归档存储

认证与鉴权

产品文档

腾讯云

文档目录

认证与鉴权
签名算法

最近更新时间：2018-05-25 16:33:11

签名描述

使用归档存储服务 CAS 时，发起的每一次 HTTP 的请求都要求为**签名请求**。签名请求根据腾讯云提供的密钥，并结合请求的内容计算出一串密文，通过在 HTTP Headers 中的 Authorization 字段的方式传入。

归档存储可以根据用户指定的访问控制策略，与传入的签名进行判断后，决定应当许可何种请求、如何控制请求内容、控制请求有效期等。

限制请求并使用签名访问，常见于以下场景：

> 校验使用者身份：计算签名的过程依赖腾讯云提供的密钥，包含了 SecretID 和 SecretKey 等元素。
>
> 校验传输的数据：数据的校验信息包含在签名的密文信息中，若传入数据的校验值与密文中的校验值不匹配，则请求将返回失败。通常这能保障请求内容被劫持时，错误的数据内容不会被记录。
>
> 保护签名不被二次使用：通过在签名中加密请求的有效时间，可以确保在签名过期后客户端无法发起请求，这也用于确保网络被监听时，第三方无法介入重复使用签名以破坏数据安全性。

签名算法

概述

归档存储的每一次请求都要求在 HTTP 请求头部中传入 Authorization 字段，这也是 HTTP 标准定义中最常见的请求验证方式。常见的请求如下：

```
PUT /-/vaults/example HTTP/1.1
Host: cas.ap-chengdu.myqcloud.com
Authorization: q-sign-algorithm=sha1&q-ak=QmFzZTY0IGlzIGEgZ2VuZXJp&q-sign-time=1480932292;1
```

以下表格中，列出了 Authorization 字段需要传入的信息：

名称	描述
q-sign-algorithm	描述该签名使用的加密方式，目前腾讯云使用的是 HMAC-SHA1 的方式加密签名。 该字段请保持默认值：sha1

名称	描述
q-ak	用于标识用户身份， SecretID 的字段，在腾讯云的 API 密钥页面中可以查看。
q-sign-time	签名的有效起止时间，其使用 10 位 Unix 时间戳来表示，有效效力精确到秒。 该字段通过分号区分起止，起时在前、止时在后。
q-key-time	用户可以自定义 SignKey 有效时间，使用 10 位 Unix 时间戳来表示，有效效力精确到秒。 该字段通过分号区分起止，起始时间在前、终止时间在后。 一般 q-key-time 的时间范围大于等于 q-sign-time。
q-header-list	提供密文中包含需要校验的 Headers 列表，必须是 `小写字符` ，且需要 `按字典序排序` ，以";"分隔
q-url-param-list	提供密文中包含需要校验的 Parameters 列表，必须是 `小写字符` ，以";"分隔
q-signature	经过 HMAC-SHA1 算法加密的请求校验信息。

计算签名

在构造签名前，必须先获取腾讯云账号的 API 密钥对，其中包含 SecretID 和 SecretKey，详情参看腾讯云控制台中的「监控与管理 - 云 API 密钥」。

签名构成

- SignKey：携带有效时间，并通过 SecretKey 进行 HMAC-SHA1 加密的密钥串。
- FormatString：将请求经过一定规范格式化后的字串。
- StringToSign：包含校验算法、请求有效时间和 Hash 校验后的 FormatString 的字符串。
- Signature：加密后的签名，使用 SignKey 与 StringToSign 通过 HMAC-SHA1 加密的字符串，填入 q-signature。

1. 计算 SignKey

为了保障 SecretKey 的安全，请求需要对 SecretKey 进行加密后传输，签名允许用户通过携带 Unix 时间戳的方式限制 SecretKey 的有效使用时间。

为了让用户可以将计算签名所需的密钥信息下发到不信任的客户端，SecretKey 支持与用户指定的有效时间一起，一并加密生成一个不可逆的密钥串，下发给不被信任的客户端使用。计算 SignKey 的格式如下：

$SignKey =

```
HMAC-SHA1($SecretKey,"<q-key-time>")
```

- SecretKey：来自腾讯云提供的 API 密钥对中的 SecretKey，例如 `AKIDZfbOA78asKUYBcXFrJD0a1ICvR98JM` 。
- q-key-time：包含 SignKey 的起始和终止有效时间，前后两个时间戳用分号分开，使用 10 位 Unix 时间戳，精确到秒。例如 `1480932292;1481012292` 。

2. 构成 FormatString

该字符串将 HTTP 请求中的关键信息进行格式化处理，并将用于作为加密签名校验的主要部分。这可以保障 HTTP 请求在被传输的过程中，信息不会被第三方篡改。

为了让归档存储的服务端可以按照固定格式来校验请求，HTTP 请求中的关键数据都需要包含在 FormatString 中，采用换行的方法陈列数据，每行一个关键要素。生成的方法如下：

$FormatString =

```
<FormatMethod>\n
<FormatURI>\n
<FormatParameters>\n
<FormatHeaders>\n
```

- FormatMethod：指该请求的 HTTP 操作行为，例如 PUT/GET/DELETE，必须转为 `小写字符` 。
- 例如发起 `Get http://cas.ap-chengdu.myqcloud.com` 其 FormatMethod 为 `get`
- FormatURI：指该请求中的 URI 部分，即除去 http:// 协议和域名的部分（通常以 / 开始），并且不包含 URL 中的参数部分（通常以？开始）。
- 例如访问地址 `http://cas.ap-chengdu.myqcloud.com/-/vaults` 其 Format URI 为 `/-/vaults`
- FormatParameters：指该请求中的参数部分（以？开始的部分），用 `key=value` 的方式表达。参数的 key 和 value 都必须经过 URL Encode，如果有多个参数对可使用 & 连接，**Key和Value必须转为小写字符，且key值按字典序排序。**
  - 例如访问 `http://cas.ap-chengdu.myqcloud.com/-/vaults?limit=2` 其 FormatParameters 为 `limit=2`
- FormatHeaders：指请求中的 HTTP 头部信息，用 `key=value` 的方式表达。头部的 key 必须全部小写，value 必须经过 URL Encode。如果有多个参数对可使用 & 连接。**key值按字典序排序**
  - 例如头部 `Host: cas.ap-chengdu.myqcloud.com` 其 FormatHeaders 为 `host=cas.ap-chengdu.myqcloud.com`

3. 计算 StringToSign

该字符串包含了签名的算法名称，签名的有效时间，和 SHA-1 哈希后的 FormatString，因此需要计算 StringToSign 必须先生成好 FormatString。

StringToSign 中使用到的签名有效时间与 SignKey 中的 SecretKey 有效使用时间概念不同，这里的有效时间仅用于校验请求是否在有效的时间内被发起。

- 如果发起请求的客户端是可信的，一般可以直接将 SecretKey 保存在客户端，并且可以在 SignKey 和 StringToSign 中使用相同的有效起止时间来保障请求的有效性。
- 如果发起请求的客户端默认不可信，则需要保护 SecretKey 不能直接将其保存在客户端，此时便可以通过对 SecretKey 进行起止时间限制和加密后，下发 SignKey 给客户端以用于计算签名，而 StringToSign 中的有效时间应当在请求发起时由客户端生成。

生成 StringToSign 的格式如下：
$StringToSign =

```
<q-sign-algorithm>\n
<q-sign-time>\n
SHA1Hash($FormatString)\n
```

- q-sign-algorithm：签名使用的加密算法，默认填 `sha1` 。
- q-sign-time：该请求的起始和终止有效时间，前后两个时间戳用分号分开，使用 10 位 Unix 时间戳，精确到秒。例如 `1480932292;1481012292` 。
- SHA1Hash($FormatString)：将构成 FormatString 部分的字符串，使用 SHA-1 算法哈希得到一个不可逆的字符串，以用于标识请求的关键内容。

4. 计算签名 Signature

此步生成的 Signature 将被放置在 q-signature 字段，用于校验请求的内容是否合法。其使用到 HMAC-SHA1 算法，使用 SignKey 作为密钥对 StringToSign 进行加密计算。生成 Signature 的格式如下：

$Signature =

```
HMAC-SHA1($SignKey,$StringToSign)
```

5. 生成 Authorization

将计算签名步骤生成的内容与请求中需要明文标识的内容，用 `key=value` 的方式表达，多个参数对使用 & 连接。

生成 Authorization 的格式如下（一个长字符串，不含换行符）：

```
q-sign-algorithm=sha1&
q-ak=<SecretID>&
q-sign-time=<SignTime>&
q-key-time=<KeyTime>&
q-header-list=<SignedHeaderList>&
q-url-param-list=<SignedParameterList>&
q-signature=<Signature>
```

权限管理

最近更新时间：2018-05-25 16:46:07

详情参见产品文档『账号相关』-『访问控制』-『用户指南』-『策略语法』，单击此处。