An Industrial-Grade API Secure Access Gateway in the Cloud-Edge Integration Scenario

Sai Liu[1(✉)], Zhen-Jiang Zhang[2], Yong Cui[3], and Yang Zhang[1]

[1] Department of Electronic and Information Engineering, Key Laboratory of Communication and Information Systems, Beijing Municipal Commission of Education, Beijing Jiaotong University, Beijing 100044, China
516742786@qq.com, zhang.yang@bjtu.edu.cn

[2] Department of Software Engineering, Key Laboratory of Communication and Information Systems, Municipal Commission of Education, Beijing Jiaotong University, Beijing 100044, China
zhangzhenjiang@bjtu.edu.cn

[3] Thunisoft Information Technology Co. Ltd., Beijing 100044, China
cuiyong@thunisoft.com

Abstract. In recent years, the Internet of Things technology has developed rapidly. Due to the large number of devices at the edge, the wide distribution range, and the complex environment, cloud computing and edge computing failed to fully consider security risks at the beginning of the combination, and traditional protection methods can no longer fully meet their security requirements. The establishment of a new cloud-edge integrated security system is of great significance for ensuring the data and privacy of Internet users. This article first investigates the current status of traditional network security and analyzes its inherent shortcomings, and analyzes the organizational structure and main advantages of the zero-trust network. Designed a security certification system that meets the needs of cloud-edge integrated applications. The API security access gateway part of the system is designed and implemented. According to the type of client access request, it is equipped with multiple authentication methods. It also realized the functions of reverse proxy, load balancing, flow control, log audit, analysis and monitoring of microservices, and finally developed a supporting UI management tool based on Vue. This design provides a new set of secure access solutions for clients and microservices, which has produced good industrial benefits. It is of great significance to promote the development and popularization of cloud-edge integration.

Keywords: Cloud-Edge Integration · API Gateway · Secure Access · Zero Trust · Authentication

1 Introduction

Today's world is increasingly digitized, diversified and interconnected in the context of the Internet of Everything, and almost everything has the ability to process data [1]. The

Y.-B. Lin and D.-J. Deng (Eds.): SGIoT 2020, LNICST 354, pp. 57–69, 2021.
https://doi.org/10.1007/978-3-030-69514-9_6

distribution and intelligence of terminal equipment are more significant. In the Internet of Things, if the massive amounts of data generated by the devices are uploaded to the cloud platform for processing, it will bring tremendous pressure to the cloud. Part of the data processing work can be performed at the edge nodes [2], but the processed data still needs to be uploaded to the central cloud for big data analysis and training and upgrading of algorithm models. The upgraded algorithm is then transferred to the front-end equipment for updating, completing the closed loop of the work [3].

In many modern scenarios, cloud computing and edge computing will form a cooperative and complementary relationship. The edge is mainly responsible for processing real-time, short-period data and performing local services, which can reduce data transmission delays and network bandwidth costs, and provide resources Services such as scheduling and distribution; cloud computing is mainly responsible for computing tasks that are difficult for edge nodes, such as optimizing business rules or models, and completing application lifecycle management. According to estimates by Uptime, by 2021, half of all workloads will be run on the cloud and network edge outside the data center. The characteristics of real-time and fast data processing and analysis, network bandwidth saving, offline operation, and high data security are fully reflected in various scenarios of cloud-edge collaboration.

Today, the scale and complexity of terminal devices and applications are growing exponentially. According to IDC estimates, the number of global Internet devices will reach 48.9 billion in 2023; according to the Cisco VNI forecasting tool and the visual network index, from 2017 to 2022, global business mobile data traffic will increase six times, and each personal computer in the network. The average weekly data usage is about 15 GB. According to Metcalfe's law, the connectivity between systems, users, applications, and devices will become more fragmented and complex as devices and users continue to join. The value and importance of network security will continue to increase, and the protection and management of the network will Becomes more important and difficult.

In summary, the highly dynamic and heterogeneous environment at the edge of the network and the numerous and complicated data on the cloud have exacerbated the difficulty of network protection. Research on cloud and edge security technologies can effectively prevent security issues in data, privacy, network, and off-site storage brought by products in services, and ensure data and privacy security for Internet users [4]. Therefore, studying the security of cloud-edge integration application is the primary prerequisite for the further development of the Internet of Everything system, which is of great significance for promoting the development of cloud-edge integration.

2 Relation Work

In this section, we will survey the related works available in the literature. In [14], the authors used blind signatures and short, randomizable signatures to provide conditional anonymous authentication. They used powerful third parties to register entities and generate a certificate for each customer, control center, and fog nodes. However, the proposed solution consumed the computational power of resource-limited edge devices to generate secret keys from public and private keys. The authors of [15] provided a secure and privacy-preserving mutual authentication solution for an Elliptic Curve Cryptography (ECC) fog-based publish-subscribe system. The proposed solution could ensure

mutual authentication between subscribers and brokers, as well as between publishers and brokers. However, the proposed solution still consumed the computational power of the resource-limited edge devices. The authors of [16] introduced three Lightweight Anonymous Authentication Protocols (LAAPs). They use lightweight cryptographic primitives, such as one-way functions and EXCLUSIVE-OR operations, which led to a limited computational cost for the resource-limited edge devices. They also introduced a novel privacy protection security architecture for the D2D-supported fog computing model, which allows end-user devices to be authenticated without the intervention of a central server. However, the proposed architecture and protocols are used to validate each edge device, network access device, and centralized cloud server. They did not consider user authentication, which is responsible for managing and maintaining the system.

Edge computing has been defined by the ECC as an open platform deployed at the edge of the network near the data source and offering intelligent services for real-time processing, data optimization, security, and privacy within the mobile edge network infrastructure [18]. To cope with the above issues, a Lightweight Edge Gateway for the Internet of Things architecture [17] has been introduced, which is based on the modularity of microservices, in order to guarantee scalability and flexibility. In [19], the author presented an intelligent IoT gateway which can communicate with different networks, has a flexible protocol that converts different sensor data into a consistent format, and has a uniform external interface.

There exist several IoT platforms which provide a connection with IoT devices. Intel provides Open VINO, a deep learning toolkit that focuses on the edge, and uses visual data to gain insight into the business. Google's Edge TPU chip and Cloud IoT Edge software stack can deploy machine learning functions on edge devices to operate on data in real time. Microsoft's Azure IoT Edge product extends cloud analysis capabilities to the edge and supports offline use. Amazon launched the AWS Green grass software to extend AWS to devices, process the data generated by the terminal locally, and perform analysis, management and persistent storage. The Link IoT Edge platform launched by Alibaba can be deployed in smart devices and computing nodes of different levels, connecting devices with different protocols and data formats, and providing efficient, safe, and intelligent communication and connection capabilities. CDN Edge, launched by Tencent, sinks data center services to edge nodes, reducing user response delay and data center network load. Baidu's intelligent edge BIE can exchange data with Baidu Cloud, filter and calculate sensitive data, and provide temporary offline independent computing services. Huawei's IEF platform extends the AI capabilities of Huawei Cloud to the edge, supports heterogeneous hardware access, and provides a safe and reliable business mechanism. It is a complete edge-cloud collaborative integrated service solution.

In addition, industrial companies are actively exploring the field practice of edge computing based on rich industrial scenarios [5]. COSMO Edge, a one-stop equipment management platform developed by Haier, supports the analysis of multiple industrial protocols, provides strong equipment connection and data processing capabilities, provides digital modeling and EaaS application models, and helps industries such as steel, petrochemicals, and electronics manufacturing.

An API gateway is an entry point for forwarding requests between many microservices, which merges multiple microservice APIs into a single client and and routes the requests from one access point to the correct microservices. The API gateway uses an existing identity management and authentication service which manages accounts, such as JWT or OAuth2.0, to allow a user or client access to certain microservices. An API gateway is a service that publishes multiple APIs, updates the published API set at runtime, and is integrated with health check, load balancing, service monitoring, and security capabilities.

3 Cloud Edge Integrated Security Certification System

3.1 Related Cybersecurity Architecture

The basic idea of the traditional network security model is to protect important resources in the network by building layers of defense. The network is divided into different areas according to the degree of trust, and the areas are separated by firewalls. Since the firewall only checks whether the source address and destination address are correct when executing the security policy, the attacker can deliver remote access tools to the internal network to obtain access rights, and then move laterally in the internal network to find valuable resources [6]. Modern network design and application models greatly weaken the protection capabilities of border-based security strategies.

In a "zero trust" network, no matter where the host is, it is regarded as an Internet host and is in a dangerous network environment. Combined with distributed strategies, the network security architecture shown in Fig. 1 can be constructed.

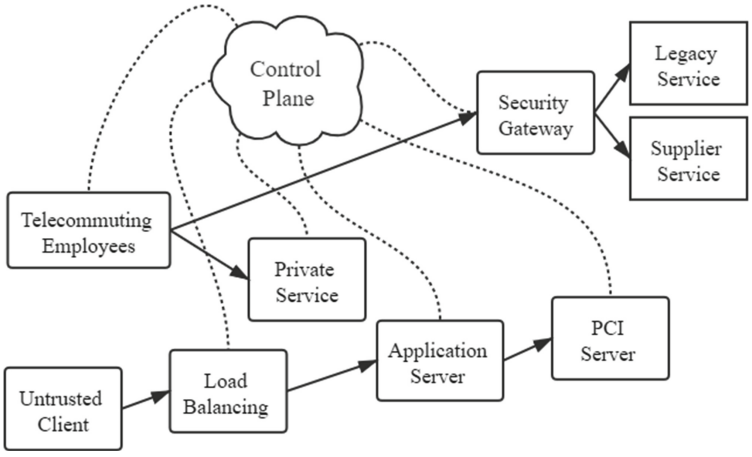

Fig. 1. "Zero Trust" Network Security Architecture Diagram

3.2 Cloud Edge Integrated Security Model

The Zero Trust model has improved the level of security to a certain extent, and even solved the contradiction between security and ease of use. Therefore, in combination with the security authentication requirements of the cloud-edge integrated computing service platform in this project, we have designed a network security architecture based on comprehensive identification and controlled by a unified identity authentication center and a trusted access gateway, [7] as shown in Fig. 2.

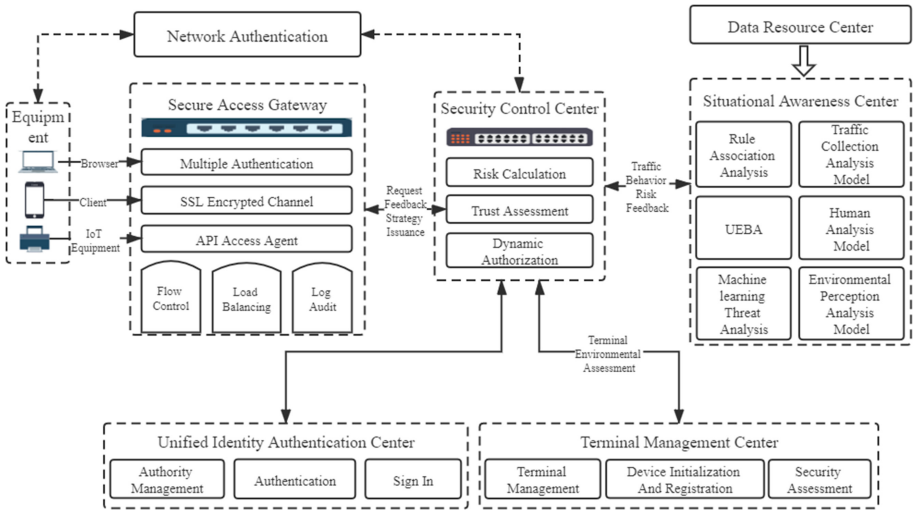

Fig. 2. Cloud Edge Integrated Security Authentication Platform Architecture Diagram

This architecture is mainly composed of the following 8 functions:

Security Control Center. According to the strategy, the authorization level is dynamically adjusted, and the "control center" conducts unified risk calculation, trust evaluation and dynamic authorization.

Risk Calculation And Trust Assessment. All access requests from the client must first go through the trust evaluation module, which conducts trust evaluation and risk calculation from the four dimensions of identity, equipment, environment, and behavior, and grants the access request corresponding trust levels based on the evaluation results.

Dynamic Authorization. Based on RBAC + ABAC, dynamic access control is established between the visiting subject and the visiting object. In terms of access subjects, this architecture will confirm the credibility of the user's identity. Basic authentication and authorization are based on user name and password authentication, while applying for higher authority requires verification with factors such as text messages, fingerprints, and faces. In addition, it is necessary to confirm whether the environment and behavior are credible, and continue to conduct behavior credibility testing based on user access behavior, because credible users also have huge security risks in dangerous environments [8]. In terms of access to objects, this architecture is based on static authorization of roles

and organizational structures, combined with the trust levels of subjects and objects, to achieve dynamic access authority control.

Secure Access Gateway. The API secure access gateway, which authenticates, authorizes and encrypts all accesses, is the connection center between front-end access and back-end microservices. All access requests to the back-end microservices are proxied by the secure access gateway, which implements multiple identity authentication, load balancing, log auditing and other functions for the requester.

Security Situation Awareness Center. After the start of the business, the Perception Center uses various models to continuously collect global real-time traffic, detect internal threats, monitor attack behaviors such as Trojan horses and viruses, and adjust the trust level according to the monitoring situation in real-time linkage with the access control system to control access and access permissions.

Terminal Management Center. Based on the AI multi-dimensional funnel-type terminal environment detection framework, the terminal management center continuously evaluates the terminal environment from the operating system, file system, application, process status and other levels, and feeds the results back to the security control center.

Unified Identity Authentication Center. Contains identity authentication services and single sign-on functions. Based on CA authentication, the user's behavior, social, biological and other attributes are used to construct its digital identity, and the multi-factor authentication (MFA) method is introduced for multiple verifications.

Authority Management. According to the model, the trust level is calculated, the access authorization is judged, and the user is finally assigned a minimum access authority.

The system has many verification links, short trust validity time, more fine-grained access control, and higher data security and privacy protection capabilities. Therefore, it can be widely used in edge computing platforms, and has outstanding advantages such as high service efficiency, comprehensive identity, dynamic access control, abnormal behavior and traffic monitoring.

3.3 API Secure Access Gateway Solution

The client may need to call multiple service interfaces to complete a business requirement. The API gateway can classify, forward, and organize external requests, avoiding cross-domain requests, complicated authentication, difficulty in reconstruction, and firewall restrictions [9]. According to the function of the API gateway, the architecture diagram is designed as shown in Fig. 3:

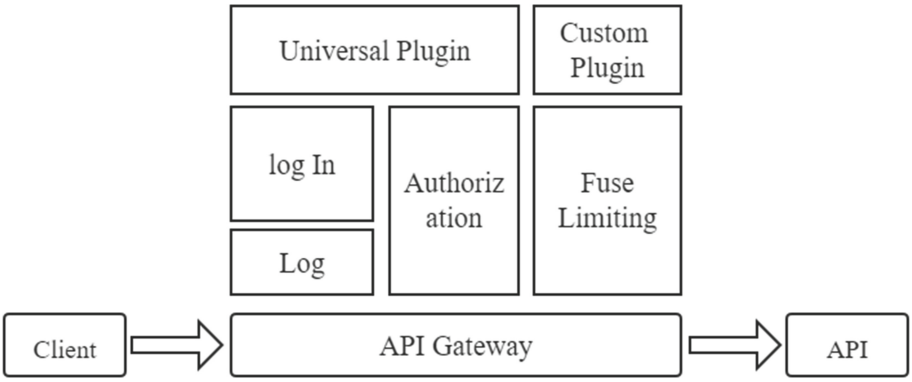

Fig. 3. API gateway architecture diagram

There are many open source API gateways, and the commonly used ones are Kong, Traefik, Ambassador, Tyk, Zuul, etc. Kong is a configurable gateway based on Nginx, mainly used for enterprise-level API management. The others are microservice gateways, providing decentralized self-service. The open source communities of Kong and Traefik are relatively active and their performance is relatively leading. In addition, compared to Traeflk and Zuul, Kong can extend functions such as identity authentication, fusing, current limiting, retry, load balancing, and health check through plug-ins. Kong has a unique architectural advantage. In terms of authentication, Kong supports Basic Auth, HMAC, JWT, Key, LDAP, OAuth 2.0 and other methods compared to other gateways. This design is implemented based on the more mature Kong gateway.

4 Implementation

By adding servers for horizontal expansion, the API secure access gateway can handle any type of request with low load. Modularization can be achieved by configuring plug-ins through Restful Admin API. The gateway can run on any infrastructure. This experiment is performed on a server with Ubuntu 18.04 system installed.

We built an Nginx development environment based on Kong Server and configured a relational database PostgreSQL. The gateway implements a reverse proxy for client requests based on Nginx. It reduces the pressure on the back-end web server and improves the response speed. In addition, we have also expanded the functions of the API gateway in terms of identity authentication, load balancing, flow control, log auditing, analysis and monitoring.

4.1 Authentication

It is usually controlled by related authentication plugins combined with configuration parameters, and requests for authentication failure or no authentication are rejected [10]. The general scheme of these plugins is as follows:

① Add Auth plugin to an API or globally;
② Create a consumer object;
③ Provide consumers with the identity authentication credentials required by the authentication plugin;
④ When there is a request to access the server, it will check whether the authentication credentials provided by it are correct. Once the authentication fails or the authentication fails, the request will be locked and the upward forwarding operation will not be performed.
⑤ When using an external authentication scheme, the gateway needs to be authenticated in conjunction with related external servers.

Our API secure access gateway is mainly equipped with the following 6 authentication methods:

Basic Auth: It is mostly used for verification of web request service. Set the user name and password on the server. When the client sends the request, enter the user name and password in the header. After receiving the request, the server will verify the request. If the verification is passed, the next step will be processed. Otherwise, "Invalid authentication credentials" will be returned.

Key-Auth: It can be used on services or routing. Before adding this plugin, all requests were proxied upstream. After setting the key for the microservice, the API can only be accessed with the correct key.

JWT-Auth: JSON Web Token as a JSON object ensures the security of information transmission between parties. The user calls the third-party interface when logging in. After logging in, a JWT token will be generated, and the returned JWT token will be put into Headers. The JWT authentication plug-in will parse out the login information before encryption at the next request, and then access the corresponding microservice. It can be used in scenarios such as authorization and information exchange.

OAuth 2.0: Used to authorize third-party applications. After the data owner agrees to the access of the third-party application, the system will generate a short-term token for the authentication of the third-party application [11].

LDAP-Auth: Add LDAP bind authentication to a route with username and password protection, and the plugin will check for valid credentials in the Proxy-Authorization and Authorization headers. All user information is stored in the LDAP server. When the user uses internal services, the LDAP server can change the original authentication strategy and must perform unified identity authentication through LDAP [12].

IP Restriction: By adding IP addresses to a whitelist or blacklist to restrict access to services or routes for some requests.

4.2 Load Balancing

The gateway uses a ring load balancer, and the addition and deletion of back-end services are handled by the gateway, without receiving updates from DNS. Ring load balancing is accomplished by configuring upstream and target entities. By default, the ring load balancer uses a weighted round-robin scheme and an IP address-based hash algorithm to implement load balancing.

When the application services deployed in multiple servers managed by load balancing are the same, two different service ports need to be opened to distinguish requests and distribute them to different ports according to the load balancing algorithm. When there are multiple back-end services, you need to use each node as a Target and set a load weight for it. The server with a higher weight is more likely to be accessed, and the server with a lower weight is less likely to be accessed.

4.3 Flow Control

There are actually two aspects of current limiting: rate limiting and request body size limiting. The more important one is the current limiting of request body size, because too large data volume can easily cause memory overflow exceptions. There are mainly three solutions:

Request-Termination: Use the specified status code and message to terminate the incoming request and fuse the specified request or service. This allows temporarily stopping the service or traffic on the route, or even blocking users.

Rate-Limiting: Control the maximum number of calls to an API interface service in a unit time. Once the limit is exceeded, the gateway will deny access and return an error message.

Request Size-Limiting: Used to limit the size of the request body. When the request body exceeds the threshold (such as 128M), the Kong gateway will reject the request.

4.4 Log Audit

We use the following 3 methods to comprehensively record the information about the service, routing and client request process on the API gateway to facilitate recording and review of user behavior:

Syslog: A standard that uses Internet protocols to transfer recorded document messages on the Internet. It is usually used for information system management or information security auditing. It can integrate log records from many different types of systems into a database.

File-Log: Write the related HTTP or HTTPS request and its response data to the log file on the disk.

Http-Log: ySend request and response logs to HTTP server. For HTTP service requests, the input and output message information of the request can be recorded in detail,.

4.5 Monitoring Alarm

It provides an open and complete monitoring solution by integrating the Prometheus system. A new model based on centralized rule calculation, unified analysis and alert notification has been formed. The core part of the program has only a single binary file, and there is no third-party dependency, so there is no risk of cascading failures. Based on its rich Client library, users can obtain the true running status of services and applications, such as CPU share. For some complex situations, it can also use Service Discovery capabilities to dynamically manage monitoring targets. In addition, intuitive information such as system operating status, resource usage, and service operating status can be directly obtained by connecting visual tools.

The server is started as a process. The data collected each time is called metrics. These data will be stored in the memory, and then periodically written to the hard disk. When the service restarts, the hard disk data is written back to the memory, so there is a certain consumption of memory.

The client uses the pull method to actively pull data. Use Node Exporter to collect the current host's system resource usage and other related data. It can process hundreds of thousands of data points per second, a single server can process millions of monitoring indicators, and has very efficient data processing capabilities. The built-in data query language PromQL of the program can realize the query and aggregation of monitoring data.

Through continuous collection and statistics of monitoring sample data, long-term trend analysis can be performed, and the time required for resource expansion can be predicted in advance. To track and compare the system, you can analyze the operating resource usage, concurrency and load changes under different capacity conditions. When a failure occurs or is about to occur, the system can respond quickly and notify the administrator through SMS, Dingding, WeChat messages, etc., to avoid affecting the business. Through the analysis of historical data, the cause of the failure can be found and the root cause can be solved.

4.6 Dashboard

Based on the VUE framework, we encapsulate the back-end API secure access gateway and other monitoring software services. Modularized the functions of the API gateway. Use Element UI framework to layout and display all pages. All modules are based on ajax technology. The front-end page sends an ajax request to call the API interface of the gateway and monitoring software. Get the response data and present the data on the page.

In all pages, we use a dynamic routing mechanism to determine whether the user has permission. If the user is not logged in, the system will force him to jump to the login page for authentication. For pages with large amounts of data, we use the paging tool of the Element UI framework to implement the paging function, and use Vuex to control the number of pages when the page jumps. Physical information, service information and log information are displayed using the dashboard directory.

We use the Element UI framework to build the layout of the login box. Use Element UI tabular tools to display data, such as host operating status, system physical information, terminal operating status, alarm information and log information.

Use the ECharts tool to build a visual chart, and then call the gateway interface to display the service information data. In the plug-in module, we use the column tool to classify various plug-ins. We can obtain the data types of all plug-in configuration items based on the pattern API interface provided by the gateway. We only need to design all possible data types in the <template> tag, and use v-if statements for conditional judgment. When you select the plug-in to be added, the page will automatically filter out the configuration items involved.

5 Performance Analysis

To test the proposed solution, different types of software and hardware components are needed. The hardware we used was a desktop computer with the Ubuntu 18.04 operating system, an AMD 64 quad core, 2 GB memory, and a 40 GB hard disk. For software, we used Docker to ease the operation of the applications, including the open-source nginx proxy server and the Edgex-Foundry edge computing framework.

To assess the practical applicability of the designed system, we analyzed the performance of gateway in reverse proxy, load balancing and security authentication. We tested the performance of the gateway on an edge computing platform with 100 back-end microservices deployed, analyzed the round-trip time of non-authentication requests and authenticated requests, the maximum data throughput of the gateway and the maximum number of concurrent connections. Table 1 shows the experimental results of sending requests to microservices with and without the authentication plugins configured. Table 2 shows the data processing capabilities of the proposed solution.

Table 1. Round trip time test results.

Test Item	Average time
Round trip time for using the microservices API without any authentication	0.004s
Round trip time for using the microservices API with Basic authentication	0.0078s
Round trip time for using the microservices API with Key authentication	0.0065s
Round trip time for using the microservices API with JWT authentication	0.0189s
Round trip time for using the microservices API with OAuth 2.0	0.0205s
Round trip time for using the microservices API with LDAP authentication	0.0223s
Round trip time for using the microservices API with IP restriction	0.0111s

Reaction time is a measure of how quickly an organism responds to a stimulus. The statistical average reaction time is 284 ms. As can be seen in Table 1, the average RTT time to complete all kinds of certifications is less than 0.023 s, which is less than the reaction time. This indicates that the proposed system is applicable within the real world and performs well. In addition, the data processing performance of the scheme shown in Table 2 fully meets the parameter requirements of load balancing.

Table 2. Data processing capabilities test results.

Test Item	Value
Data handling capacity	20 Gbps
Maximum number of concurrent connections	10 million
New connections per second	0.4 million

6 Summary and Future Work

This article mainly studies the cloud-edge integrated security authentication method under the guidance of the concept of zero trust. It analyzes and summarizes the current cloud-edge-end integration process and the research status of related authentication technologies. Designed the cloud edge integrated platform security authentication system and industrial-grade API security access gateway. At the technical level, it has completed the construction of Nginx-based API security access gateway and related functions such as identity authentication, [13] load balancing, reverse proxy, log audit, analysis and monitoring. Dashboard is designed to facilitate user management. After testing on the edge computing platform of this project, the utility and security of the proposed scheme and design have been verified, which greatly meets the complex security certification requirements in the cloud-edge integrated computing application platform.

The entire cloud-edge collaboration process also needs to be fully identified to ensure that every request and distribution process must be authenticated and authorized. Next, the author will continue to promote the improvement of the cloud-edge integrated security certification system, mainly from the following aspects:

(1) Construct a unified identity authentication center. Perform comprehensive identity management on the platform, realize basic CA authentication, and single sign-on functions, and provide multi-factor authentication services based on the above functions.
(2) Design authority management system. The design and implementation supports a dynamic access control system based on RBAC + ABAC, centralized management of permissions, and realization of different granular permissions on demand, reducing the maintenance cost of the permission system.
(3) Build a comprehensive platform that supports various algorithms such as RSA and AES, key storage, and KMS activation capabilities. Realize the function of calling API based on Python or java to achieve encryption and decryption.

Acknowledgments. The ideas in this article come from discussions and research collaborations with two people: Zhigang Xiong and Jianjun Zeng. I would also like to thank the following people who provided valuable feedback on the design of this article and helped improve it: Quancheng Zhao, Lulu Zhao and the anonymous reviewers. This work was supported by the National Key Research and Development Program of China (grant number 2018YFC0831304) and the National Natural Science Foundation (Grant number 61772064).

References

1. Shi, W., Sun, H., Cao, J., et al.: Edge computing: a new computing model in the internet of everything era. Comput. Res. Dev. **54**(**5**), 907–924 (2017)
2. Satyanarayanan, M.: The emergence of edge computing. Computer **50**(1), 30–39 (2017)
3. Enqing, X., Enran, D.: Exploration and practice of collaborative development of cloud computing and edge computing. Commun. World **801**(09), 48–49 (2019)
4. Lu, X.: Research on task migration and resource management of mobile edge computing (2019)
5. Yong, S., Xiaofeng, L.: Cloud-edge integrated edge computing products help enterprises' digital transformation. Shanghai Inf. Technol. **10**, 59–61 (2018)
6. Zhang, J.: Overview of cloud computing platform security technology patents. Information and Computer: Theoretical Edition 000(011), pp. 126–129 (2015)
7. Yingnan, Z.: Zero-trust architecture: a new paradigm for network security. Financ. Electron. **11**, 50–51 (2018)
8. Zhang, Q.: SAB-IABS: a design of an anonymous two-way identity authentication system for interconnected clouds based on secure active bundles (2014)
9. Zhang, J.: Research on improved trusted network connection based on behavior analysis (2017)
10. Ni, J., Lin, X., Shen, X.: Efficient and secure service-oriented authentication supporting network slicing for 5G-enabled IoT. IEEE J. Sel. Areas Commun., 1 (2018)
11. Jiang, S.: Research and implementation of cloud integrated security solutions (2015)
12. Mukherjee, M., Matam, R., Shu, L., et al.: Security and privacy in fog computing: challenges. IEEE Access **5**, 19293–19304 (2017)
13. Pang, H.H., Tan, K.-L.: Authenticating query results in edge computing. In: Proceedings. 20th International Conference on Data Engineering. IEEE (2004)
14. Zhu, L., Li, M., Zhang, Z., et al.: Privacy-preserving authentication and data aggregation for fog-based smart grid. IEEE Commun. Mag. **57**, 80–85 (2019)
15. Botta, A., Donato, W.D., Persico, V., et al.: Integration of Cloud computing and Internet of Things: A survey. Future Gener. Comput. Syst. **56**(MAR), 684–700 (2016)
16. Gope, P.: LAAP: lightweight anonymous authentication protocol for D2D-Aided fog computing paradigm. Comput. Secur. **86,** 223–237 (2019)
17. Morabito, R., Petrolo, R., Loscrì, V., et al.: LEGIoT: a lightweight edge gateway for the Internet of Things. Future Gener. Comput. Syst. **81**, 1157–1171 (2017)
18. Wang, S., Zhang, X., Zhang, Y., et al.: A survey on mobile edge networks: convergence of computing, caching and communications. IEEE Access, **PP**(99), 1 (2017)
19. Guoqiang, S., Yanming, C., Chao, Z., et al.: Design and implementation of a smart IoT gateway. Green Computing & Communications. IEEE (2013)