributes description vector $\mathrm{WNN} = (\mathrm{dt}, \mathrm{id}, \mathrm{ho}, \mathrm{hf})$ of the nodes as input. The digest value of the node hardware and software is judged by the feature evaluation function $M_{\mathrm{SAS}}(\cdot)$ whether the IIoT node is trusted, and the IIoT node trusted evaluation is performed by the network node. In case x is the superior node and $(x.\mathrm{dt}, x.\mathrm{ho}, x.\mathrm{hs})$ denotes the device type, hardware, and software information submitted at the first registration period to x; the device trust value of the nodes $M_{\mathrm{SAS1}}(\mathrm{WNN}, \wedge)$ in SAS is calculated by

$$M_{\mathrm{SAS1}}(\mathrm{WNN}, \wedge) = (x.\mathrm{id} \wedge \mathrm{id})(x.\mathrm{dt} \wedge \mathrm{dt})(x.\mathrm{ho} \wedge \mathrm{ho})(x.\mathrm{hs} \wedge \mathrm{hs}). \tag{1}$$

IIoT node submits attributes such as device information when it is registered, and the superior node will compare the submitted information with the one registered to judge whether the node is a fake or not. And $M_{\mathrm{SAS1}}(\mathrm{WNN}, \wedge) = 1$ indicates that basic information such as node software and hardware is not forged.

4.2.2. Dynamic Behavior Metric. The node device metric is only a verification of the identity, and the dynamic behavior metric of the node is also necessary. The node network attributes $\mathrm{NS} = (\mathrm{bw}, \mathrm{td}, \mathrm{tt}, \mathrm{tl})$ reflect the network quality level. Assuming the network state expectation set by the SDN controller at time t is $E(NS)$, the cosine similarity is used to calculate the network state similarity function to decide the network environment similarity $f_{\mathrm{NS}} = E(\mathrm{NS}) \cdot \mathrm{NS}/|E(\mathrm{NS})||\mathrm{NS}|$. The data trading latency td must be in the tolerable range, with a threshold value of θ. If $\mathrm{td} > \theta$, the probability of the attacked node is increased and the data trading delay metric value is $T_{\mathrm{td}} = \begin{cases} \mathrm{td}^{(\mathrm{td}-\theta)/\theta}, & \mathrm{td} > \theta \\ \mathrm{td}, & \text{otherwise} \end{cases}$. The malicious node drops specific packets by probabilistic forwarding or by spoofing, tampering, or retransmitting routing information through routing loops. Therefore, the data forwarding amount and the repetition rate can be used to detect the abnormal behavior of the node. The data forwarding of the node can be obtained from the flow table for the requested data forwarding amount fq and the actual data forwarding amount fs and the data repetition rate φ; then, the actual forwarding ratio value of the node is $T_{\mathrm{fs}} = \mathrm{fs} \cdot \log_2(1 + (\mathrm{fs}/(\mathrm{fq} + 1)))$. The threshold value of data forwarding repetition rate is δ, if $\varphi > \delta$; it is possible to occur the blackhole attack and the repetition rate for forwarding behavior is $T_{\varphi} = \begin{cases} \mathrm{fq}(1 - \delta^{\varphi}), & \varphi < \delta \\ 0, & \text{otherwise} \end{cases}$.

In summary, the node dynamic metric can reflect the impact of the anomalous nodes. The node dynamic behavior trust value $M_{\mathrm{SAS2}}(\mathrm{NS}, f_{\mathrm{NS}})$ is defined by the network environment similarity, data trading delay metric, the actual forwarding ratio value, and the data repetition rate. Thus, $M_{\mathrm{SAS2}}(\mathrm{NS}, f_{\mathrm{NS}})$ is calculated by

$$M_{\mathrm{SAS2}}(\mathrm{NS}, f_{\mathrm{NS}}) = f_{\mathrm{NS}}\left(\varepsilon_{\mathrm{td}} \cdot T_{\mathrm{td}} + \varepsilon_{\mathrm{fs}} \cdot T_{\mathrm{fs}} + \varepsilon_{\varphi} \cdot T_{\varphi}\right), \tag{2}$$

where ε_{td}, ε_{fs}, and ε_{φ} denotes the weights, and $\varepsilon_{td} + \varepsilon_{fs} + \varepsilon_{\varphi} = 1$.

4.2.3. Task Execution Environment Metric. The node operational state is dynamic, and the node task execution is used to evaluate whether the node operation is as expected. The accuracy and dynamics of the node metric require checking the node behavior change in the time window Δt. And the node behavior change is denoted by $\mathrm{BN} = \{\Delta \mathrm{Topo}_{i \longrightarrow j}, \mathrm{AT}\}$, the expected behavior of the node is presented by $\overrightarrow{BN} = \{\Delta \mathrm{Topo}_{i \longrightarrow j}, \mathrm{AT}\}$, and the Jaccard similarity coefficient is used to calculate the execution environment trust value $M_{\mathrm{SAS3}}(\mathrm{BN}, \overrightarrow{\mathrm{BN}})$ by

$$M_{\mathrm{SAS3}}\left(\mathrm{BN}, \overrightarrow{\mathrm{BN}}\right) = \frac{\mathrm{Jaccard}\left(\mathrm{BN}, \overrightarrow{\mathrm{BN}}\right)}{1 + \mathrm{Jaccard}\left(\mathrm{BN}, \overrightarrow{\mathrm{BN}}\right)} \cdot \left(|\mathrm{BN}| + \left|\overrightarrow{\mathrm{BN}}\right|\right), \tag{3}$$

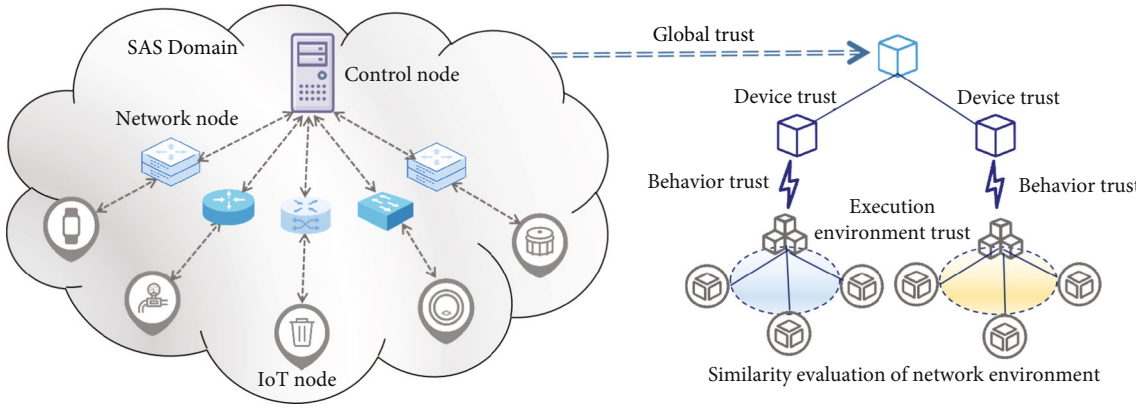


Figure 4: The trusted judgment model in SAS (including the trusted judgment of SDN controller, network nodes, and IIoT nodes).

where $\text{Jaccard}(\text{BN}, \overrightarrow{\text{BN}}) = |\text{BN} \bigcap \overrightarrow{\text{BN}}|/|\text{BN} \bigcup \overrightarrow{\text{BN}}|$ denotes the Jaccard similarity coefficient of the set BN with $\overrightarrow{\text{BN}}$. The larger the $M_{\text{SAS3}}(\text{BN}, \overrightarrow{\text{BN}})$ value, the more similar to the actual operational behavior of the node and the expected behavior, as well as the higher the dynamic behavior trust value. Jaccard similarity coefficient can be used to quickly estimate the similarity of two sets using the MinHash algorithm [19].

4.2.4. Global Trust Value of the Node. The node global trust value is aggregated by the node device trust value $M_{\text{SAS1}}(\text{WNN}, \wedge)$, the node dynamic behavior trust value $M_{\text{SAS2}}(\text{NS}, f_{\text{NS}})$, and the node task environment trust value $M_{\text{SAS3}}(\text{BN}, \overrightarrow{\text{BN}})$. If the node device trust value is 0, the node is faked. The global trust value of the node Trust_{X_i} is calculated by

$$\text{Trust}_{X_i} = M_{\text{SAS1}}(\text{WNN}, \wedge)\left(\alpha \cdot M_{\text{SAS2}}(\text{NS}, f_{\text{NS}}) + \beta \cdot M_{\text{SAS3}}\left(\text{BN}, \overrightarrow{\text{BN}}\right)\right), \tag{4}$$

where α and β denote the weights, $\alpha + \beta = 1$.

In SAS, the controller performs the security situation assessment of each node, including initial integrity verification and dynamic behavior verification. If the trust value of the node is higher than the given trusted threshold, the node is considered trusted; otherwise, the node is untrusted to restrict the node operation to ensure the SAS domain is trusted.

4.3. The IPAS Intradomain Trusted Judgment Model. In IPAS, the security situation assessment of the node in the distributed network is realized by its neighbor nodes. The node using store-and-forward messages can obtain the recommendation trust value by the neighbor nodes, as illustrated in Figure 5.

4.3.1. Direct Recommendation Trust Value. In IPAS, the node is connected to the network and exchanges identity information with neighbor nodes to be evaluated, moreover, let node X_j evaluate node X_i has a times normal interaction and b times abnormal interaction. The interaction result obeys the $\text{beta}(p|a, b)$ distribution, and p is the posterior probability of (a, b) [20]. According to the Bayesian trust model, the expectation is used as the trust value, the directly recommendation trust value $M_{\text{IP}_1}(a, b)$ can be calculated by

$$M_{IP_1}(a, b) = E(\text{beta}(p|a, b)) = \frac{a+1}{a+b+2}. \tag{5}$$

The Bayesian trust model only considers the interaction among nodes and does not consider the effect of the recommendation trust value on the current nodes, such as the decline of trust value with the increase of time. Assume in the time window Δt, for the node X_i, the sequence of n directly recommendation trust values is $\{M_1, M_2, \cdots, M_n\}$. Let M_1 be a trust value of node X_i with the longest time, and M_n is the trust value at the current time. The decay function exhibits that the node trust value decays according to the changes of the execution times n, so the decay function is defined as $H(i) = \begin{cases} 1 & i = n \\ H(i-1) = H(i) - (1/n), & 1 \le i \le n \end{cases}$, where $H(i) \in [0, 1]$.

A reward/punishment factor is used to evaluate the successful/failed interaction behavior of nodes, then the reward/punishment factor is set to $F(x) = N(x)/(N(a) + N(b))$, where $x = (a, b)$ indicates successful interaction or failed interaction behavior, $N(a)$ is the number of successful interactions, and $N(b)$ is the number of failed interactions. The reward value R_i of the current node is related to the times i that the node is judged to be trusted and the reward value R_{i-1} of the previous time. The node reward factor is $F(x)_R$ and the punishment factor is $F(x)_F$, so the node reward/punishment value can be calculated by $R_i(F(x)) = \begin{cases} R_{i-1} * (1 + F(x)_R^{i-1}) \\ R_{i-1} * (1 - F(x))_P^{i-1} \end{cases}$. The trust value is decayed with time while the reward/punishment factor needs to be updated. When a node completes an interaction, if the interaction behavior is a successful interaction, it is counted in the sequence $M(a_{ij})$, then $a_{ij} = \sum_{i=1}^{n} R_i \cdot H(i) \cdot M(a_{ij})$. If the interaction is failed, it is counted in the sequence $M(b_{ij})$, then $b_{ij} = \sum_{i=1}^{n} R_i \cdot H(i) \cdot M(b_{ij})$. a_{ij} and b_{ij} are substituted

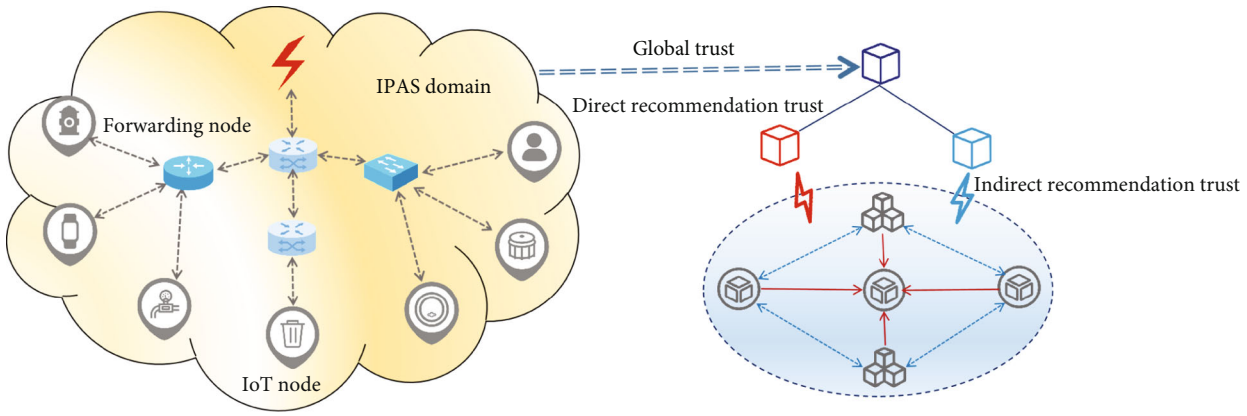


FIGURE 5: The distributed networking and trusted judgment model in IPAS (including direct recommendation trust and indirect recommendation trust).

into the Bayesian trust model (Equation (5)); then, the direct recommendation trust value M_{IP1} of the node is calculated in

$$M_{\mathrm{IP1}}(a_{ij}, b_{ij}) = \frac{a_{ij}+1}{a_{ij}+b_{ij}+2}. \tag{6}$$

4.3.2. Indirect Recommendation Trust Value. For the distributed networking model, the neighbor nodes need to be evaluated comprehensively, as shown by the blue line in Figure 4, where malicious nodes may raise or devalue the trust value of the evaluated node. Let the initial recommendation trust value sequence for the node X_i be $m_{r1}, m_{r2}, \cdots, m_{rk}$, and the direct recommendation trust mathematical expectation is $E(m_r) = 1/k\sum_{i=1}^{k} m_{ri}$, and calculate the recommendation trust and its mathematical expectation trust similarity as the deviation of the trust data; the further away from the expected value the smaller the weight, the more likely it is to be malicious and defamatory. The Euclidean distance similarity discrimination is used to calculate the trust data evaluation dispersion as $f_{m_{ri}} = |E(m_r) - m_{ri}|/\sqrt{\sum_{i=1}^{k}(E(m_r) - m_{ri})^2}$. According to the indirect recommendation trust value and its weight, the indirect recommendation trust value is M_{IP2}, as shown by

$$M_{\mathrm{IP2}}\left(m_{ri}, f_{m_{ri}}\right) = \sum_{i=1}^{k}\left(1 - f_{m_{ri}}\right)\cdot m_{ri}. \tag{7}$$

4.3.3. Global Trust Value of the Node. With the distributed networking model of IPAS, the global trust value of the evaluated node X_i at the moment t merges the direct recommendation trust value and the indirect recommendation trust value, and the global trust value Trust_{X_i} is calculated by

$$\mathrm{Trust}_{X_i} = \alpha \cdot M_{\mathrm{IP1}}(a_{ij}, b_{ij}) \cdot + \beta \cdot M_{\mathrm{IP2}}\left(X_i, f_{m_{ri}}\right)\Big), \tag{8}$$

where α and β are the adaptive weight of direct recommendation trust and indirect recommendation trust, and $\alpha + \beta = 1$. The information entropy $H(M_{\mathrm{IP1}}), H(M_{\mathrm{IP2}})$ is used to determine the weights corresponding to each indicator to overcome the limitations of empirically weights [20]; then α and β are calculated as follows:

$$\alpha = \frac{1 - (H(M_{\mathrm{IP1}})/\log_2(M_{\mathrm{IP1}}))}{(1 - (H(M_{\mathrm{IP1}})/\log_2(M_{\mathrm{IP1}}))) + (1 - (H(M_{\mathrm{IP2}})/\log_2(M_{\mathrm{IP2}})))},$$
$$\beta = \frac{1 - (H(M_{\mathrm{IP2}})/\log_2(M_{\mathrm{IP2}}))}{(1 - (H(M_{\mathrm{IP1}})/\log_2(M_{\mathrm{IP1}}))) + (1 - (H(M_{\mathrm{IP2}})/\log_2(M_{\mathrm{IP2}})))}. \tag{9}$$

The security situation assessment of IIoT nodes in IPAS, if the trust value of the node is larger than the threshold value, the node is considered to be trusted; otherwise, the node is untrusted, and the node operation is restricted to ensure the IPAS domain is trusted.

5. Group Signature Scheme and Security Analysis

5.1. Difficult Problems and Assumptions of the Group Signature. In hybrid IP/SDN architecture, a node in SAS transmits data to a remote node in IPAS, the sending node needs to show it is trusted to the remote node firstly, and then the remote node verifies the identity of the sending node and verifies the sending node is trusted to the superior node based on its trust value. We propose a remote attestation scheme using group signature that any member in the group can sign on behalf of it. Our group signature scheme is based on the q-SDH assumption and the concept is defined as follows.

Theorem 4 (Bilinear mapping). *Let G_1, G_2, and G_T are multiplicative cyclic groups of order prime p and g_1, g_2 are the generating elements of the group G_1, G_2. Given a mapping $e: G_1 \times G_2 \longrightarrow G_T$, for any $a, b \in_R \mathbb{Z}_p^*$, there exists $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$.*

Theorem 5 (Computational Diffie-Hellman problem, CDH). *Let there exist* $a, b \in_R \mathbb{Z}_p^*$ *and* $g, g^a, g^b \in G$. *Given the tuple* $< g, g^a, g^b >$ *under the unknown* a, b *condition, it is difficult to compute* g^{ab}.

Theorem 6 (q-strong Diffie-Hellman, q-SDH assumption). *Let* $a \in_R \mathbb{Z}p$. *Given as input a* $(q+1)$*-tuple* $(g, g^a, g^{a^2}, \cdots, g^{a^q}) \in G^{t+1}$, *for every adversary* $\mathscr{A}_{q-SDH}$, *the probability* $Pr[\mathscr{A}_{q-SDH}(g, g^a, g^{a^2}, \cdots, g^{a^q}) = (x, g^{1/(x+a)})] \geq \varepsilon$ *for any value of* $x \in \mathbb{Z}_p$, ε *is the negligible quantity.*

5.2. The Scheme of the Group Signature. The group signature scheme is firstly required to construct a group, consisting of a group manager GM and several group members U_i. In hybrid SDN, the SDN controller and IP gateway act as GM of their separate group, and the IIoT node as group member U_i to sign for external signatures, and the signed message is the global trust value of the node. GM generates the group public key Gpk and private key Gsk, U_i negotiates with GM the signature private key Usk_i, and GM adds the registration information item about U_i to the group registry Reg[i]. The node trust value Trust_{X_i} is signed by group public key Gpk and user private key Usk_i to group signature S_i. The verification of the signature is implemented by the group public key Gpk, the message Trust_{X_i}, and the group signature S_i. If S_i is the group signature of message Trust_{X_i}, the node is the trusted source of data trading and submits the trust value Trust_{X_i} to the superior node to judge whether the trust value is higher than the threshold value to judge whether secure data trading can be realized or not. If the identity of the signer is questioned, the signature can be opened to find out the identity of the group member based on the group signature S_i and the GM private key Gsk and the registry entry Reg[i]. When a group member revokes its signature from the group, GM gets a new revocation item by using the group public key Gpk, private key Gsk, and Reg[i] as input and adds it to the revocation list (RL). After the revocation, the new signature private key Usk'_i is calculated based on the member private key Usk_i in RL. The group signature scheme includes system parameter creation setup, signature, verification, and signature open and signature revoke processes.

5.2.1. Setup. Randomly select three bilinear mapping cyclic groups G_1, G_2, G_T of order large prime p, where the bilinear mapping of G_1, G_2, G_T satisfies $e: G_1 \times G_2 \longrightarrow G_T$, given $H_u: \{0,1\}^* \longrightarrow \{0,1\}^{n_u}$ and $H_m: \{0,1\}^* \longrightarrow \{0,1\}^{n_m}$ are collision-free Hash function that maps the message to the required length. The group manager GM selects the generation element of G_1, G_2 is g_1, g_2, then randomly selected the secret value $y \in_R \mathbb{Z}_p^*$, the group public key is $\text{Gpk} = g_1^y$, the group private key is $\text{Gsk} = y$, and the system parameters are $(g_1, g_2, p, \text{Gpk}, H_u, H_m)$.

GM selects $x_i \in_R \mathbb{Z}_p^*$ and computes g_1^{1/x_i+y} to send to group member U_i, and the user public key is $\text{Upk}_i = g_1^{x_i}$, and the user private key is $\text{Usk}_i = \sigma = g_1^{1/x_i+y}$. U_i uses Upk_i to interact with GM when joining the group. The ID of user U_i is a bit string of length n_m, $\text{ID} \subseteq \{1, 2, \cdots, n_u\}$. GM needs to authenticate the user U_i, using the group private key Gsk to sign the ID. Then, GM selects $u \in G_2$, the identity information is $R = g_2^y \cdot (u \prod_{j \in ID} u_j)$, using the user private key Us k_i to sign R as $\pi = (R^y)^{1/x_i+y}$. We send the signature $(R, \pi, (u \prod_{j \in ID} u_j))$ to GM for verification and use the public key Gpk and Upk_i to decrypt. If $e(\pi, \text{Upk}_i \cdot \text{Gpk}) = e(R, g_1)$, the signature is valid, and U_i is a legal member of the group, and GM adds the registry entry $\text{Reg}[i] = (\text{Upk}_i, \text{Usk}_i, x_i, \pi)$. Otherwise, the signature is invalid, and the user is rejected as a member of the group.

5.2.2. Signature. The group member U_i chooses $\alpha_1, \alpha_2, \delta_1, \delta_2, \gamma_1, \gamma_2, \gamma_3 \in_R \mathbb{Z}_p^*$, $\alpha = \alpha_1 + \alpha_2$, and $\eta = g_1 = g_2$ to compute $\lambda_1 = g_1^{\alpha_1}$, $\lambda_2 = g_2^{\alpha_2}$, and $\lambda_3 = \sigma \cdot \eta^{\alpha}$ with its private key and computes $d_1 = \lambda_1^{\delta_1} g_1^{-\gamma_1}$, $d_2 = \lambda_2^{\delta_1} g_2^{-\gamma_2}$, and $d_3 = e(\lambda_3, g_2)^{\delta_1} \cdot e(\eta, g_1)^{\delta_2} \cdot e(\eta, g_2)^{-(\gamma_1+\gamma_2)} \cdot e(g_1, g_2)^{\gamma_3}$. By calculating the trust value in the previous section, the global trust value Trust_i of the device node i is calculated to map it to a bit string $(m \prod_{j \in \text{Trust}} m_j)$ of length n_m; then, the digest value $c = H_m(\lambda_1 \| \lambda_2 \| \lambda_3 \| d_1 \| d_2 \| d_3 \| m \prod_{j \in \text{Trust}} m_j)$ is calculated for the corresponding group member U_i. Let $\omega_1 = \delta_1 + c$, $\omega_2 = \delta_2 - c \cdot \alpha$, $\omega_3 = \gamma_3 + c$, $\omega_4 = \gamma_1 + c \cdot \alpha_1$, and $\omega_5 = \gamma_2 + c \cdot \alpha_2$, and the signature of group member U_i is $S_i = (\lambda_1, \lambda_2, \lambda_3, c, \omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$.

5.2.3. Verification. Firstly, the identities of the sending node U_i and the remote node V_i of the two groups are verified. The remote verifier V_i receives the signature $S_i = (\lambda_1, \lambda_2, \lambda_3, c, \omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$ and trust value Trust_i from the group member U_i, and the remote verifier needs to verify the signature, and calculate $\widetilde{d_1} \stackrel{?}{=} \lambda_1^{\omega_1} g_1^{-\omega_4}$, $\widetilde{d_2} \stackrel{?}{=} \lambda_2^{\omega_1} g_2^{-\omega_5}$, $\widetilde{d_3} \stackrel{?}{=} e(\lambda_3, g_2)^{\omega_1} \cdot e(\eta, g_1)^{\omega_2} \cdot e(\eta, g_2)^{-(\omega_4+\omega_5)} \cdot e(g_1, g_2)^{\omega_3} \cdot e(\lambda_3, \eta)^c \cdot e(g_1, g_2)^{-c}$, and $\tilde{c} \stackrel{?}{=} Hu(\lambda_1 \| \lambda_2 \| \lambda_3 \| \widetilde{d_1} \| \widetilde{d_2} \| \widetilde{d_3} \| m \prod_{j \in \text{Trust}} m')$. If $\tilde{c}$ is equal to c, it indicates that S_i is a valid signature and the verifier queries GM whether the trust value is trusted. If the trust value is higher than the threshold value, it means that the source of the data sending node is trusted. Otherwise, the data trading is rejected.

5.2.4. Open. For traceability of signature, GM can open the signature and find the identifier of the signer i. Firstly, verify the signature S_i as a valid signature, then get λ_1, λ_2, and λ_3 from $S_i = (\lambda_1, \lambda_2, \lambda_3, c, \omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$, and get the private key of group member U_i by computing $\lambda_3/(\lambda_1 \cdot \lambda_2) = (\sigma \cdot \eta^{\alpha})/(g_1^{\alpha_1} \cdot g_2^{\alpha_2}) = (\sigma \cdot \eta^{\alpha})/\eta^{\alpha} = \sigma$. *GM* query σ at Reg[i]. GM searches σ at Reg[i] and tracks the identity i corresponding to Usk_i.

5.2.5. Revoke. If the node in hybrid SDN is untrusted, it cannot participate in the group signature. Let the group member who is revoked by GM is U_j, and after the revocation, there are t group members left in the group; then, GM needs to regenerate the group public key $\text{Gpk} = g_{1t}^{y_{1t}}$ and update the

group private key to $\text{Gsk} = y_{1t} = \prod_{t=1}^{t} y_{jt}$, update the member private key to $\text{Usk}_i = \sigma = g_{1t}^{1/x_i + y_{1t}}$, and renew Reg[$i$].

5.3. Security Analysis under the Standard Model. The group signature scheme meets the security requirements such as correctness, indistinguishability, anonymity, traceability, forward security, and unforgeability. Security proof under the standard model can provide better guarantees for the design of the group signature scheme [21].

5.3.1. Correctness. The challenger needs to verify the correctness of the signature when a signature is received. Verify whether the identity information is satisfied.

$$e(\pi, \text{Upk}_i \cdot \text{Gpk}) = e\left((R^y)^{1/(x_i+y)}, g_1^{1/(x_i+y)} \cdot g_1^y\right) = e(R, g_1). \tag{10}$$

The challenger selects $\alpha_1, \alpha_2, \delta_1, \delta_2, \gamma_1, \gamma_2, \gamma_3 \in_R \mathbb{Z}_p^*$ and sends it to the verifier, the challenger calculates $\lambda_1, \lambda_2, \lambda_3$, d_1, d_2, d_3, the verifier selects c as the challenge value, and the challenger calculates $\omega_1, \omega_2, \omega_3, \omega_4, \omega_5$. Then,

$$\tilde{d}_1 = \lambda_1^{\omega_1} \cdot g_1^{-\omega_4} = \left(g_1^{\alpha_1}\right)^{(\delta 1 + c)} \cdot g_1^{-(\gamma_1 + c\cdot\alpha_1)} = g_1^{\alpha_1 \cdot \delta_1} \cdot g_1^{-\gamma_1} = d_1,$$

$$\tilde{d}_2 = \lambda_2^{\omega_1} g_2^{-\omega_5} = \left(g_2^{\alpha_2}\right)^{(\delta_1 + c)} \cdot g_2^{-(\gamma_2 + c\cdot\alpha_2)} = g_2^{\alpha_2 \cdot \delta_1} g_2^{-\gamma_2} = d_2,$$

$$\begin{aligned}\tilde{d}_3 &= e(\lambda_3, g_2)^{\omega_1} \cdot e(\eta, g_1)^{\omega_2} \cdot e(g_1, g_2)^{\omega_3} \cdot e(\eta, g_2)^{-(\omega_4+\omega_5)} \\ &\quad \cdot e(\lambda_3, \eta)^{-c} \cdot e(g_1, g_2)^{-c} = e(\lambda_3, g_2)^{\delta_1 + c} \cdot e(\eta, g_1)^{\delta_2 - c\cdot\alpha} \\ &\quad \cdot e(g_1, g_2)^{\gamma_3 + c} \cdot e(\eta, g_2)^{-(\gamma_1 + c\cdot\alpha_1 + \gamma_2 + c\cdot\alpha_2)} \cdot e(\lambda_3, \eta)^{-c} \cdot e(g_1, g_2)^{-c} \\ &= e(\lambda_3, g_2)^{\delta_1} \cdot e(\eta, g_1)^{\delta_2} \cdot e(g_1, g_2)^{\gamma_3} \cdot e(\eta, g_2)^{-(\gamma_1 + \gamma_2)} \\ &\quad \cdot e(\lambda_3, g_2)^{c} \cdot e(\eta, g_1)^{-c\cdot\alpha} \cdot e(g_1, g_2)^{c} \cdot e(\eta, g_2)^{-(c\cdot\alpha_1 + c\cdot\alpha_2)} \\ &\quad \cdot e(\lambda_3, \eta)^{-c} \cdot e(g_1, g_2)^{-c} = d_3 \cdot e(\lambda_3, g_2)^{c} \cdot e(\eta, g_1)^{-c\cdot\alpha} \\ &\quad \cdot e(g_1, g_2)^{c} \cdot e(\eta, g_2)^{-c\cdot\alpha} \cdot e(\lambda_3, \eta)^{-c} \cdot e(g_1, g_2)^{-c} = d_3.\end{aligned} \tag{11}$$

The verification equations $d_1 = \tilde{d}_1$, and $d_2 = \tilde{d}_2$, and $d_3 = \tilde{d}_3$ are verified, and the group signature scheme is proved to be correct by the above derivation process.

5.3.2. Indistinguishability. Under the standard model, the challenger gets the system parameters and sends them to the attacker. The challenger can generate the group public key Gpk and the member private key Usk_i. The challenger selects the designcrypt oracle H_m from the private key σ_{i_0}, σ_{i_1} of U_{i_0}, U_{i_1} based on the given message Trust_i. If the challenger learns σ and chooses α_1, α_2 randomly, and computes the values of $\lambda_1 = g_1^{\alpha_1}$, $\lambda_2 = g_2^{\alpha_2}$, and $\lambda_3 = \sigma \cdot \eta^{\alpha}$, $(\lambda_1, \lambda_2, \lambda_3)$ is solved using a linear encryption scheme and generated by the simulator is indistinguishable from the actual distribution. The adversary randomly selects $(\lambda_1, \lambda_2, \lambda_3)$, due to $\omega_1 = \delta_1 + c$, when c, ω_1 is fixed, then δ_1 is fixed, when α_1 is fixed, then $\lambda_1 = g_1^{\alpha_1}$ is fixed, and the value of d_1 in $d_1 = \lambda_1^{\delta_1} g_1^{-\gamma_1}$ is consistent with the γ_1 distribution. Randomly selects c, ω_2, in $\omega_2 = \delta_2 + c \cdot \alpha$, when c, ω_2 is fixed, and α_2 is fixed,

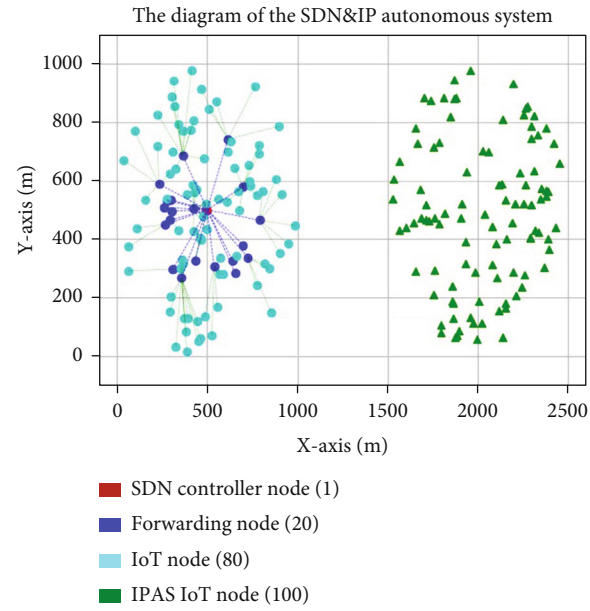


Figure 6: SAS and IPAS domain distribution and construction simulation (101 nodes and node centralized connection in SAS, 100 nodes and distributed connection in IPAS).

then $\lambda_2 = g_2^{\alpha_2}$ is fixed, due to $\alpha = \alpha_1 + \alpha_2$, then δ_2 is fixed, so in $d_2 = \lambda_2^{\delta_1} g_2^{-\gamma_2}$, the value of d_2 is consistent with the γ_2 distribution. Similarly, the value of d_3 is consistent with the γ_3 distribution. Therefore, it is difficult to distinguish the values obtained by the simulator from the actual values; therefore, the group signature algorithm satisfies indistinguishability.

5.3.3. Anonymity. The attacker is given the value $(\lambda_1, \lambda_2, \lambda_3)$ to (d_1, d_2, d_3), and the message Trust_i is known to get $c = H_m(\lambda_1 \| \lambda_2 \| \lambda_3 \| d_1 \| d_2 \| d_3 \| \text{trust}_i)$. $H_m : \{0,1\}^* \longrightarrow \{0,1\}^{n_m}$ is a collision-free Hash function; then, the attacker is able to correctly return the group signature $S_i = (\lambda_1, \lambda_2, \lambda_3, c, \omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$ to the challenger. The challenger can output the guessed value based on the returned signature and send the guessed value to the attacker. The attacker will give a reply with the guessed result. $(\lambda_1, \lambda_2, \lambda_3)$ is the guessed result of one of the $\sigma_{i_0}, \sigma_{i_1}$. If the challenger can attack the anonymity of the group signature with the advantage of ε, the attacker can also attack the linear encryption scheme with the advantage of ε. Therefore, the group signature scheme satisfies anonymity.

5.3.4. Traceability. Given λ_1, λ_2, and λ_3, the challenger is able to select $\alpha_1, \alpha_2, \delta_1, \delta_2, \gamma_1, \gamma_2, \gamma_3 \in_R \mathbb{Z}_p^*$ and calculate $(\lambda_1, \lambda_2, \lambda_3, d_1, d_2, d_3, c, \omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$. It is known from $\lambda_3/(\lambda_1 \cdot \lambda_2) = (\sigma \cdot \eta^{\alpha})/(g_1^{\alpha_1} \cdot g_2^{\alpha_2}) = (\sigma \cdot \eta^{\alpha})/\eta^{\alpha} = \sigma$ that even if α, α_1, and α_2 is unknowable, the private key of group member U_i can still be traced, which satisfies full traceability.

5.3.5. Unforgeability. The group signature is a double-layer signature, assuming that the challenger and the attacker forge the signature with each other, then the attacker first

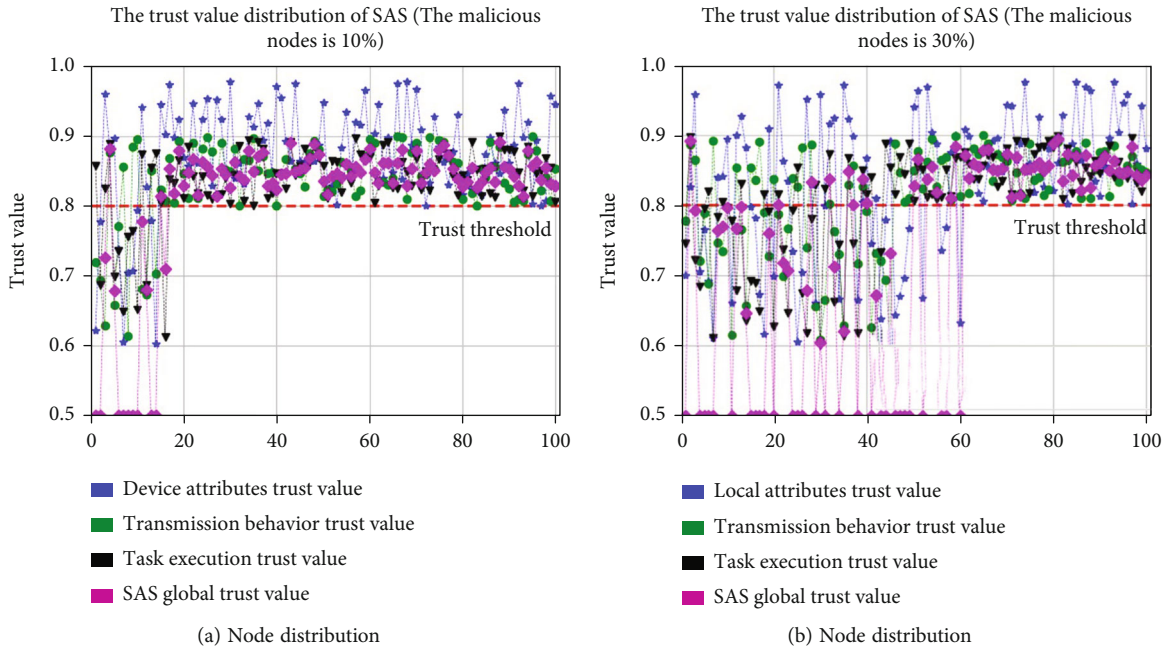


Figure 7: The process of calculating trust values in the SAS domain. (a) Distribution of each trust component when the number of malicious nodes accounts for 10%. (b) Distribution of each trust component when the number of malicious nodes accounts for 30%.

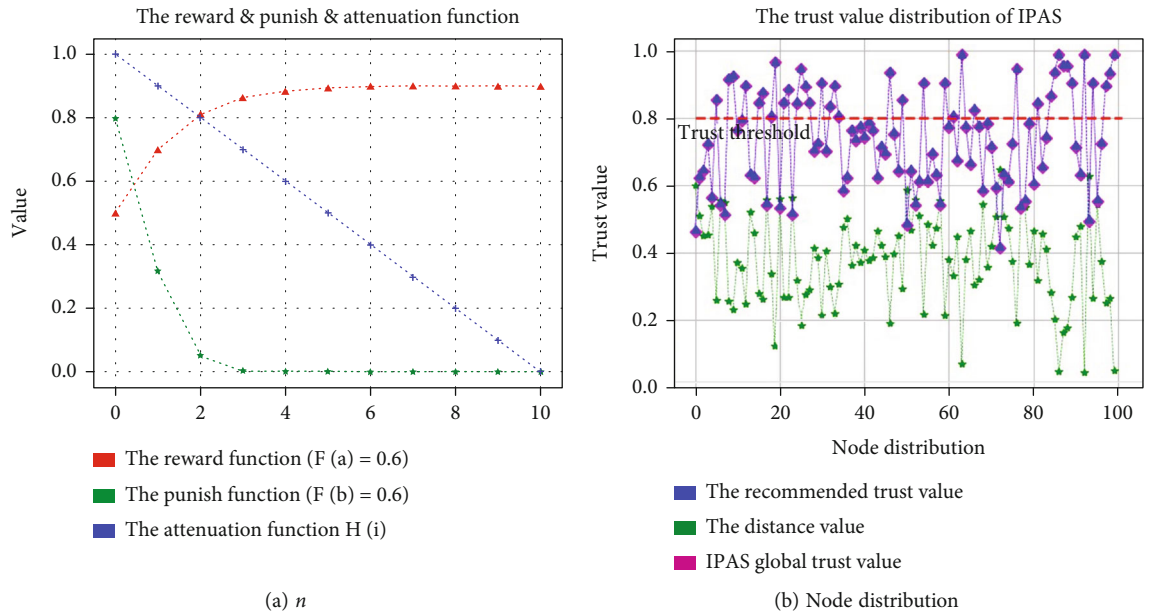


Figure 8: (a) Direct recommendation trust calculation component curves in the IPAS domain, including reward/punishment values and decay function values. (b) Indirect recommendation trust value calculation and distribution, judged by the distance function and direct recommendation trust.

uses the group private key Gsk to sign ID, the signature is completed by the group manager GM, and the group member U_i, and the signature is judged to be valid for the legitimate members of the group, and then, the message Trust_i is signed by S_i. In case the challenger sends a signature request to get a forged group signature $S_i^* = (\lambda_1, \lambda_2, \lambda_3, d_1, d_2, d_3, c, \omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$, when the signature is challenged, GM opens the signature to get σ_i^*, that is, the attacker forges a signature that can be traced to the identity. If $\sigma_i^* = \sigma_i$, $q - 1$ (σ_i, x_i) pairs are obtained according to the q-SDH assumption. When the signature is tracked with probability ε/n and no questions are asked by the challenger for i to the designcrypt oracle, the group signature algorithm satisfies unforgeability.

5.3.6. Resistance to Joint Attack. The group signature scheme is traceability and unforgeability, the only way to get the private key Usk_i of the group member is to sign legitimately,

Table 2: Performance comparison of signature schemes.

Scheme	Indistinguishability	Anonymity	Unforgivability	Traceability	Resistance to joint attack	Forward security	Revocability
[23]	√	√	√	×	×	×	×
[24]	√	√	√	√	×	×	×
[25]	√	√	√	√	√	×	×
[26]	√	√	√	√	√	√	×
[22]	√	√	√	√	√	√	√
Ours	√	√	√	√	√	√	√

and GM can open the signature to trace the signer U_i by the registry entry Reg[i]. In case multiple group members collude to forge a signature, GM traces the signature to discover the forger, so the group signature can resist joint attacks.

5.3.7. Forward Security. The signature key of any group member U_i is obtained Upk_i and Usk_i by selecting the secret value $x_i \in_R \mathbb{Z}_p^*$. In case the signature key at the t moment is Usk_t, if Upk_i and Usk_i before the t moment are obtained, each time the selected secret value is destroyed after use; hence, the attacker cannot deduce the previous signature key based on the key at the moment, and the group signature scheme has forward security.

6. Experiments and Analysis

The experiments use Python to simulate the remote attestation scheme between nodes in the hybrid SDN architecture, including the trusted judgment of nodes and the group signature scheme. The trust value of the node is calculated according to Section 3, and when the trust value is lower than the system setting threshold (0.8), the node is judged to be a malicious node, and its operation is restricted to communicate with other nodes. The efficiency comparison of the signature algorithm is completed according to Section 4.

We simulate a hybrid SDN architecture about SAS and IPAS, including SDN controller nodes, forwarding nodes, and IIoT nodes. The experiment set 201 nodes distributed in an area range of 2500 m × 1000 m, the nodes can be classified as SAS and IPAS domains, where 101 nodes (including 1 control node, 20 forwarding nodes, and 80 IIoT nodes) in SAS and 100 nodes in IPAS. In SAS, centralized management is used, the forwarding nodes are connected to the controller nodes, and the IIoT nodes are connected to the forwarding nodes. In IPAS, distributed connectivity is used. The layout of the simulation experiment is illustrated in Figure 6.

6.1. Analysis of the Correctness of Trusted Judgment

6.1.1. Trusted Judgment in SAS Domain. The calculation of trust value in SAS depends on device attributes trust value, forwarding behavior trust value, and task execution trust value, if hardware and software attributes are judged to be forged, the global trust value is directly judged to be untrusted, as illustrated in Figure 7, when device attributes trust value (blue dots) < 0.8, the global trust value (purple dots) is directly judged to be untrusted. Figure 7(a) shows the trust value distribution when the number of malicious nodes accounts for 10%, and Figure 7(b) shows the trust value distribution when the number of malicious nodes accounts for 30%. The comparison shows that the more malicious nodes have a greater impact on the global trust value within the SAS domain.

Table 3: Comparison of computational overhead of signature schemes.

Scheme	Signature length	Signature algorithm	Verification algorithm
[27]	2092 bit	6Exp	7Exp+EE
[21]	767 bit	4Exp	Exp+2EE
[28]	1662 bit	5Exp	5Exp+EE
Ours	1491 bit	4Exp	4Exp+EE

6.1.2. Trusted Judgment in IPAS Domain. In the IPAS domain, the node direct recommendation trust value is related to the reward/punishment factor and the decay factor, and the reward factor $F(a) = 0.6$ and punishment factor $F(b) = 0.6$ are illustrated in Figure 8(a). With the increasing judgment times n, if the node is judged as a trusted node with the reward factor $F(a) > 0.5$, the initial value of the reward is 0.5 (untrusted threshold). If the reward value which is continuously judged as a trusted node is increased exponentially and reaches the threshold after $n = 4$ times, and the reward value continues to be the same after the node is judged to be a trusted node again. If the node is an untrusted node with punishment factor $F(b) > 0.5$, the initial value of punishment is 0.8 (trust threshold); after the second judgment of untrusted, the punishment value is reduced to 0.5. And $n = 3$ times, it is still judged as the untrusted node, then the punishment value drops to 0. If it is judged as untrusted again, the punishment value is always 0. The initial value of the decay function is 1. As the recommendation times increase, the current trust value decays according to $H(i)$, as illustrated by the blue line in Figure 8(a).

The indirect recommendation of trust in IPAS is based on the distance function to calculate the similarity, and the green dots in Figure 8(b) show the distribution distance between the current node and the neighbor nodes; the closer the distance indicates the larger the indirect recommendation trust value. The distribution of the node trust value is

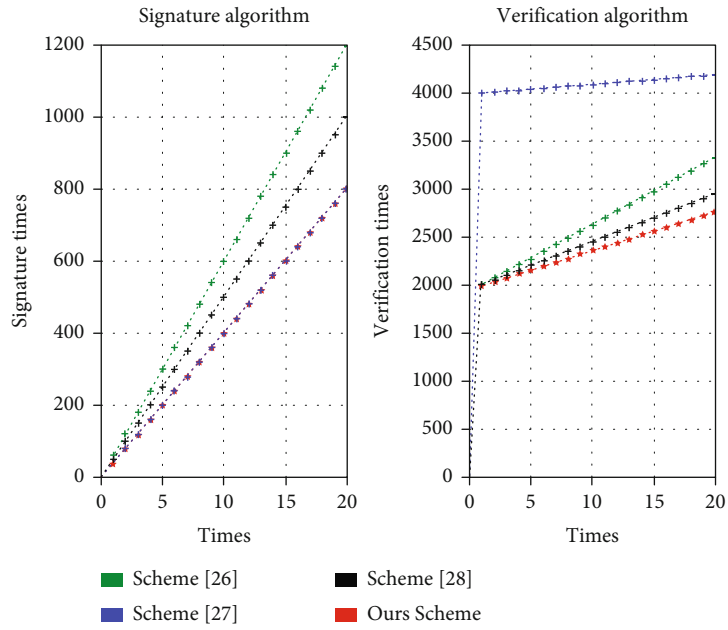


Figure 9: Computational performance comparison of signature and verification algorithms.

similar to that of the direct recommendation trust values, with only a difference in the coefficients (judged by the distance function), so the indirect recommendation trust values overall overlap with the direct recommendation trust values. In the IPAS domain, the larger the direct recommendation trust value of the node, the larger the global trust value, and the closer the distance, the larger the global trust value.

6.2. Analysis of the Performance of the Group Signature Algorithm. We use group signature to realize remote attestation and compare it with other schemes in terms of performance and efficiency. The group signature must satisfy not only correctness but also the three security properties of anonymity, traceability, and unforgeability [22]. As shown in Table 2, our scheme can fully satisfy indistinguishability, anonymity, traceability, unforgeability, resistance to joint attack, forward security, and revocability. Our scheme and [22] satisfy the revocability of group members, and our scheme group public key and user private key are shorter and more suitable for IIoTs.

Our group signature scheme compares the performance by the computational overhead and communication overhead of the signature and verification processes. EE denotes the bilinear pair e operation, and Exp denotes the modular power operation. Since $e(g_1, g_2)$ can be precomputed, it can be neglected in the validation algorithm. Therefore, in our scheme signature algorithm, $d_1 = \lambda_1^{\delta_1} \cdot g_1^{-\gamma_1} = g_1^{\alpha_1 \cdot \delta_1} \cdot g_1^{-\gamma_1} = g_1^{\alpha_1 \cdot \delta_1 - \gamma_1}$ needs to perform 1 modulus power operation on g_1, in the same way, $d_2 = \lambda_2^{\delta_1} \cdot g_2^{-\gamma_2} = g_2^{\alpha_2 \cdot \delta_2 - \gamma_2}$ performs 1 modulus power operation on g_2. In d_3 due to $e(\lambda_3, g_2)^{\delta_2} = e(\sigma \cdot \eta^{\alpha}, g_2)^{\delta_2} = e(\sigma, g_2) \cdot e(\eta, g_2)^{\alpha+\delta_2}$, if it is ignored for e operation, only 1 modulus power operation on g_1 and 1 modulus power operation on g_2 are required in d_3. Simultaneously, the verification algorithm needs to perform only 4 modulus power operations and 1 e operation. Let the order in the group p be 170 bit, the length of the Hash algorithm H_u, H_m is 128 bit, and the length of the elements in the group G_1, G_2, G_T is 171 bit. The comparison of the signature length and computational overhead of each scheme is shown in Table 3.

As can be seen from Table 3, [21] has the shortest signature length and the signature algorithm has the same computational overhead as our scheme. Comparing the validation algorithm, the computational overhead of the e operation is much greater than the modulus power operation, so our scheme verification algorithm has the least computational overhead, as illustrated in Figure 9. Therefore, our scheme has a short signature length and minimal computational overhead for the signature and verification algorithms.

6.3. Analysis of Security Situation Assessment. The security situation assessment is available in SAS and IPAS to learn the security states in SAS and IPAS. Figure 10 shows the variation of the trust value with malicious nodes. As the malicious nodes increase, the trust value reduces more and more. The domain's initial trusted value is 1, and all the nodes in SAS or IPAS are trusted. When the number of malicious nodes reaches 20%, the global trust value of SAS or IPAS remains above the trust threshold (0.8). The SAS intradomain trust value is computed based on the device attributes trust value, and the malicious node is judged by the management node. There are 20 malicious nodes, and the SAS global trust value is directly reduced to 0.8. The IPAS domain global trust value is defined by the number of interactions of nodes, and malicious nodes may still interact with neighbor nodes with a global trust value greater than 0.8. Malicious nodes in the distributed IPAS domain need to combine the multiple nodes security situation

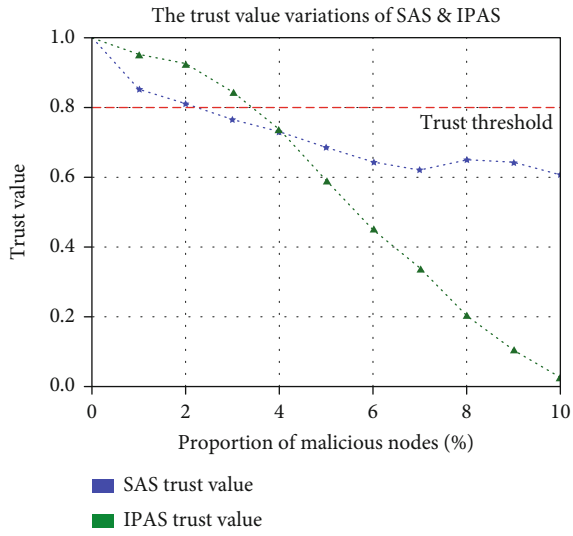


Figure 10: SAS and IPAS domain trust values with malicious node variation.

assessment judgment including the malicious node discovery, repair, and exclusion process. When the number of malicious nodes is greater than 30%, both the SAS domain and IPAS domain are judged to be untrusted. Therefore, the domain security situation is represented by calculating the global trust value in a domain.

7. Conclusion

The data in IIoTs is derived from different types of heterogeneous networks. A hybrid SDN architecture is used to realize the management of IIoT heterogeneous networks which are divided into different autonomous systems (SAS and IPAS). To guarantee IIoT data trading source is trusted, firstly, the standardized models are established about the attributes and interaction behaviors of nodes in SAS and IPAS for realizing the process of trusted judgment for nodes in the domain to calculate the trust value for the IIoT nodes. Secondly, the node identity and trust value are used in a revocable group signature scheme which can balance privacy and security, to complete the remote attestation of IIoT nodes and guarantee the secure data trading for different domains. Finally, the simulation experiment verifies the trusted judgment mechanism and the reward/punishment mechanism, decay function, and security situation assessment are correct which is capable to reflect the dynamic operation states of IIoT nodes. The performance and efficiency verification of the group signature scheme is analyzed by comparing the computational overhead of the signature algorithm and validation algorithm. Our scheme has good performance in trusted judgment, group signature, and verification process and can effectively confirm the security of the IIoT nodes.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Basic Research Program of China (973 Program) (2019YFB2102303), National Natural Science Foundation of China (NSFC 61971014 and NSFC 11675199), Young Backbone Teacher Training Program of Henan Colleges and Universities (2021GGJS170), and Henan Province Higher Education Key Research Project (23B520014).

References

[1] "Iot analytics," 2021, https://soft.chinabyte.com/71/725831571.shtml/.

[2] Y. S. Sandhya and K. Haribabu, "A survey: hybrid SDN," *Journal of Network and Computer Applications*, vol. 100, pp. 35–55, 2017.

[3] P. Shrivastava and K. Kataoka, "Topology poisoning attacks and prevention in hybrid software-defined networks," *IEEE Transactions on Network and Service Management*, vol. 19, no. 1, pp. 510–523, 2022.

[4] R. Wang, Z. Zhang, Z. Zhang, and Z. Jia, "ETMRM: an energy-efficient trust management and routing mechanism for SDWSNs," *Computer Networks*, vol. 139, pp. 119–135, 2018.

[5] M. H. C. U. I. Shijian and Y. I. Peng, "CAOHS: control architecture on hybrid software defined networks," *Journal of Information Engineering University*, vol. 20, pp. 257–262, 2019.

[6] T Group, "Line spacing in latex documents," https://www.Trustedcomputinggroup.org/home/2013.

[7] Z. H. Caiqiu, Y. A. Yuwang, and W. A. Yongjian, "Behavior measurement scheme for the wireless sensor network nodes," *Journal of Tsinghua University (Science and Technology)*, vol. 57, no. 1, pp. 39–43, 2017.

[8] R. Xie, C. Xu, C. He, and X. Zhang, "A new group signature scheme for dynamic membership," *International Journal of Electronic Security and Digital Forensics*, vol. 8, no. 4, pp. 332–351, 2016.

[9] Y. Liu, Y. Kuang, Y. Xiao, and G. Xu, "SDN-based data transfer security for internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 257–268, 2018.

[10] R. Zhou, Y. Lai, Z. Liu, and J. Liu, "Study on authentication protocol of SDN trusted domain," in *IEEE twelfth international symposium on autonomous decentralized systems*, pp. 281–284, Taichung, Taiwan, 2015.

[11] J. Chen, B. Gong, Y. Wang, and Y. Zhang, "Construction of internet of things trusted group based on multidimensional attribute trust model," *International Journal of Distributed Sensor Networks*, vol. 17, no. 1, Article ID 1550147721989888, 2021.

[12] G. Zheng, B. Gong, and Y. Zhang, "Dynamic network security mechanism based on trust management in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 6667100, 10 pages, 2021.

[13] J. Yu, G. Wang, Y. Mu, and W. Gao, "An efficient generic framework for three-factor authentication with provably secure instantiation," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2302–2313, 2014.

[14] B. Gong, J. Liu, and S. Guo, "A trusted attestation scheme for data source of internet of things in smart city based on dynamic trust classification," *IEEE Internet of Things Journal*, vol. 8, no. 21, pp. 16121–16141, 2021.

[15] J. Liu, B. Gong, and Q. Wang, "A trusted proof mechanism of data source for smart city," *Future Generation Computer Systems*, vol. 128, pp. 349–364, 2022.

[16] Z. Zhang, R. Wang, X. Cai, and Z. Jia, "An SDN-based network architecture for internet of things," in *In 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pp. 980–985, IEEE, 2018.

[17] Y. Chen and Z.-m. Duan, "Congested link inference algorithms in dynamic routing IP network," *Mathematical Problems in Engineering*, vol. 2017, Article ID 6342421, 17 pages, 2017.

[18] X. Huang, S. Cheng, K. Cao, P. Cong, T. Wei, and S. Hu, "A survey of deployment solutions and optimization strategies for hybrid SDN networks," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1483–1507, 2019.

[19] P. Jin, X. Hao, X. Wang, and L. Yue, "Energy-efficient task scheduling for CPO-intensive streaming jobs on Hadoop," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 6, pp. 1298–1311, 2019.

[20] L. Xie and R. Wei, "Dynamic trust evaluation method for IoT nodes," *Journal of Computer Applications*, vol. 39, no. 9, pp. 2597–2603, 2019.

[21] Z. Y. L. I. Yanqiong and L. I. Jiguo, "Certificateless signature scheme without random oracles," *Journal on Communications*, vol. 36, pp. 1–10, 2015.

[22] R. Xie, C. Xu, C. He, and X. Zhang, "An efficient dynamic group signature with non-frameability," *KSII Transactions on Internet and Information Systems*, vol. 10, no. 5, pp. 2407–2426, 2016.

[23] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Advances in Cryptology – CRYPTO 2004*, pp. 41–55, Springer, Berlin, Heidelberg, 2004.

[24] C. Delerablée and D. Pointcheval, "Dynamic fully anonymous short group signatures," in *International Conference on Cryptology in Vietnam*, pp. 193–210, Springer, 2006.

[25] B. Gong, X. Zhang, Y. Cao, Z. Li, J. Yang, and W. Wang, "A threshold group signature scheme suitable for the Internet of Things," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 13, 2021.

[26] Y. Wu, B. Gong, and Y. Zhang, "An improved efficient certificateless hybrid signcryption scheme for internet of things," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 6945004, 11 pages, 2022.

[27] J. Y. Hwang, S. Lee, B.-H. Chung, H. S. Cho, and D. Nyang, "Group signatures with controllable linkability for dynamic membership," *Information Sciences*, vol. 222, pp. 761–778, 2013.

[28] B. Gong, Y. Zhang, and Y. Wang, "A remote attestation mechanism for the sensing layer nodes of the Internet of Things," *Future Generation Computer Systems*, vol. 78, pp. 867–886, 2018.