S.A.F.E. End User Manual Transparency Software 1.2.0

Please note that only the German version is binding. No guarantee is given for the accuracy of the translation.

This document contains all relevant information needed to use the S.A.F.E Transparency Software. Please read these instructions carefully before using the software for the first time.

1. Accessing the Transparency Software

1.1. System requirements

In order to run the Transparency Software, your system must meet the following requirements:

You must have Java Runtime or the Java Development Kit (JDK) version 16 or later installed. This software may already be pre-installed on your PC. If not, you can download a version at https://jdk.java.net/17/ or at https://www.oracle.com/java/technologies/downloads/#java17

At least 50 MB of memory (RAM) must be available to operate the software.

You can look up the additional system requirements at the OpenJDK and Oracle websites https://jdk.java.net or https://www.oracle.com/java/.

1.2. Download the S.A.F.E. Transparency Software

The Transparency Software is a computer application that can be operated either on a stationary or mobile PC system. The application is based on the Java framework and as such requires Java being installed on the PC system.

To install and launch the application, please proceed as follows:

1. Download the current version of the Transparency Software to your computer and unpack it into a folder of your choice.
   https://www.safe-ev.de/de/transparenzsoftware.php
2. Ensure that a current version of the Java framework is installed. If you are unable to open the Transparency Software, it is likely that JAVA is not installed.
3. Double click to open the Transparency Software

2. Creating Data Tuples in the Charging Device

During a charging procedure at publicly accessible charging systems, a range of values/attributes are captured that are required for invoicing later. Apart from date and energy meter information, this mainly includes Contract ID/Session ID/Transaction ID, which unequivocally link the invoice recipient with the measurement values. These values are collated into what is referred to as a 'data tuple'. The current requirements for the content of a data tuple according to calibration law are set down in the PTB's REA document 6 A.

https://oar.ptb.de/files/download/58d8ffad4c9184f55a2f94e3

The data tuples are created in the charging device in compliance with calibration law, and then transferred to the invoicing server via OCPP. This is where data are stored long-term, and

the charging system user is invoiced by a Mobility Service Provider [MSP]. The invoice recipient obtains access to this data tuple as part of the invoicing process.

The data tuples are digitally signed so that no modifications/manipulation of calibration law-compliant attributes can take place during the transfer of data tuples from the charging device to the invoicing server, nor from there to the invoicing process respective the user. The data tuple's public key gives the user the option to verify the validity of the digital signature. Should any attributes have been modified or falsified, verification of the signature cannot return a positive result – which creates transparency between data capture and invoice. A user-friendly way of verifying digital signatures is the objective of the Transparency Software, and is presented here in detail.

3. Accessing Public Keys for Data Tuples

In order to verify digitally signed data tuples, you need the following information:

- Digitally signed data tuple (in hexadecimal code or as a file)
- Public key for the charging device

Public keys are calibration law related numerical sequences that are unique to each charge point, and are printed onto measuring devices. They enable charge point users to verify the correctness of remote readings of measurement values. The public key can be a component of the digitally signed data tuple you receive from your EMP. If the public key is not contained in the data tuple, you can access it as follows:

- Printed onto the (local) charging device
- Directly from its MSP
- From the Federal Network Agency's (Bundesnetzagentur) PKI database: https://www.bundesnetzagentur.de/DE/Sachgebiete/ElektrizitaetundGas/Unternehmen_Institutionen/E-Mobilitaet/Ladesaeulenkarte/start.html

When verifying a digital signature using the Transparency Software, please note that you can check the public key of the charging device, and make sure that you trust the source from which you obtained the key. Public keys printed on invoices or contained in the data tuple itself may be incorrect, and may have to be verified by the user.

4. Verifying the Data Tuple

4.3. Transparency Software 1.2.0 interface

When opening the Transparency Software 1.2.0 application, the following interface appears – initially without any data tuple content.

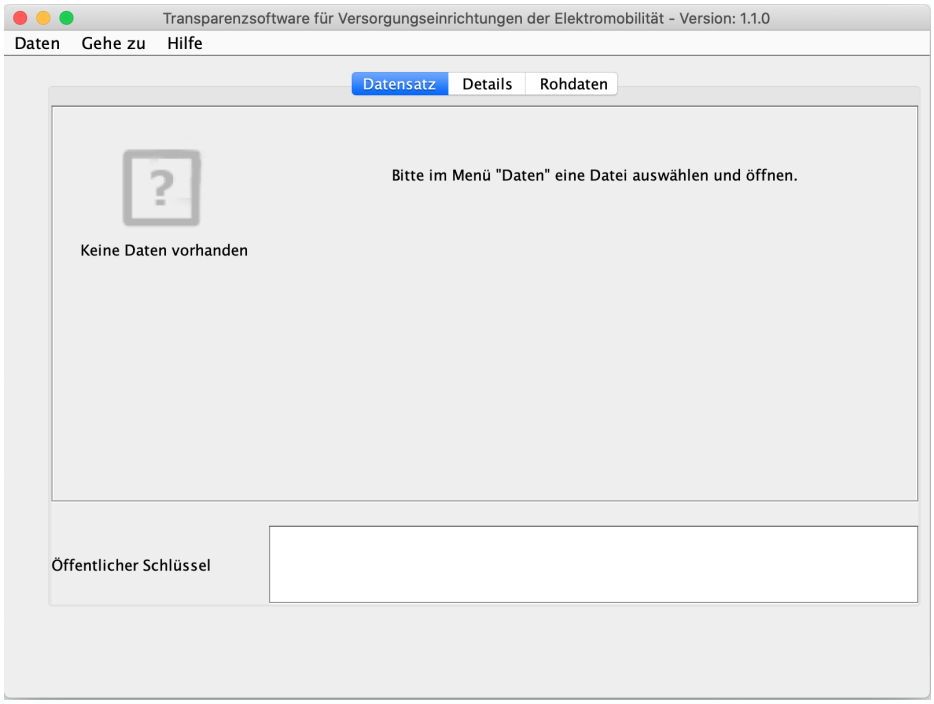


Figure 4.1.1: Transparency Software interface without data tuple content

The 'File' tab can be used to conduct verification either by opening a saved file, or by entering sets of measurement value data manually in hexadecimal form:

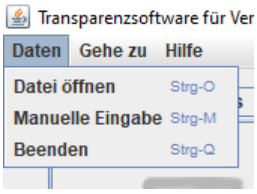


Figure 4.1.2: Entering data into the Transparency Software.

By displaying the measurement data in the next step, invoiced items can be compared – thus creating transparency.

4.4. Entering data tuples via saved files

An XML file or Porsche Charging Data file can be opened in the Transparency Software using File ➡ Open file [CTRL-O]. If a live boot medium is being used, external drives such as USB sticks are included in the /run/media/root directory. This is the standard directory displayed on opening.

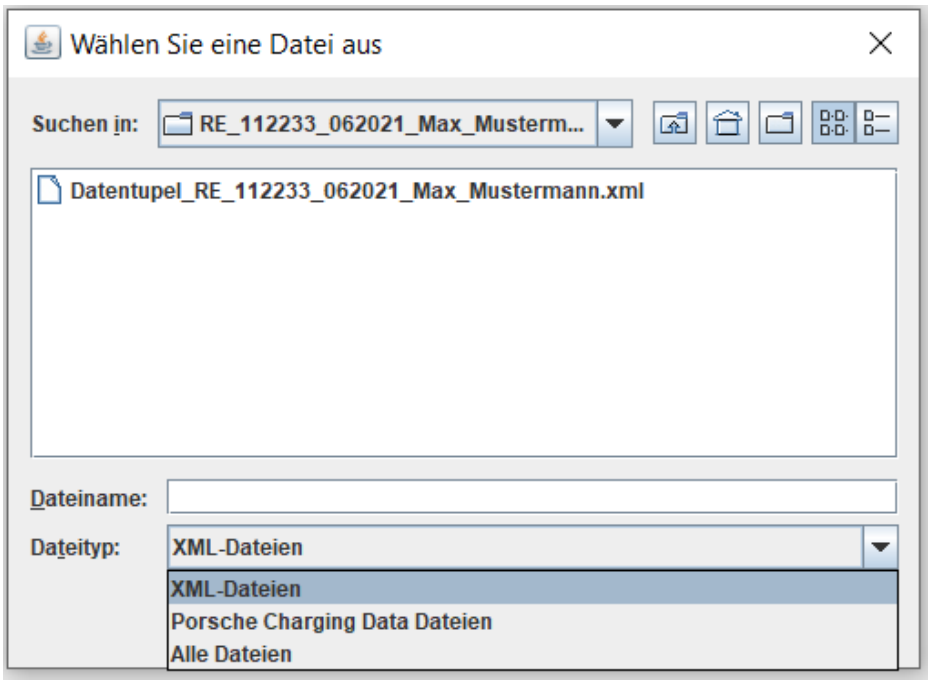


Figure 4.2.1: Dialogue box for opening files.

The following view is displayed after opening the file:

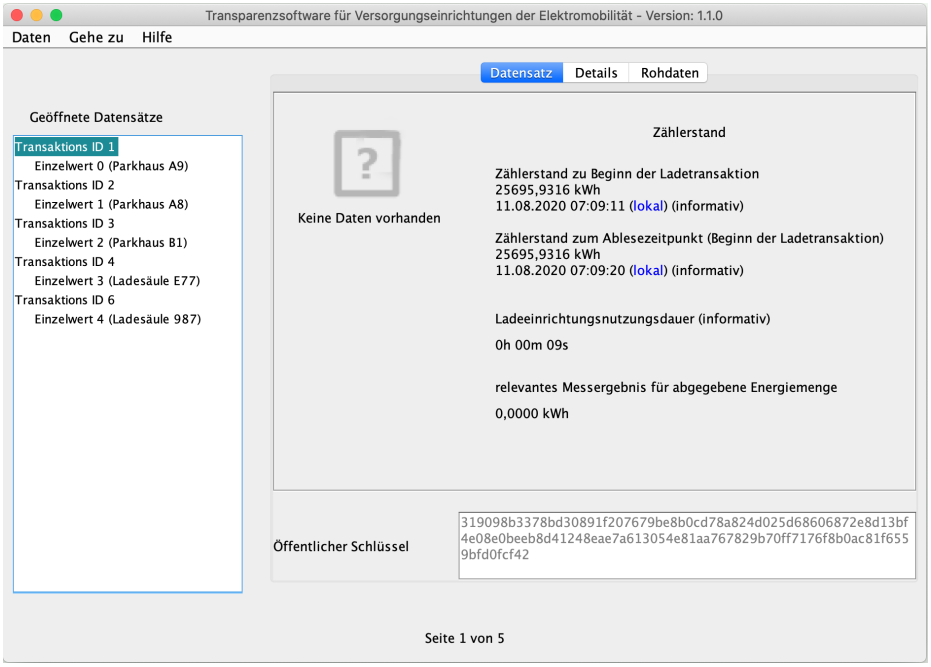


Figure 4.2.2: Transparency Software interface with XML data sets opened (left hand side).

The content of the data value set of the first transaction as well as the public key are immediately displayed to the user in their unvalidated state. Using Transaction ID/Contract ID/Session ID, the charging procedure in the data tuple is uniquely associated with a specific charging location, charging date, and invoice recipient. The

invoice recipient is then shown this ID at the corresponding invoiced item, and it can then be selected accordingly in the individual values display in the left hand column of the user interface. Selecting it then prompts direct validation of the digital signiture by the Transparency Software. Figure 4.2.3. shows how the green tick indicates successful validation. The data tuple's attributes are still in their original state when signed in the charging device, a process that occurred in compliance with German calibration law.

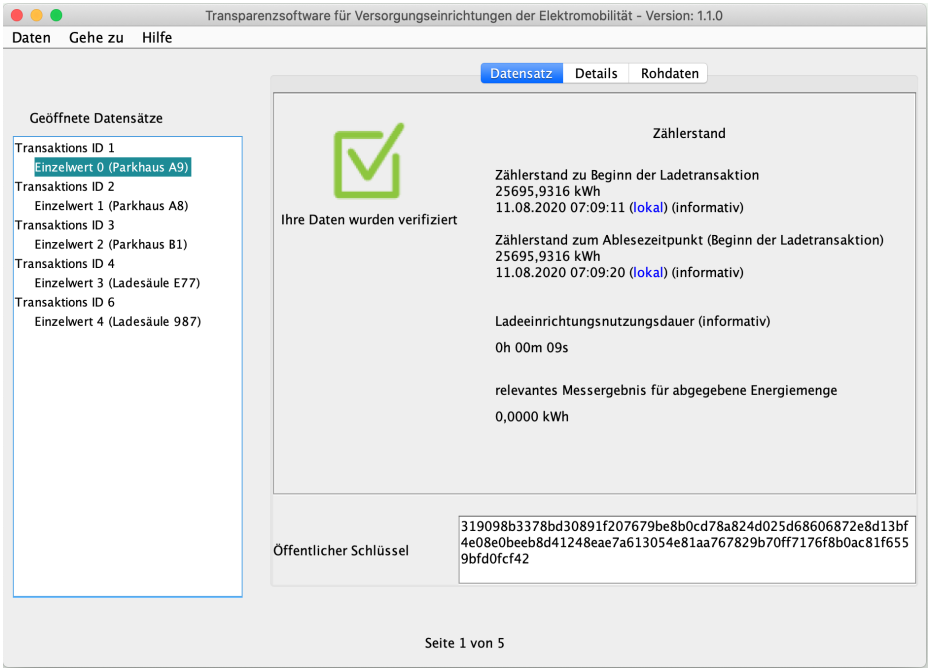


Figure 4.2.3: Transparency Software interface with XML data sets opened (left hand side).

The user is shown a translation of the content of the hexadecimal data tuple, including information relevant to the charging procedure, such as start/end and amount of charge, and of course furnished with the respective time stamps for retrieval as well as charging duration. The user can verify these items by comparing directly against the respective invoice items.

Important: The file may contain only one set of data values, in which case a selection from multiple options in the left column is not available, and the digital signature is verified directly on opening the file.

Important: A range of data tuple formats are on the market, which means that the way their content is displayed may vary.

Important: The public key of the data tuple can always be verified in its hexadecimal form – see the box at the bottom of the window.

The 'Details' tab allows the user to access further information regarding signed data tuple content.

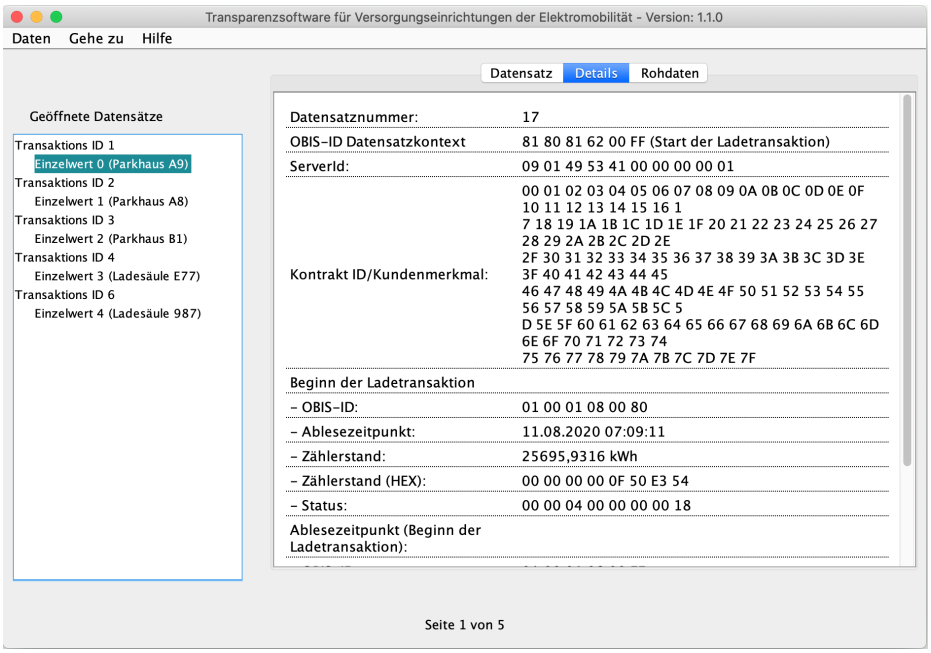


Figure 4.2.4: Transparency Software interface with further information on verified sets of data values.

The 'Dataset' tab allows the hexadecimal display of the data tuple being viewed as a single string.

4.3. Entering data tuples using manual data entry

Dialogue window opens via ➡ Manual data entry [CTRL-M]

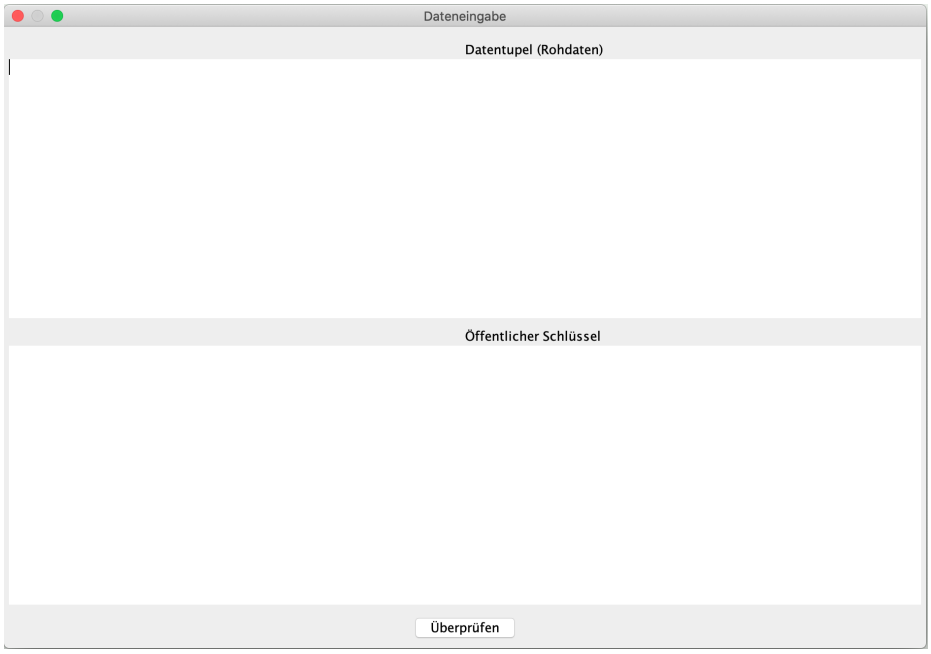


Figure 4.3.1: Interface window for manual data entry (without data).

The hexadecimal values for the data tuple ('Raw data') and the public key can now be entered using copy&paste.

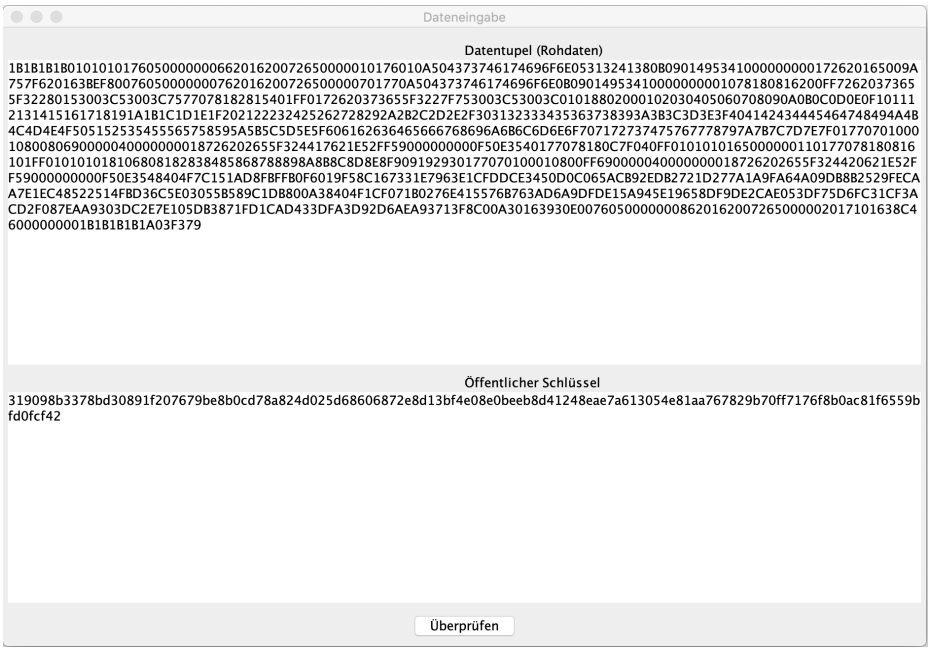


Figure 4.3.2: Interface window for manual data entry (with data).

Following an additional click on 'Verify', the digital signature is verified, and the signed data tuple is displayed in translated form – see Chapter 4.2.

4.4. Error messages in the Transparency Software

Should the entered data contain errors, the user receives an error message and an error code on a red background. In principle, this means that an error has occurred with the signed measurement data or the digital signing process, and that transparent invoice verification cannot be performed. Error messages are displayed in the same place on all three tabs.

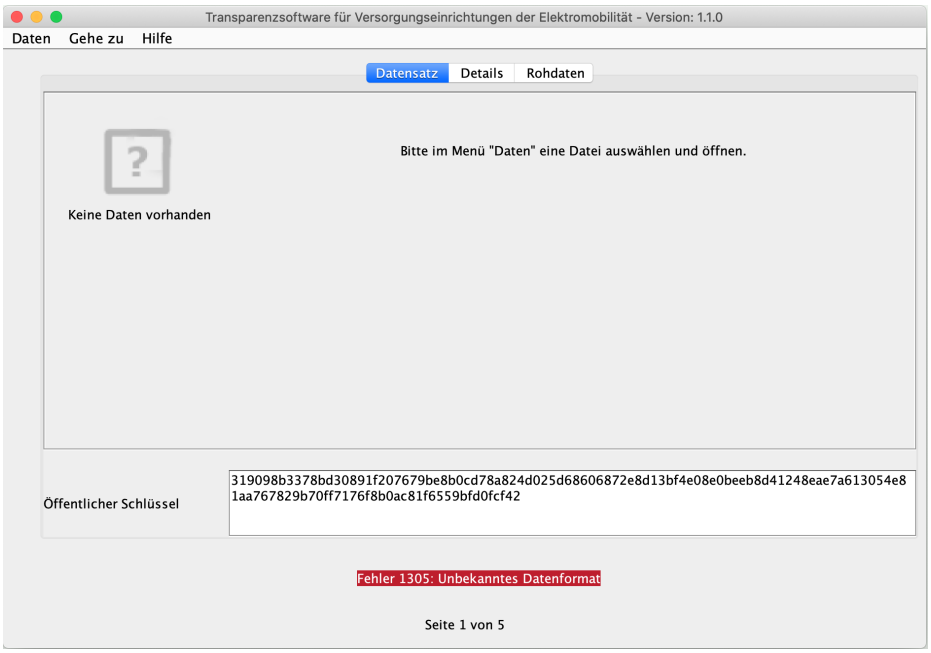
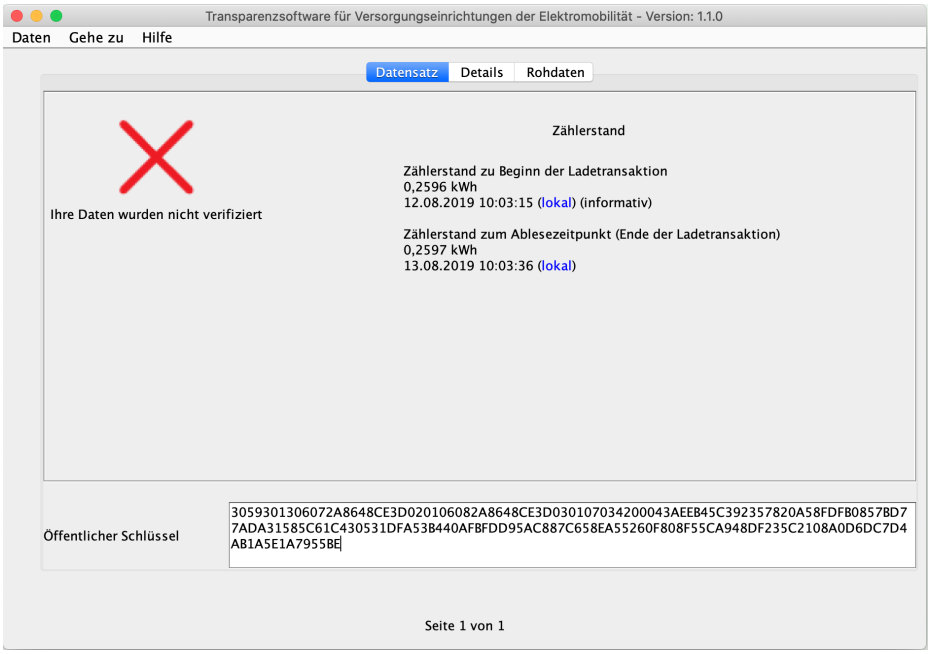


Figure 4.4.: Error message display on the Transparency Software interface.

If the digital data signature cannot be verified because of an incorrect public key or modified data, the green tick is replaced with a red 'X', and the error message 'Your data has not been verified' is displayed:



Important: In this case, please immediately contact your EMP (invoice issuer) and mention the error code. A customer service unit will then clarify the issue.

4.5. Further functions of the Transparency Software

Version information can be obtained via the 'Help / About' menu options. This shows which version of the software is currently being used, a checksum (SHA-256) for the software, as well as a list of the range of libraries used within the software.

The functions 'Go to / Next entry' (or CTRL-N) and 'Previous entry' (or CTRL-P) can be used to switch between different transactions – as long as the file or data set contains more than one transaction. If no list or tree diagram with transactions is displayed on the right hand side, then only one transaction is included, and the switching functions are not activated.

The menu option 'Help / Help' contains a link to the S.A.F.E. association's website.

The menu option 'Data / Close' can be used to close the application.

List of error codes:

Error code	Error message
Error 1101	Invalid length for ALFEN data sets
Error 1102	Invalid ALFEN data
Error 1201	Unable to create %s file
Error 1202	Unable to write %s file
Error 1203	Output format error
Error 1301	Invalid base 32 data
Error 1302	Invalid base 64 data
Error 1303	Invalid Hex data
Error 1304	Unknown encoding type detected
Error 1305	Unknown data format
Error 1306	Invalid XML format for data entered
Error 1307	Invalid curve name
Error 1308	Unable to read input parameters
Error 1309	Invalid embedded public key
Error 1310	Invalid public key
Error 1311	Invalid length of signature data
Error 1401	No data entry file entered
Error 1402	No public key found to verify data (this can also occur if unable to read data)
Error 1403	Invalid OCMF signature algorithm
Error 1404	Invalid OCMF version
Error 1405	Invalid OCMF data
Error 1204	Unable to create output file because it already exists.
Error 1406	Unable to parse use data
Error 1407	The indicated path does not lead to a file
Error 1501	Unable to read file
Error 1502	PCDF invalid, missing signature

Error 1503	Invoicing not allowed
Error 1504	Invalid charging procedure counter
Error 1505	Invalid charging duration
Error 1506	Invalid consumption data
Error 1507	Data missing in tuple
Error 1508	Invalid DCMeter type
Error 1509	End marker missing in data tuple
Error 1510	<ETX> missing
Error 1511	Incorrect charging data format
Error 1512	Incorrect hardware serial number length
Error 1513	Incorrect OBIS code
Error 1514	Invalid charging procedure ID
Error 1515	Invalid length of charging procedure ID
Error 1516	Invalid length of software checksum
Error 1517	Invoicing not allowed
Error 1518	<STX> missing
Error 1519	Corrupted time information
Error 1520	Invalid length of time information
Error 1521	Invalid time signal
Error 1522	Invalid PCDF signature
Error 1601	SML data incomplete for verification
Error 1602	Invalid measuring value unit in SML data set
Error 1603	Invalid Server ID value transmitted
Error 1604	Invalid signature in XML file
Error 1605	No measurement values transmitted in XML
Error 1606	Time stamp missing for measurement value
Error 1607	Time stamp missing for measurement value
Error 1608	Invalid SML, missing Customer ID
Error 1609	Invalid SML, missing logbook entry index
Error 1610	Invalid SML, missing meter reading

Error 1611	Invalid SML, missing OBIS identifier
Error 1612	Invalid SML, missing pagination
Error 1613	Invalid SML, missing seconds count
Error 1614	Invalid SML, missing Server ID
Error 1615	Invalid SML, missing signature
Error 1616	Invalid SML, missing time stamp
Error 1617	Invalid SML, time stamp missing for Customer ID
Error 1618	Invalid SML data
Error 1701	Unknown encoding
Error 1702	Validating error when processing data
Error 1703	Public key missing in entered data
Error 1704	Entered data do not contain digitally signed data
Error 1705	No measurement values in transaction start value
Error 1706	No start value in transaction
Error 1707	No measurement values in transaction end data
Error 1708	No end value in transaction
Error 1709	Selected file contains no value fields
Error 1710	Unable to decode public key entered
Error 1711	No data in public key entered
Error 1712	Unknown public key format
Error 1713	Unable to decode entered signature
Error 1714	No information in entered digitally signed data
Error 1715	Unknown digitally signed data format
Error 1716	Unknown format in signed data
Error 1717	More than one starting value in transaction
Error 1718	More than one end value in transaction
Error 1719	Unable to verify data
Error 1720	Unable to translate Mennekes entry format
Error 1721	Data contains no unique public key