Research on Identity Authenticated Encryption System for Application Services of the New Generation Dispatch Control System

Jing Wang1*, Ye Liang1, Yi Guan2, and Yong Liu2

1 NARI Group Corporation/State Grid Electric Power Research Institute, Beijing, 100000, China
2 State Grid Shandong Electric Power Company, 250000, Jinan, China
*Corresponding author’s e-mail: wangjing19891110@126.com

Abstract. This paper designs an identity authenticated encryption system for application services based on the problems of identity authentication and communication encryption of application service in the new generation dispatch control system, combined with the security protection requirements of dispatch control system. When the application is started, it will actively initiate authentication to the authentication center and obtain the application ID according to the national encryption algorithm after the authentication is passed. When the application makes a service invocation, the application directly sends the application ID to the service. The service verifies that the application ID is correct to provide the service without complicated bi-directional authentication and key agreement. At the same time, the session key for encryption in communications is automatically generated based on the information related to the application ID, which effectively reduces the impact of security authentication on service invocation performance while improving the security of service invocation.

1. Introduction
The new generation dispatch control system uses a lot of ICT technologies such as "Cloud Computing, Big Data, Internet of Things, Mobile Internet, Artificial Intelligence", and uses a unified support platform under the new architecture of "physical distribution and logical unity". The new generation dispatch control system is composed of a monitoring system deployed on the dispatch end, a model data center and an analysis and decision center deployed in the cloud. Its goal is to realize the vision of "full business information perception, full system collaborative control, full process online decision making, full time-space optimize balance, and full-scale load scheduling" to fully support the operation of a new generation of power system [1-3].

At the beginning of designing the new system, the concept of micro-service was introduced into the system design. Therefore, under the concept of service, most functions in the system are realized in the form of services. At the same time, due to the openness of the new generation dispatch control system, the service interfaces are all open to the public, which inevitably leads to potential security risks when the application invokes the service. Therefore, the identity of the application that invokes the service needs to be authenticated, and the data in the process of service call should be encrypted and protected.
2. Related Technology
The current authenticated encryption technologies for application services are mainly included as the following:

1. Based on single sign on and OAuth protocol, Zhang et al. [4] proposed a method of authentication between mutual trust application systems. This method is based on HTTPS and ticket authentication, and realizes mutual trust by verifying the tickets between applications.

2. Liu et al. [5] designed a service interface gateway system for registering and authenticating services with Redis, OAuth2 and other technologies. The system uses the token as a credential for authentication between different services, which realizes the unified export of services, authority control and monitoring.

3. Jin et al. [6] proposed a lightweight micro-service authentication method based on the OAuth technology, which authenticates each micro-service by issuing a token and uses a hybrid encryption mechanism to ensure the security of communications between services, it achieves high security between services while maintaining high communication efficiency.

4. Han et al. [7] proposed an API gateway system based on the Netty framework. The system builds a security filter chain that can certification and authenticate service invocation requests. The system can effectively meet the security requirements of various services on the enterprise service bus and support hundreds of millions of service security invocations per day.

5. Zhou et al. [8] analyzed the platform architecture of Openstack cloud platform, keystone of security authentication component, the main security issues of identity authentication in cloud computing and the foundation of current mainstream authentication technology in cloud environment, and proposed an improved method for the unified identity authentication mechanism and the vulnerabilities of the unified identity authentication technology under the cloud platform, which is based on OpenID technology and implemented the identity authentication technology on the Openstack platform.

In summary, the current service authentication in a service-oriented environment is based on http/https, using OAuth or OpenID and other web technologies to achieve security authentication of services, and using encryption algorithms such as RSA and AES for communication encryption. However, the http / https and open source software are prohibited in the production control area of the new generation dispatch control system.

For the current status of application service authentication and encryption in the new generation of dispatch control systems, this paper analyzes the current mainstream service authentication methods and combines the relevant security requirements of dispatch control systems to design an identity authenticated encryption system for application services. The system uses an independent application authentication center, based on TCP and a custom communication protocol, and uses the national encryption algorithm to perform bi-direction authentication for applications. The system generates an application ID based on the national encryption algorithm in service authentication for the authenticated application, and uses the secret key based on the application ID for encrypted communication, thereby realizing application service authentication and encryption in the new generation dispatch control system.

3. General design

3.1. System architecture
The identity authentication encryption system for application services consists of the following two parts:

1. Application authentication center: it mainly includes modules such as application information management, application authentication, application identification management, and certificate management. Among them, the application information management module is mainly used to manage the basic information of the application, including the registration, deletion, modification and other functions of the application; the application authentication module, mainly used to authenticate applications, including application authentication function; the application identification management module is mainly used to generate application identification after application authentication is passed, and is used for service authentication, including application identification generation, analysis, verification and other functions; the certificate management module is mainly used to generate relevant certificates for the application after successful registration, which is used for subsequent authentication, including the p12 file generation of the application certificate, public key certificate generation, certificate distribution and other functions.

2. Authenticated encryption module: it is deployed in various applications and services of the new generation dispatching control system in the form of a local interface library, and is mainly used for bi-direction authentication with an authentication center, application identification verification, and communication encryption and decryption between application services.

3.2. System process

Figure 1. Architecture of identity authenticated encryption system for application services

Figure 2. Identity authentication and encryption process for application services
The process of identity authenticated encryption for application service is as follows:

1. When the application starts, it actively initiates an authentication request to the authentication center for bi-direction authentication.
2. The authentication center certifies and generates an application ID for the application after passed the authentication.
3. When the application requests the service, it encrypts the original request message and forms a new request message together with the application identifier. Then send the new request message to the service.
4. After the service receives the request, it firstly verifies the application ID. If the verification is passed, the original request message is decrypted and the service is provided according to the request parameters, otherwise the service is rejected.

4. The Detailed System Design

4.1. Bi-direction authentication between application and authentication center

4.1.1. Bi-direction authentication process

When the application starts, it needs to perform bi-direction authentication with the authentication center, and the application ID can only be obtained after passing the authentication. This application ID is used to access the service, which indicates its identity. The main process is as follows:

As shown in Figure 3, the bi-direction authentication process between the application and the authentication center is:

1. When the application starts, the application uses the public key of the certification authority to encrypt the application name, manufacturer number and random number R, and sends it to the authentication center to request authentication.
2. The authentication center uses its own private key to decrypt the information, firstly query the database based on the application name and the vendor number to confirm whether the application is registered. If registered, a string M is generated, and R and M are encrypted using the application's public key and sent back to the application, otherwise the authentication failure is returned.
3. The application uses its own private key to decrypt the random numbers R' and M. If R’ and R are equal, the string M is encrypted under the public key of the authentication center and sent to the authentication center.
4. The authentication center uses its own private key to decrypt the string to obtain M'. If M' and M are equal, the bi-direction authentication is successful. The authentication center generates an application ID based on the application information and encrypts it with the application's public key to send back to application.

5. When the application receives the application ID, decrypt and save it.

4.1.2. Bi-direction authentication communication message
The communication message in the bi-direction authentication process between the application and authentication center adopts the JSON format, and its specific format is as follows:

1. The data structure of the authentication request sent by the application to the authentication center is as follows:

   ```
   {
   "type": message type,  // String
   "data": SM2(AppAuthInfo), // String
   "data_len": the length of SM2(AppAuthInfo) // Integer
   }
   ```

   The data structure of the authentication information AppAuthInfo is as follows:

   ```
   {
   "softname": software name,  // String
   "softfirm": manufacturer's number, // String
   "random": random number  // String
   }
   ```

2. The data structure returned by the authentication center to the random number is as follows:

   ```
   {
   "type": message type,  // String
   "data": SM2(RMInfo), // String
   "data_len": the length of SM2(RMInfo) // Integer
   }
   ```

   The data structure (JSON) of the random number information RMInfo is as follows:

   ```
   {
   "random": random number R,  // String
   "message": randomly generated string M, // String
   }
   ```

3. The data structure of the encrypted string M sent by the application to the authentication center is as follows:

   ```
   {
   "type": message type,  // String
   "data": SM2(M), // String
   "data_len": the length of SM2(M) // Integer
   }
   ```

4. The data structure of the application ID returned by the authentication center is as follows:

   ```
   {
   "type": message type,  // String
   "data": SM2(application ID), // String
   "data_len": the length of SM2(application ID) // Integer
   }
   ```

5. The data structure of the error code returned by the authentication center is as follows:

   ```
   {
   "type": message type, // String
   "ret_code": error code // Integer
   }
   ```
4.1.3. Design of application ID
The main function of the application ID is to indicate to the service that it is credible and has been authenticated by the authentication center when it requests the service.

The application ID consists of two parts: application ID basic information (AppIdInfo) and signature information (SignInfo).

The data structure of the basic information of application ID is as follows:

```json
{
    "authname": the name of the authentication center (issuer), // String
    "softname": application name(objects issued),                   // String
    "softfirm": manufacturer number,                                          // String
    "timestamp": timestamp (issuing time),                                 // String
    "overtime": expiration date                                                    // Integer
}
```

The basic information of application ID is expressed in JSON format and serialized into a JSON string.

The signature information is a character string obtained by signing the application ID basic information (AppIdInfo) using a national secret algorithm.

Finally, encode the basic information and signature information of the application ID with the Base64 algorithm, and then use '.' to combine them to form the application ID. The resulting application ID is a character string of the form " Base64(AppIdInfo).Base64(SignInfo)".

The application ID generation algorithm is as follows:
1. The authentication center generates AppIdInfo based on the application name and manufacturer number, and serializes it into a JSON format string;
2. Parse the P12 authenticate document of the authentication center to obtain the SM2 private key;
3. Use the SM2 private key to sign AppIdInfo to generate signature information SignInfo.
4. Encode TokenInfo and SignInfo using Base64 algorithm to get Base64 (TokenInfo) and Base64 (SignInfo);
5. Use '.' To connect them to generate a complete application ID, namely "Base64 (TokenInfo) .Base64 (SignInfo)".

The verification algorithm of the application ID is as follows:
1. Split the character string with '.' to get Base64 (AppIdInfo) and Base64 (SignInfo);
2. Use Base64 algorithm to decode AppIdInfo and SignInfo;
3. Deserialize the AppIdInfo expressed in JSON format to get the name of the authentication center, the generation time, and the expiration time;
4. Get the current time, and then compare the generation time according to the validity period to determine whether the application ID is invalid. If it fails, the verification fails, otherwise, go to the next step;
5. Get the public key certificate of the authentication center according to the name of the authentication center, and parse the public key certificate to obtain the public key;
6. Use the public key to verify the AppIdInfo and SignInfo. If the verification is not successful, it means failure.

4.2. Authentication and encryption between applications and services
When an application uses a service, the service must authenticate the application and the communication data of both parties must be encrypted. The specific process is as follows:
As shown in Figure 4, the authentication and encryption process between the application and the service is:

1. The application calculates a symmetric key K based on the time stamp in the application ID. When the application requests service, the symmetric key is used to encrypt the request message.

2. The application sends the application ID and the encrypted request message to the service together.

3. After receiving the request, the service separates the request message and the application ID, obtains the public key of the authentication center based on the name of the authentication center in the application ID. Then the service uses the public key to perform local verification on the application ID. If the verification is passed, the next step is executed, otherwise, the verification failure information is returned to the application.

4. The service calculates a symmetric key SK according to the time stamp in the application ID, and uses this key to decrypt the request message.

5. The service responds to the application based on the decrypt request message and generates a response message.

6. The service uses the key SK to encrypt the response message and send it back to the application.

7. The application uses the symmetric key K to decrypt the message, obtain the response message, and the service invocation is completed.

4.3. Identity authenticated encryption interface for application services

The identity authenticated encryption interface for application services is deployed in application and service, and is mainly used for application authentication, application ID verification, data encryption and decryption, etc. The main interfaces are as follows:

1. Bi-direction authentication interface: mainly used for bi-direction authentication between application and authentication center. The application encrypts the software name, manufacturer number and random number, and then sends it to the authentication center to initiate bi-direction authentication. If the authentication is successful, the application ID issued by the authentication center is returned, otherwise an error is returned.

2. Request message encryption interface: mainly used to securely encrypt request messages. The original request message is passed into the interface, and the interface returns a new request message. Firstly, the interface generates a symmetric encryption key according to the Token, encrypts the
request message into a cipher text, and then forms a new request message together with the Token and the cipher text. The structure of the new request message is: total length (4 bytes) + application ID length (4 bytes) + application ID (N bytes) + ciphertext length (4 bytes) + ciphertext (N bytes).

3. Request message decryption interface: mainly used to decrypt the request message and return the original request message. First, the interface parses the application ID and ciphertext according to the structure of the new request message, and then verifies the application ID. If the verification is passed, a symmetric encryption key is generated according to the application ID, and then the ciphertext is decrypted to obtain the original request message, so that the service can provide the corresponding service according to the request parameters.

4. Symmetric encryption interface: mainly used for symmetric encryption of plaintext, and then return the encrypted ciphertext.

5. Symmetric decryption interface: mainly used for symmetric decryption of ciphertext, and then returns the decrypted plaintext.

5. System verification
In order to verify the research content of this paper, a prototype system was implemented. The system is composed of three parts: application authentication center, application and service.

The verification steps of the authentication function of the application service are as follows:
1. When the application starts, without authentication, directly request the service and check whether the service can be called normally.
2. When the application starts, call the bi-direction authentication interface, and then request the service to check whether the service can be called normally.

The prototype system is used for verification. The verification result is: In step 1, the application cannot call the service normally, and receives an error message that the authentication fails and the service is not provided. In step 2 the service can be called normally.

The verification steps of the encryption function of the application service are as follows:
1. communication without encryption: when the application is started, authentication is started, the encrypted communication interface is not called, request messages of different lengths are sent to the service, and finally counts the time spent on service invocation.
2. Encrypted communication with the method of this paper: when the application starts, it starts authentication, then calls the encrypted communication interface, sends request messages of different lengths to the service, and finally counts the time spent on service invocation.
3. Encrypted communication with traditional methods: For each time when the application makes a service invocation, it firstly performs key agreement with the service, then uses the exchanged key to encrypt the request message, and then sends request messages of different lengths to the service, and finally statistics the time spent on service invocation.

The statistical results are shown in the following table 1:

| Request message size | Time cost (communication without encryption) | Time cost (encrypted communication) | Time cost (encrypted communication of the key agreement each time) |
|----------------------|---------------------------------------------|-------------------------------------|---------------------------------------------------------------|
| 1KB                  | 5ms                                         | 8ms                                 | 245ms                                                         |
| 512KB                | 30ms                                        | 52ms                                | 283ms                                                         |
| 1MB                  | 36ms                                        | 61ms                                | 311ms                                                         |

6. Conclusion
To solve the problems of identity authentication and communication encryption for application services in the new generation dispatch control system, in this paper, it designs an identity authenticated encryption system for application services according to the security protection requirements of the dispatch control system, based on the national encryption algorithm and
autonomous communication protocol. The system uses a unified application authentication center to authenticate the applications. After passed the authentication, it generates an application ID for service authentication. The application ID is designed based on the national encryption algorithm, and has the characteristics of global uniqueness, anti-tampering, self-signing and local verification. When an application makes a service invocation, it does not need to perform complicated bi-direction authentication and key agreement with the service, but directly sends the application ID to the service. If the service verifies that the application ID is correct, the service can be provided. At the same time, the secret keys for both communication parties are automatically generated based on the relevant information of the application ID, which effectively reduces the impact of security authentication on the performance of service invocation while improving the security of service invocation.

Acknowledgments
This paper is supported by the State Grid Corporation of China Science and technology project "Research on lightweight active immune technology for power monitoring system”.

References
[1] Feng Shuhai, Yao Jianguo, Yang Shengchun, et al. (2015) Overall design of integrated analysis centre for physically-distributed and logically-integrated dispatch system. J. Electric Power Automation Equipment, 35(12): 138-144.
[2] Xu Hongqiang, Yao Jianguo, Yu Yijun, Tang Biqiang. (2018) Architecture and key technologies of dispatch and control system supporting integrated bulk power grids. J. Electric Power Automation Equipment, 42(6): 1-8.
[3] Guo Jiancheng, Nan Guilin, Xu Dan, Zhuang Weijin, Shan Xin, Zhao Lin, Yang Shengchun. (2018) Connotation and key technology of global monitoring for large powergrid . J. Electric Power Automation Equipment, 42(18): 1-7+16.
[4] Zhang Qi. (2016) Identity authentication method between application systems of mutual trust. Central China Normal University.
[5] Liu Hongbo, Xiang Qiquan, Tang Shulin, Wang Xiaoyong. (2019) Design and Research on API Gateway Based on GIS Service [J]. Geospatial Information, 17(01):65-67+11.
[6] Jin Yike. (2018) Research on Data Transmission and Authentication Security of Microservice Architecture. Hangzhou Dianzi University.
[7] Han Xue. (2018) The design and implementation of a API gateway based on enterprise service bus. Southeast University.
[8] Zhou Changchun, Tian Xiaoli, Zhang Ning, Yang Yunjun, Li Duo. (2016) Research on identity authentication technology in cloud computing. J. Computer Science, 43(S1):339-341+369.