ICS 35.030
CCS L 80

NATIONAL STANDARD OF THE PEOPLE'S REPUBLIC OF CHINA

GB/T 47475-2026

Cybersecurity technology - Open resource authorization protocol for third parties

网络安全技术 开放的第三方资源授权协议

Issued on: April 30, 2026 **Implemented on: November 1, 2026**

Issued by: State Administration for Market Regulation;
Standardization Administration of the PRC

Table of Contents

5 General Rules
5.3 Protocol Endpoint Types
6 Client Types and Requirements
6.4 Identity Verification
7 Authorization Process
7.1 License Type
7.2 Authorization Code Licensing
7.3 Client Identity Credentials Permission
7.4 Equipment Licensing
7.4.3 Authorization Response
8 Token Issuance and Refresh
8.1 Token Types

5 General Rules

5.1 Protocol Roles The roles in the agreement are as follows:

---Resource owner. Responsible for deciding whether to grant access to protected resources and providing identity credentials during the authentication process.

---Service Provider. The platform or service to which the resource server or authorization server belongs. The authorization server is responsible for authorizing resource owners. Identity authentication verifies the legitimacy and authenticity of the client and determines whether to issue an authorization license based on the resource owner's authorization intention. Access token. The resource server is responsible for hosting the protected resources, receiving and verifying the access token submitted by the client, and processing the token based on the verification result. If it responds to the resource request.

---Client (Third-party application). Responsible for initiating authorization requests to the resource owner and using tokens on behalf of the resource owner to access the protected [resource/resource]. Protect resources.

5.2 Basic Protocol Flow The basic process of the protocol is shown in Figure 1, which illustrates the relationship between the three roles. resource owner, service provider, and client.

a) The client requests authorization from the resource owner. The client does not interact directly with the resource owner, but instead acts through an authorization server. The

intermediary sends a request to the resource owner [see

5.3 Protocol Endpoint Types

5.3.1 Endpoint Classification This document specifies three functional endpoints.

a) Authorized endpoint;

b) Token endpoint;

c) Redirect endpoints. Authorization endpoints and token endpoints allow clients to add parameters to their requests that indicate the scope of access, specifying the resource access request. The request specifies the scope of protected resource access. Accordingly, the authorization server informs the client by including the access scope parameter in the response. The response specifies the scope of protected resource access granted by the issued access token. If the actual granted access scope differs from the requested scope, the response will provide... A parameter that indicates the actual scope of access, informing the client of the scope of protected resources that are actually allowed to be accessed. If the client omits the scope parameter when requesting authorization, the authorization server responds to the request with a predefined default value, or rejects it. This request. The authorization server describes the requirements and default values for the range parameters in its service documentation.

5.3.2 Authorized Endpoints The authorization endpoint resides on the authorization server, receiving authorization requests from clients, the resource owner's identity credentials and authorization, and authorizing the process via user proxy. The client is redirected to the redirect endpoint along with an authorization code. When the authorization server receives the client's authorization request, the authorization server first... First, verify the identity of the resource owner. The authorization endpoint URI is typically provided by the authorization server's service documentation. The URI of the authorized endpoint may contain a query component. When additional query parameters are added, this query component is retained, and the endpoint URI does not contain it. Fragment component. The authorization endpoint of the authorization server should be capable of handling HTTP GET requests and should also be able to handle HTTP POST requests. ability.

5.3.3 Token Endpoint The client submits an authorization or refresh token to the authorization server's token endpoint. After verifying the request, the authorization server issues the token via the token endpoint. Give the access token to the client. This document does not specify how the client obtains the token endpoint URI (this is usually provided by the authorization server's service documentation). The token endpoint URI may contain a query component. This query component is retained when additional query parameters are added. The token endpoint URI does not contain... Includes fragment components. The client uses the HTTPPOST method when requesting an access token from the token endpoint of the authorization server. When the authorization server's token endpoint

receives a request from a client with credential protection or other clients that have been granted identity credentials... When a request is made, the authorization server authenticates the client's identity.

5.3.4 Redirecting Endpoints During its registration phase, the client pre-registers the URI with the authorization server as a redirect endpoint. The authorization server then completes the process with the resource owner. After the interaction with the user, the resource owner's user agent is redirected to the client's redirection endpoint, and the authorization result is returned to the client. The redirect endpoint URI is an absolute URI. The redirect endpoint URI can contain query components; additional information can be added to the redirect endpoint URI. When querying parameters, retain this component. Redirect endpoint URIs do not contain fragment components. The security requirements for redirected endpoints should comply with B.2.

6 Client Types and Requirements

6.1 Type This document specifies two types of clients, depending on whether they have the ability to protect their identity credentials.

a) Clients with credential protection capabilities The client has the ability to maintain the confidentiality of its credentials (e.g., the client runs on a secure server that strictly enforces access control). (above), thus proving the authenticity of one's identity by providing secure credentials, or the client having the ability to do so through other means. Prove your identity in ways (beyond the scope of this document);

Note. A typical example is a web application, where the resource owner accesses the application through an HTML user interface, which is defined by the user. The user agent in the device is used for rendering. Client identity credentials and all access tokens issued to the client are stored on the web server. This is unavailable and inaccessible to the resource owner.

b) Clients that lack credential protection capabilities The client lacks the ability to maintain the confidentiality of its credentials (e.g., the client runs on a device used by the resource owner, and the local application should be confidential). (Using browser-based applications, etc.), they cannot provide secure identity credentials to prove their identity authenticity, and therefore cannot... Forces seek other ways to prove the authenticity of their identity.

Note. Typical examples include applications running on user agents and local applications. Applications running on user agents are downloaded from the web server to the resource. It is executed on the local device and in a user agent (such as a web browser), and its protocol data and credentials are accessible to the resource owner. The application is installed and runs on the device used by the resource owner, and its protocol data and credentials are accessible to the resource owner. The determination of client type depends on the authentication security requirements of the authorization server and the risk of the

authorization server's client identity credentials being obtained. The level of acceptance is typically provided in the authorization server's service documentation. The authorization server does not make any assumptions about the type of client. The client can be implemented by a set of distributed components, each with a different client type and security context (e.g., client type, security context... The client has both server-based components with credential protection and browser-based components without credential protection. Registration of such clients is not specified in this document; typically, client operators can register each component of the client with the licensing service. On the device.

6.2 Identifiers A client identifier is an application identifier issued by the authorization server to a registered client. This identifier is a string, and the authorization server... The server uses this string to uniquely identify a client. This document does not specify the length of the client identifier. Authorization servers should describe the client identifiers they issue in their service documentation. The length.

6.3 Registration Before the agreement is finalized, the client's provider needs to register the client's information (e.g., redirect endpoint URI) on the authorization server. (Client type, etc.) establishes a trust relationship between the client and the authorization server. The client provider uses the registration method permitted by the authorization server. (Usually provided by the authorization server's service documentation) Complete registration. When registering a client, the client provider should provide the following information to the authorization server.

b) A redirect endpoint pointing to the client;

c) Other information required by the authorization server (e.g., application name, website and description, client authentication method, etc.).

6.4 Identity Verification

6.4.1 Identity Authentication Scheme For clients with credential protection capabilities, when the authorization server uses an identity credential-based authentication scheme to authenticate the client... At that time, the authorization server should allow the client to add relevant parameters to the request body to pass the client's identity credentials for authentication. The client uses the HTTP Basic Authentication scheme (see RFC 9110). When using an authentication scheme that includes client credentials in the request body... When transmitting parameters, the following parameters should be placed in the HTTP body using the POST method, and should not be transmitted using the HTTHGET method. Included in the requested URI. The client identifier described in 6.2.

b) client_secret [Required] Client key.

c) nonce [Required condition] A random string generated by the client. This parameter is required when client_secre uses the method described in