Hash Signing developer documentation

Introduction

Cleverbase offers hash signing through a HTTP interface that complies with the Cloud Signature Consortium (CSC) Standard.

Table of Contents

Hosts

Environment	Host
Pre-production	https://connect.acc.cleverbase.com
Production	https://connect.cleverbase.com

Client registration

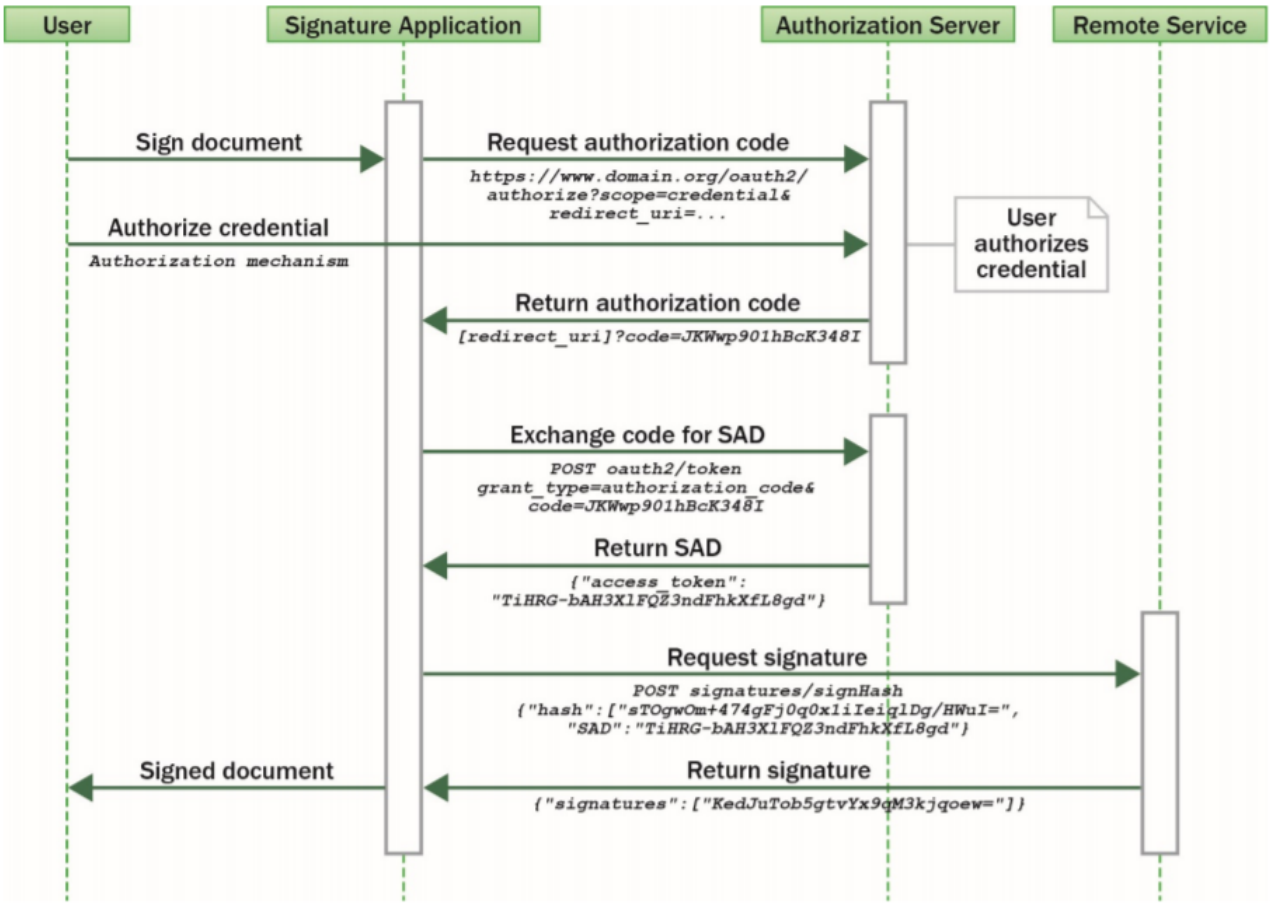
Client registration is a manual process performed in collaboration with an account manager of Cleverbase. As input, the redirection URI(s) and client name (for displaying purposes) of the client is required. Client registration will lead to a confidential OAuth 2.0 Client with two parameters:

Parameter	Classification
OAuth 2.0 Client Identifier	Public Information
OAuth 2.0 Client Secret	Private information

This OAuth 2.0 Client can subsequently act as a client to the hash signing service of Cleverbase.

Flow

13.5 Create a remote signature with a credential protected by OAuth2 with Authorization Code flow



Requests

The connection support suite currently offers five endpoints to test integration of your application with Cleverbase hash signing.

OAuth 2.0 Authorization endpoints

Cleverbase's CSC hash signing application uses OAuth 2.0 for authorization.

GET /oauth2/authorize

- RFC 6749
- CSC

Description

Starts the OAuth 2.0 authorization server using an Authorization Code flow, as described in Section 1.3.1 of RFC 6749, to request authorization for the user to access the remote service resources. The authorization is returned in the form of an authorization code, which the signature application SHALL then use to obtain an access token with the oauth2/token method. The authorization server SHOULD support two access token

scopes: “service” and “credential”. These scopes SHALL only be used separately to obtain an access token suitable for service and credential authorization respectively.

Input parameters

Parameter	Presence	Value	Description
response_type	REQUIRED	String	Must always be "code"
client_id	REQUIRED	String	unique Client Identifier (see Client Registration)
redirect_uri	REQUIRED	String	A registered redirect URL to redirect to after the authorization process
scope	REQUIRED	String	"service" or "credential"
state	REQUIRED	String	Up to 255 bytes of arbitrary data from the signature application that will be passed back to the redirect URI.
lang	OPTIONAL	String	Language
credentialID	REQUIRED conditional	String	Required if scope is "credential"
numSignatures	REQUIRED conditional	Int	Required if scope is "credential" Must be 1.
hash	REQUIRED conditional	String	Required if scope is "credential". The to-be-signed hash value base64 url encoded.

Response

Parameter	Presence	Value	Description
code	REQUIRED	String	The authorization code generated by the authorization server
state	REQUIRED	String	Should match state given in request
error	OPTIONAL	String	A single error code string
error_description	OPTIONAL	String	Human-readable text providing additional error information

Examples

Service Scope Request

```
GET /oauth2/authorize?
response_type=code&
client_id=<OAuth2_client_id>&
redirect_uri=<OAuth2_redirect_uri>&
scope=service&
```

```
lang=en-US&
state=12345678
```

Service Scope Response

```
HTTP/1.1 302 Found
Location: <OAuth2_redirect_uri>?
code=FhkXf9P269L8g&
state=12345678
```

Credential Scope Request

```
GET /oauth2/authorize?
response_type=code&
client_id=<OAuth2_client_id>&
redirect_uri=<OAuth2_redirect_uri>&
scope=credential&
credentialID=GX0112348&
numSignatures=1&
hash=MTIzNDU2Nzg5MHF3ZXJ0enVpb3Bhc2RmZ2hqa2zDtnl4
&state=12345678
```

Credential Scope Response

```
HTTP/1.1 302 Found
Location: <OAuth2_redirect_uri>?code=HS9naJKWwp901hBcK348IUHiuH8374&
state=12345678
```

Error Response

```
HTTP/1.1 302 Found
Location: <OAuth2_redirect_uri>?error=invalid_request&
error_description=Invalid%20Authorization%20Code&state=12345678
```

POST /oauth2/token

- RFC 6749
- CSC

Description

Obtain an OAuth 2.0 bearer access token from the authorization server by passing either the client credentials pre-assigned by the authorization server to the signature application, or the authorization code returned by the authorization server after a successful user authentication, along with the client ID and client secret in possession of the signature application.

Input parameters

Parameter	Presence	Value	Description
grant_type	REQUIRED	String	Must always be "authorization_code"
code	REQUIRED	String	Code obtained from result oauth2/authorize
client_id	REQUIRED	String	Unique Client Identifier (see Client Registration)
redirect_uri	REQUIRED	String	A registered redirect URL where the user was redirected after the authorization process completed.

Input headers

Parameter	Presence	Value	Description
Authorization	REQUIRED	String	Must always be of type Basic Auth. e.g. Basic Y2xpZW50SUQ6cGFzc3dvcmQ (RFC 6749)

Response

Parameter	Presence	Value	Description
access_token	REQUIRED	String	The short-lived access token to be used depending on the scope of the OAuth 2.0 authorization request. When the scope is “service” then the authorization server returns a bearer token to be used as the value of the “Authorization: Bearer” in the HTTP header of the subsequent API requests within the same session. When the scope is “credential” then the authorization server returns a Signature Activation Data token (SAD) to authorize the signature request. This value SHOULD be used as the value for the SAD parameter when invoking the signatures/signHash method."
token_type	REQUIRED	String	Must be Bearer or SAD
expires_in	OPTIONAL	Int	The lifetime in seconds of the service access token.

Examples

Sample Request (Authorization code flow)

```
POST oauth2/token HTTP/1.1
Authorization: Basic Y2xpZW50SUQ6cGFzc3dvcmQ
Content-Type: application/x-www-form-urlencoded
grant_type=authorization_code&
code=FhkXf9P269L8g&
client_id=<OAuth2_client_id>&
redirect_uri=<OAuth2_redirect_uri>
```

Sample Response (for service scope)

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
{
 "access_token": "4/CKN69L8gdSYp5_pwH3XlFQZ3ndFhkXf9P2_TiHRG-bA",
 "token_type": "Bearer",
 "expires_in": 3600
}
```

Sample Response (for credential scope)

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
{
 "access_token": "3XlFQZ3ndFhkXf9P24/CKN69L8gdSYp5H3XlFQZ3ndFhkXf9P2",
 "token_type": "SAD",
 "expires_in": 300
}
```

Credential and Signing endpoints

For the whole CSC specification, see here

POST /csc/v1/credentials/list

See CSC Chapter 11.4

Description

Returns the list of credentials associated with a user identifier.

Input Parameters

Parameter	Presence	Value	Description

Parameter	Presence	Value	Description
maxResults	OPTIONAL	String	Maximum amount of credentialIDs returned
pageToken	OPTIONAL	String	An opaque token to retrieve a new page of results
clientData	OPTIONAL	String	CSC 7.5

Input headers

Parameter	Presence	Value	Description
Authorization	REQUIRED	String	Must always be of type Bearer. Obtained from the "service" scope OAuth flow. e.g. Bearer 4/CKN69L8gdSYp5_pwH3XlFQZ3ndFhkXf9P2_TiHRG-bA

Response

Parameter	Presence	Value	Description
credentialIDs	REQUIRED	Array of String	Credential Identifiers

Sample request

```
POST /csc/v1/credentials/list HTTP/1.1
Authorization: Bearer 4/CKN69L8gdSYp5_pwH3XlFQZ3ndFhkXf9P2_TiHRG-bA
Content-Type: application/json
{
 "maxResults": 10
}
```

Sample response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
{
 "credentialIDs": [ "GX0112348", "HX0224685" ]
}
```

POST /csc/v1/credentials/info

See CSC Chapter 11.5

Description

Returns the main identity information and public key certificate of the credential obtained with the `/csc/v1/credentials/list` endpoint.

Input Parameters

Parameter	Presence	Value	Description
credentialID	REQUIRED	String	The unique identifier associated to the credential

Input headers

Parameter	Presence	Value	Description
Authorization	REQUIRED	String	Must always be of type Bearer. Obtained from the "service" scope OAuth flow. e.g. Bearer 4/CKN69L8gdSYp5_pwH3XlFQZ3ndFhkXf9P2_TiHRG-bA

Response

Parameter	Presence	Value	Description
key/status	REQUIRED	String	The status of the signing key of the credential: “enabled”: the signing key is enabled and can be used for signing. “disabled”: the signing key is disabled and cannot be used for signing. This MAY occur when the owner has disabled it or when the RSSP has detected that the associated certificate is expired or revoked.
key/algo	REQUIRED	Array of String	The list of OIDs of the supported key algorithms
key/len	REQUIRED	String	The length of the cryptographic key in bits
cert/certificates	REQUIRED	Array of String	One or more Base64-encoded X.509v3 certificates from the certificate chain.
authMode	REQUIRED	String	Specifies one of the authorization modes. Must be "oauth2code"

Sample request

```
POST /csc/v1/credentials/info HTTP/1.1
Authorization: Bearer 4/CKN69L8gdSYp5_pwH3XlFQZ3ndFhkXf9P2_TiHRG-bA
Content-Type: application/json
{
 "credentialID": "GX0112348"
}
```

Sample response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
{
  "key" : {
    "status" : "enabled",
    "algo" : [
      "1.2.840.113549.1.1.1"
    ],
    "len" : 2048
  },
  "cert" : {
    "certificates" : [
      "MIIGDSGDGSD...."
    ]
  },
  "authMode" : "oauth2code"
}
```

POST /csc/v1/signatures/signHash

See CSC Chapter 11.9

Description

Calculate the remote digital signature of one or multiple hash values provided in input. This method requires credential authorization in the form of Signature Activation Data (SAD).

Input Parameters

Parameter	Presence	Value	Description
credentialIDs	REQUIRED	String	Credential Identifiers
SAD	REQUIRED	String	Signature activation data (result from "credential" scope flow
hash	REQUIRED	Array of String	The to be signed data representation. Must be a SHA256 base64 encoded hash.
hashAlgo	REQUIRED	String	Hash algorithm identifier. Must be "2.16.840.1.101.3.4.2.1"
signAlgo	REQUIRED	String	Signing algorithm identifier. Must be "1.2.840.113549.1.1.1"
clientData	Optional	String	CSC 7.5

Input headers

Parameter	Presence	Value	Description
Authorization	REQUIRED	String	Must always be of type Bearer. Obtained from the "service" scope OAuth flow. e.g. Bearer 4/CKN69L8gdSYp5_pwH3XlFQZ3ndFhkXf9P2_TiHRG-bA

Response

Parameter	Presence	Value	Description
Signature	REQUIRED	Array of String	Credential Identifiers

Examples

Sample request

```
POST /csc/v1/signatures/signHash HTTP/1.1
Content-Type: application/json
Authorization: Bearer 4/CKN69L8gdSYp5_pwH3XlFQZ3ndFhkXf9P2_TiHRG-bA
{
 "credentialID": "GX0112348",
 "SAD": "3XlFQZ3ndFhkXf9P24/CKN69L8gdSYp5H3XlFQZ3ndFhkXf9P2",
 "hash":
 [
 "sTOgwOm+474gFj0q0x1iSNspKqbcse4IeiqlDg/HWuI="
 ],
 "hashAlgo": "2.16.840.1.101.3.4.2.1",
 "signAlgo": "1.2.840.113549.1.1.1"
}
```

Sample response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
{
 "signatures":
 [
 "KedJuTob5gtvYx9qM3k3gm7kbLBwVbEQRl26S2tmXjqNND7MRGtoew=="
 ]
}
```