A-Trust Gesellschaft für Sicherheitssysteme
im elektronischen Datenverkehr GmbH
Landstraßer Hauptstraße 1b
The Mall E02
A-1030 Wien

https://www.a-trust.at
E-Mail: office@a-trust.at

A-Trust Timestamp Developer Manual

Version: 1.0
Date: July 26, 2023

Contents

Date	Rev	Autor	Changes
26.07.2023	1.0	Patrick Hagelkruys	new Design
05.01.2021	0.3	Patrick Hagelkruys	update chapter 2.7.1
29.12.2020	0.2	Patrick Hagelkruys	add example timestamp query
29.12.2020	0.1	Patrick Hagelkruys	first version

Table 1: document history

1 Overview

This document describes the timestamping solution, interfaces and possible additional parameters. The service is implemented in accordance with RFC 3161 [ACPZ01] and available via an HTTP interface.

2 Timestamping Interface

The timestamping service is available via an HTTP interface.

2.1 Request: HTTP POST

Add the timestamping request as binary data as body to an HTTP POST request, including '`application/timestamp-query`' as content-type header parameter. A sample request is shown in listing 1.

```
POST / HTTP/1.1
Content-Type: application/timestamp-query
Host: ...
Content-Length: 59

{binary request data}
```

Listing 1: timestamping request HTTP POST

2.2 Request: HTTP GET

Add the base64 or base64url encoded timestamping request as part of the URL path. A sample request is shown in listing listing 2. Furthermore, important is the correct content-type of '`application/timestamp-query`', otherwise, the request is not recognized as a timestamping request.

```
GET /.../{based64url request} HTTP/1.1
Content-Type: application/timestamp-query
Host: ...
```

Listing 2: timestamping request HTTP GET

2.3 Request: HTTP GET - query parameter

Add the timestamping request as a query parameter with one of the keys '`timestamp`', '`ts`' or '`query`'.

```
GET /.../timestamp={based64url request} HTTP/1.1
Content-Type: application/timestamp-query
Host: ...
```

Listing 3: timestamping request HTTP GET with query parameter

2.4 Response

The response for all request types is as shown in listing 4.

```
HTTP/1.1 200 OK
Content-Length: 2406
Content-Type: application/timestamp-reply


{binary response data}
```

Listing 4: timestamping response

2.5 Authentication

The timestamping service requires authentication, this can be done via HTTP Basic Authentication. Therefore add a HTTP header '`Authorization`' to the request, as shown in listing 5.

```
POST / HTTP/1.1
Content-Type: application/timestamp-query
Host: ...
Content-Length: 59
Authorization: Basic dGVzdEhzbVJTQTp0ZXN0


{binary request data}
```

Listing 5: timestamping request HTTP POST with authentication

For a detailed explanation on HTTP Basic Authentication refer to the documentation of your HTTP client library or other online documentation e.g. this Wikipedia article [Wik20].

If the client does not support the manipulation of the HTTP header, try to set the username and password as part of the URL as shown below:
`https://{username}:{password}@{domain}/{path}/`

2.6 Session-id and request tracking

To track problems and errors in the requests or the timestamping system the following HTTP Headers are supported:

- `X-Request-ID`
- `X-Correlation-ID`

If one of these HTTP Headers is included in the timestamping request, the corresponding value is used as session-id for the log messages. This allows the tracking of the request.

2.7 Request parameter

2.7.1 MessageImprint algorithm

The timestamping service supports the following messageImprint algorithm, see [ACPZ01] chapter 2.4.1 for a description of 'messageImprint' - 'hashAlgortihm'.

- SHA-224 (OID: 2.16.840.1.101.3.4.2.4)
- SHA-256 (OID: 2.16.840.1.101.3.4.2.1)
- SHA-384 (OID: 2.16.840.1.101.3.4.2.2)
- SHA-512 (OID: 2.16.840.1.101.3.4.2.3)
- SHA3-224 (OID: 2.16.840.1.101.3.4.2.7)
- SHA3-256 (OID: 2.16.840.1.101.3.4.2.8)
- SHA3-384 (OID: 2.16.840.1.101.3.4.2.9)
- SHA3-512 (OID: 2.16.840.1.101.3.4.2.10)
- RIPEMD-160 (OID: 1.3.36.3.2.1)
- RIPEMD-256 (OID: 1.3.36.3.2.3)

3 Testsystem

For testing purposes, the following parameters should be used.

URL: https://hs-abnahme.a-trust.at/timestamp/

Test user with an RSA Key
Username: test
Password: test

Test user with a ECDSA Key
Username: testEcdsa
Password: test

Test user with blocked password
Username: testBlocked
Password: test

Test user inactive
Username: testInactive
Password: test

URL incl. username and password
URL: https://test:test@hs-abnahme.a-trust.at/timestamp/

4 Production system

The production system is not yet available

5 Example timestamp query

Listing 6 shows a timestamp query with OpenSSL and curl. In line 1 the timestamp request is generated using OpenSSL and saved to 'request.tsq'. In the next step in line 2, the previously generated request is sent to the timestamp service and the response is saved to 'response.tsr'. Finally, in line 3 the response is validated against the request using OpenSSL.

```
1 openssl.exe ts -query -data loremIpsum.txt -cert -sha256 -no_nonce -out
    request.tsq
2 curl.exe -s -S -X POST -H "Content-Type: application/timestamp-query"
    --data-binary @request.tsq
    "https://test:test@hs-abnahme.a-trust.at/timestamp/" -o response.tsr
3 openssl.exe ts -verify -in response.tsr -queryfile request.tsq -CAfile
    ATrustCaFile.pem
```

Listing 6: Example with openssl and curl

References

[ACPZ01] Adams, C., P. Cain, D. Pinkas, and R. Zuccherato: *Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)*. RFC 3161, RFC Editor, August 2001. http://www.rfc-editor.org/rfc/rfc3161.txt, http://www.rfc-editor.org/rfc/rfc3161.txt.

[Wik20] Wikipedia: *Basic access authentication — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Basic%20access%20authentication&oldid=995307412, 2020. [Online; accessed 29-December-2020].