Bedienungsanleitung für den qualifizierten Zeitstempeldienst des BEV

Version 1.4

März 2009

BEV - Bundesamt für Eich- und Vermessungswesen

1 Dokumenteninformation

1.1 Zweck und Gültigkeit

Dieses Dokument enthält die Bedienungsanleitung für den qualifizierten Zeitstempeldienst des BEV und wird auf der BEV-Homepage veröffentlicht.

1.2 Dokumentenhistorie

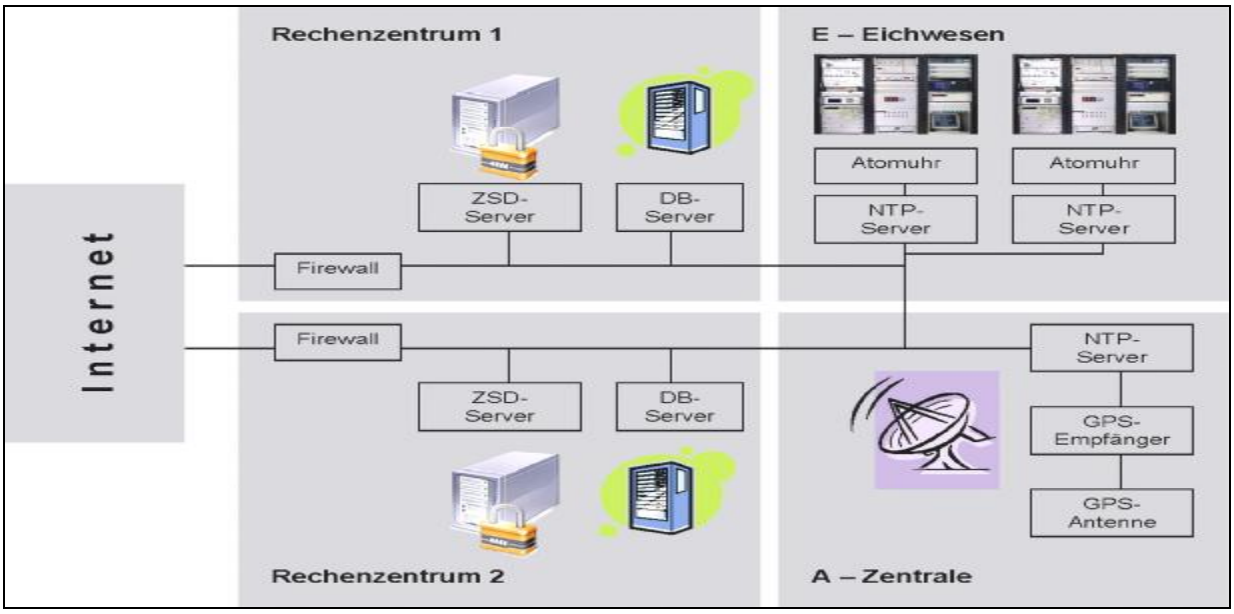
Version	Datum	Änderungsgrund In diesem Dokument
1.0	19.01.2007	Version für die Aufnahme des Testbetriebs
1.1	16.02.2007	Version für die Aufnahme des kostenlosen Betriebs für öffentliche Verwaltung, kostenlose Teststellung für Private. Vereinfachung de r Grafik Seite 1. Korrektur Seite 3: Durchsatz 10 Zeitstempel/ Sekunde. Aussage über Verfügbarkeit gelöscht, da in Policy enthalten.
1.2	28.03.2007	Seite 4: Hyperlink zum Zertifikat eingefügt.
1.3	18.06.2007	Seite 2: Tabelle 1 TimeStampReq ergänzt.
1.4	03.2009	Änderung der Bezeichnung des Dienstes. Auslaufen des kostenlosen Betriebs. Anpassung der Grafik Seite 4. Anpassungen auf Seite 7 des Dokuments – Glossar: Novelle des Signaturgesetzes BGBl I Nr. 8/2008 und BGBl I Nr. 59/2008 sowie der Signaturverordnung BGBl II Nr. 3/2008. Hyperlinks zu diesen beiden Rechtsnormen. Änderung der Kontaktinformationen.

2 Inhaltsverzeichnis

1 Dokumenteninformation ... 2
1.1 Zweck und Gültigkeit ... 2
1.2 Dokumentenhistorie ... 2
2 Inhaltsverzeichnis ... 3
3 Allgemeines ... 4
3.1 Zeitinformation ... 4
3.2 Aufbewahrung von Aufzeichnungen über den Betrieb des QZSD ... 4
3.3 Request/Response-Format ... 5
3.4 Transportprotokoll ... 6
3.5 Reaktionszeit/Durchsatz ... 6
4 HTTP ... 6
5 Kontaktinformationen ... 6
6 Referenzen ... 7
7 Glossar ... 7
8 Anhang ... 8

3 Allgemeines

Seit 2. Jänner 2007 hat das BEV einen qualifizierten Zeitstempeldienst (QZSD) eingerichtet. Dieser Dienst wird unter der Policy OID [1.2.40.0.10.1.8.1] betrieben und öffentlich angeboten. Es besteht die Möglichkeit, eine kostenlose Teststellung einzurichten.



Schematische Darstellung

3.1 Zeitinformation

Die Zeitinformation wird über NTP-Server von den Atomuhren des BEV bezogen.
Die Genauigkeit der Atomuhr beträgt $5 \cdot 10^{-13}$ s.
Die Genauigkeit der NTP-Server beträgt 20 ms.
Die Zeitinformation im Zeitstempel wird mit einer Genauigkeit von 1 Sekunde angegeben.
Ein Ausfall der Atomuhr bzw. der NTP-Server bewirkt keine Änderungen auf den Betrieb des QZSD. Ist der NTP-Server des BEV nicht verfügbar, so wird die Zeitinformation von einem GPS-Empfänger verwendet.

3.2 Aufbewahrung von Aufzeichnungen über den Betrieb des QZSD

Der QZSD speichert Log-Einträge; diese werden 3 Jahre aufbewahrt.
Folgende Daten werden in eine Oracle DB geloggt:

- Uhrzeit/Datum
- IP-Adresse des Senders
- eingehender Request und retournierte Response
- Synchronisations-Ereignisse des NTP-Clients

3.3 Request/Response-Format

Der QZSD implementiert die Zeitstempel-Funktionalität gemäß [RFC 3161] und dem spezifizierten Profil der ETSI [ETSI TS 101 861].
Der QZSD akzeptiert Requests in folgendem Format:

Format von TimeStampReq	
Feld	**Wert**
version	V1 (1)
messageImprint/ hashAlgorithm	SHA-1, SHA-256, SHA-384, SHA-512, RIPEMD-160
messageImprint/ hashedMessage	Hash-Wert für den Zeitstempel
reqPolicy	s. [RFC 3161]
Nonce	s. [RFC 3161]
certReq	s. [RFC 3161]

Tabelle 1 TimeStampReq

Etwaige im TimeStampReq enthaltenen Extensions werden nicht ausgewertet und ein Fehler-Code retourniert. Ein Zeitstempel wird nicht erstellt (Verhalten lt. RFC 3161).
Der QZSD-Server retourniert Responses (TimeStampResp) im Format gemäß [RFC 3161].
Die im TimeStampResp enthaltene und vom QZSD signierte Struktur TSTInfo entspricht folgendem Format:

Format von TSTInfo	
Feld	**Wert**
version	V1 (1)
policy	[OFFEN]
messageImprint	Wert vom Request
serialNumber	Wert einer SEQUENCE im Oracle Cluster beginnend mit 1 (bis zu 160 Bit lang)
genTime	Zeitangabe im Format YYYYMMDDhhmmss.sssZ
accuracy	millis=500
ordering	False
Nonce	Wert vom Request
Tsa	CN=“Sicherer Zeitstempeldienst“ OU=“Bundesamt für Eich- und Vermessungswesen“ O=“Republik Österreich“ C=“AT“

Tabelle 2 TSTInfo

Format der CMS-Signatur	
Signatur Algorithmus	SHA-1 mit RSA

Tabelle 3 CMS-Signatur

3.4 Transportprotokoll

Als Transportprotokoll wird gemäß [RFC3161] HTTP ohne Server-/Client-Authentisierung verwendet.
Abweichend von [ETSI TS 101 861] Abschnitt 6 wird **kein** *store and forward*-Protokoll realisiert.

3.5 Reaktionszeit/Durchsatz

Es können zumindest 10 Zeitstempel / Sekunde erstellt werden.

4 HTTP

Das Zertifikat ist unter der URL
http://www.signatur.rtr.at/SearchCertificatesServlet?DN=O%3DBundesamt+f%FCr+Eich-+und+Vermessungswesen%2C+C%3DAT&CA=alle&validFrom=&validTo=&serialNumberFrom=&serialNumberTo=&revokedDateFrom=&revokedDateTo=&status=alle&Submit=Suchen&locale=de

und der QZSD-Dienst ist unter der URL
http://szsd.bev.gv.at/szsd/service

für die Kundinnen und Kunden verfügbar gemacht.

Ein JAVA Beispielprogramm befindet sich im Anhang.

5 Kontaktinformationen

Technische Anfragen, konkrete Betriebswünsche richten Sie bitte an:

BEV – Kundenservice

Tel.: +43-(0)1-21110 – 2160
Fax: +43-(0)1-21110 – 2162
E-Mail: kundenservice@bev.gv.at

Ihre notwendigen Angaben zur Freigabe sind:

- IP Adresse des Request-Clients (z.B.: 192.168.0.34)
- Client Betriebssystem

Für allgemeine und kaufmännische Anfragen wenden Sie sich bitte an:

BEV – Physikalisch technischer Prüfdienst (PTP)
Tel.: +43-(0)1-21110 – 6327
Fax: +43-(0)1-21110 – 6000
E-Mail: ptp@bev.gv.at

6 Referenzen

Referenz	Titel	Verfasser	Datum
[ETSI TS 101 861]	Time Stamping Profile, V1.2.1; http://webapp.etsi.org/WorkProgram/Report_WorkItem.asp?WKI_ID=14947	ETSI	03.2002
[ETSI TS 102 023]	Electronic Signatures and Infrastructures (ESI): Policy requirements for time-stamping authorities, V1.2.1; http://webapp.etsi.org/workprogram/Report_WorkItem.asp?WKI_ID=17661	ETSI	01.2003
[PVP]	Spezifikation Portal Verbund Protokoll, V1.5.3; http://zmr.bmi.gv.at/files/PortalVerbundProtokollV1.5.pdf	BMI	04.2002
[RFC 2030]	Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI; http://www.ietf.org/rfc/rfc1305.txt	IETF	10.1996
[RFC 2246]	The TLS Protocol – Version 1.0; http://www.ietf.org/rfc/rfc2246.txt	IETF	01.1999
[RFC 3161]	Internet X.509 Public Key Infrastructure – Time-Stamp Protocol; http://www.ietf.org/rfc/rfc3161.txt	IETF	08.2001
[RFC 3628]	Policy Requirements for Time-Stamping Authorities (TSAs); http://www.ietf.org/rfc/rfc3628.txt	IETF	11.2003
[ZERT]	X.509 Zertifikatserweiterungen für die Verwaltung, V 1.0.2; http://www.cio.gv.at/it-infrastructure/pki/ X509ext-20030218.pdf	CIO	02.2003

7 Glossar

Abkürzung	Bezeichnung	Beschreibung
BEV	Bundesamt für Eich- und Vermessungswesen	Bundesbehörde mit den Aufgabenbereichen Vermessung und Geoinformation und Mess- und Eichwesen, dem Bundesministerium für Wirtschaft, Familie und Jugend nachgeordnet.
HSM	Hardware Security Module	Eine separate Hardware-Komponente, die für die sichere Erzeugung und Anwendung von kryptographischen Schlüsseln verwendet wird. (wird erst in Phase 2 benutzt)
NTP	Network-Time-Protokoll	IP-Protokoll für die Übermittlung von Zeit-Information.
PVP	Portal Verbund Protokoll	Das Protokoll definiert die Kommunikation zwischen Portalen im Portalverbundsystem der österreichischen Verwaltung.
SigG	Signaturgesetz	Bundesgesetz über elektronische Signaturen (Signaturgesetz - SigG), BGBl. I Nr. 190/1999 in der gültigen Fassung
SigV	Signaturverordnung	Verordnung des Bundeskanzlers über elektronische

		Signaturen (Signaturverordnung 2008 – SigV 2008), BGBl. II Nr. 3/2008 in der gültigen Fassung
QZSD	Qualifizierter Zeitstempeldienst	Ein Dienst, der mittels kryptographischen Methoden Daten mit vertrauenswürdiger Zeitinformation sicher verbindet.

8 Anhang

IAIK-Bibliothek zu beziehen bei:

Mario Ivkovic,
IAIK – Graz University of Technology
Inffeldgasse 16a, 8010 Graz, Austria
Tel: +43 316 873 5528
http://jce.iaik.tugraz.at

SampleClient.java

```
package demo;

import iaik.asn1.ObjectID;
import iaik.asn1.structures.AlgorithmID;
import iaik.asn1.structures.GeneralName;
import iaik.asn1.structures.GeneralNames;
import iaik.tsp.MessageImprint;
import iaik.tsp.PKIStatus;
import iaik.tsp.TSTInfo;
import iaik.tsp.TimeStampReq;
import iaik.tsp.TimeStampResp;
import iaik.tsp.TimeStampToken;
import iaik.tsp.transport.http.TspHttpConstants;

import java.io.ByteArrayOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.math.BigInteger;
import java.net.HttpURLConnection;
import java.net.URL;

import sun.misc.BASE64Encoder;

/**
 * This class demonstrates the use of the IAIK TSP 2.0 library for
 * generating a time-stamp by using the BEV QZSD time-stamping service.
 *
 * For executing this demo, the IAIK TSP 2.0 and all the JARs it depends
 * upon (IAIK-JCE and IAIK-CMS) must be on the classpath.
 */
public class SampleClient {
    private static final String SERVICE_URL = "http://szsd.bev.gv.at/service";
    //private static final String SERVICE_URL = "http://praktiker:8888/szsd/service";
    //private static final String USER = "Praktiker";
    //private static final String PASSWORD = "xYlCB35";
    private static final String POLICY_ID = "1.2.40.0.10.1.8.1";
    private static final byte[] DATA_TO_HASH = "sample data".getBytes();
    private static final String ISSUER =
        "CN=Test CA - Zeitstempeldienste,OU=Technik und Standards,O=Stabstelle IKT-Strategie des Bundes,C=AT";
    private static final BigInteger SERIAL = new BigInteger("1");


    public static void main(String[] args) throws Exception {

        // build the TimeStampReq
        byte[] hashedData =
            MessageImprint.calculateHash(DATA_TO_HASH, AlgorithmID.sha1);
        MessageImprint msg = new MessageImprint(AlgorithmID.sha1, hashedData);
        TimeStampReq req = new TimeStampReq(msg);
        req.setCertReq(false);
```

```
req.setNonce(new BigInteger("12345678")); // should be random
req.setTSAPolicyID(new ObjectID(POLICY_ID));

// set up an HTTP connection for sending the request
URL url = new URL(SERVICE_URL);
HttpURLConnection connection = (HttpURLConnection) url.openConnection();
OutputStream out;
InputStream in;

connection.setRequestMethod("POST");
connection.setDoOutput(true);
connection.addRequestProperty("Content-Type",
        TspHttpConstants.CONTENT_TYPE_REQUEST);
if (USER != null && PASSWORD != null) {
    // add HTTP authentication
    String encodedAuth = new BASE64Encoder().encode((USER + ":" + PASSWORD).getBytes());
    connection.setRequestProperty("Authorization", "Basic " + encodedAuth);
}

// write request
out = connection.getOutputStream();
out.write(req.getEncoded());
out.flush();

// read response
ByteArrayOutputStream responseBytes = new ByteArrayOutputStream();
byte[] buf = new byte[1024];
int bytesRead;
TimeStampResp resp;

in = connection.getInputStream();
while ((bytesRead = in.read(buf)) > 0) {
    responseBytes.write(buf, 0, bytesRead);
}
responseBytes.flush();
resp = new TimeStampResp(responseBytes.toByteArray());

// check for correct HTTP response type
if (!TspHttpConstants.CONTENT_TYPE_REPLY.equals(connection.getContentType())) {
    System.err.println("Wrong Content-Type in response");
    System.exit(1);
}

// check for errors or bad data in response
PKIStatus pkiStatus = resp.getPKIStatusInfo().getPKIStatus();

if (pkiStatus.getStatus() != PKIStatus.GRANTED) {
    System.err.println("PKIStatus not GRANTED");
    System.exit(1);
}

TimeStampToken token = resp.getTimeStampToken();
TSTInfo tstInfo = token.getTSTInfo();


if (!token.getTSTInfo().getNonce().equals(req.getNonce())) {
    System.err.println("Nonce does not match");
    System.exit(1);
}

if (!tstInfo.getTSAPolicyID().getName().equals(POLICY_ID)) {
    System.err.println("Wrong policy ID");
    System.exit(1);
}

if (!tstInfo.getMessageImprint().equals(req.getMessageImprint())) {
    System.err.println("Message Imprint does not match");
    System.exit(1);
}

GeneralNames names = token.getCertID().getIssuerSerial().getIssuer();
GeneralName name = (GeneralName) names.getNames().nextElement();

if (!name.getName().toString().equals(ISSUER)) {
    System.err.println("Issuer does not match");
```

```
            System.exit(1);
        }

        if (!token.getCertID().getIssuerSerial().getSerialNumber().equals(SERIAL)) {
            System.err.println("Serial number does not match");
            System.exit(1);
        }

        System.out.println("Sent TimeStampReq OK, Got TimeStampResp OK");
    }

}
```