The Logical Hierarchy of Software Protection Requirements in IR46

Xuezhong Fan¹, Congcong Li²

¹Shandong Aipu electrical equipment co., ltd Jinan Gaoxin, branch, Jinan, 250000, China;
²State Grid Shandong Electric Power Research Institute, Jinan, 250000, China.
The corresponding author’s e-mail address: 286953555@qq.com

Abstract. The updated version of OIML recommendation “Active Electrical Energy Meters” (IR46) requires protection of metrological properties, it provides a new idea for the development of new type watt-hour meter. These requirements are mainly focused on software protection, because for modern static electricity meters, software is the key part for processing and storing legal metrological data. According to the logical relationship between these requirements, this paper analyzes them to form a logical level. The validation methods are analyzed at the same time, which shows the feasibility and the necessity of new generation smart meter.

1. Introduction

Nowadays software in electricity meters plays an important role for control of measurement, processing of measurement data, storing of legal metrological results, communication and interaction to users, etc.

IR46 standard refers to IEC standard before revision and MID after revision. The difference is mainly embodied in the definition of intelligent watt-hour meter level, the concept of relevant current parameters and the form of watt-hour meter detection and evaluation. In the future, smart watt-hour meter will be a new generation of smart watt-hour meter with its own characteristics. It integrates the existing specifications with IR46 and emphasizes management unit. In order to adapt to the development of huge data era, it also puts forward new requirements for the communication mode of smart watt-hour meter, so as to achieve the separation of equipment components and the measurement of work. Independent energy module and non-measurement module upgrade do not affect the effect of measurement module.

For accuracy of measurement and justice of trade, the updated version OIML recommendation “Active Electrical Energy Meters” (IR46) [1] requires protection of metrological properties which are mainly focused on software. The requirements in IR46 are based on OIML D31 [2] and WELMEC 7.2, which are about software requirements for general measuring instruments. At the same time, IR46 considers some special characteristics of electricity meters.

In this paper we analyze the underlying logical relation of the requirements. According to the logical relation we can classify the requirements to into several levels to form a hierarchy.
2. The Logical Hierarchy

The basis for protection is to guarantee that an electricity meter executes in accordance with the design logic as much as possible, namely, the hardware performs operations as the hardware logic prescribes and the same case for the software. Therefore, we should prevent interference to hardware, which makes the program deviate from its normal execution. At the same time we should prevent interference to codes and data of software, because they are the essential part for legal metrology.

If the basis has been guaranteed, we can have an upper level requirement on the accuracy of functionalities. This kind of requirement is on how to guarantee that the functionalities of a meter have no error and defect.

A further upper level is on regulation and supervision of legal metrology by government. This kind of requirement guarantees that the software in a meter has some functionality which makes it convenient to supervise.

Table 1 shows the above mentioned three levels. In Table 1, the most fundamental level is the bottom level.

| Regulation and Supervision |
|----------------------------|
| Functionalities            |
| Prevention of Interference and Attack to Software Execution, Codes and Data |

2.1. Prevention of Interference and Attack to Software Execution, Codes and Data

A meter is not located in absolutely isolated space. Actually it is interfered by its surrounding environment. There are three common sources of interference from the environment, nature-physical interference, human-physical interference, and human-information interference.

Natural environment could interfere a meter by physical means, such as electromagnetic wave, moisture, hit, etc. We call this kind of interference nature-physical interference[3]. Such as, all means to protect the metrological properties of an electricity meter intended for outdoor locations shall withstand solar radiation.

Human-beings can intrude the software and the hardware of a meter by performing physical interference, such as damage to hardware circuit, damage to memory, etc. Nowadays, a large number of meters have been connected to the Internet as show in figure 1, and the meters connected to the Internet are more vulnerable to attack. Human-beings can also intrude the software through communication interfaces by information exchanging, such as forging data and commands.

Electricity meters shall be provided with the means to protect their metrological property. National authorities shall determine levels of authorised access for software protection (Term 3.6.3), parameter protection (Term 3.6.4) and checking facility event record (Term 3.6.9).

![Figure 1: Connection of Internet of Things](image_url)
To prevent the three sources of interference, there are five sublevels of requirements, which are shown in Table 2. The most fundamental sublevel is on the bottom. Each sublevel is based on its lower sublevel in logical relation.

1. Hardware Stability. A meter should avoid external intrusion so that the hardware can keep stable and normal. Term “3.6.3.2.1” requires a meter should be secured against unauthorized modification, loading, or changes by swapping the memory devices. Term “3.6.7.4” requires protection of the system clock.

2. Hardware Separation. In a meter, maybe not all the hardware components are responsible for legal metrology. We should prevent that the legally non-relevant hardware components inadmissibly influence the legally relevant hardware components. In IR46 it is term “3.6.5”. In practical application, the measurement part and the function part should be relatively independent[4], as show in figure 2.

3. Software Stability. If the above two requirements are satisfied, we can go a further step to software. If the software in a meter allows external revision or setting, it should prevent unauthorized intervention. Term “3.6.3.2.3” requires making unauthorized intervention to software codes and data impossible or evident. Term “3.6.4.1” and term “3.6.4.2” require adjustment of parameters should be authorized. Term “3.6.8” focuses on software update both in local and from remote. For example, The software of a measuring instrument is constructed such that there is no way to modify the parameters and legally relevant configuration but via a switch protected menu[5]. This switch is mechanically sealed in the inactive position, making modification of the parameters and of the legally relevant configuration impossible. To modify the parameters and configuration, the switch has to be switched, inevitably breaking the seal by doing so.

4. Software Separation. Software can also be classified into legally relevant software and legally non-relevant software. Like hardware separation term “3.6.6” is on software separation. For example, an electricity meter is equipped with an optical interface for connecting an electronic device to read out measurement values. The meter stores all the relevant quantities and keeps the values available for being read out for a sufficient time span. In this system only the electricity meter is the legally relevant device. Other legally non-relevant devices may exist and may be connected to the interface of the instrument provided requirement 3.6.5.2 is fulfilled. Securing of the data transmission itself (see 3.6.7) is not required.

5. Protection on Program Logic. The above mentioned four kinds of requirements protect the hardware and software in a meter from external intrusion, which provides a sound base for the logic for program execution. In program logic level, the software provides information exchanging and interaction means to users. Therefore, the software needs to guarantee that the software design has no defect, and prevent attack by using the defects. Term “3.6.3.1” is a general requirement for it. Term “3.6.7.1.2” in on authenticity and integrity of data received from communication or read from insecure storage.
Table 2. Requirements of Prevention of Interference and Attack

| Protection on Program Logic |
|-----------------------------|
| Software Separation         |
| Software Stability          |
| Hardware Separation         |
| Hardware Stability          |

2.2. Functionalities

This level concerns on the functionality correctness and completeness of the software in a meter. It focuses on specific functionalities, which is necessary for a meter. Term “3.6.4.3” is on registering energy, which says that registering shall be stopped when changing the relevant parameters. Term “3.6.7.1.2” is requires that stored or transmitted data shall have necessary information accompanied. Term “3.6.7.3” is on data transmission, which requires that measurement shall not be influenced by transmission delay, and when network services are unavailable, no legal data is lost.

2.3. Regulation and Supervision

Besides basic functionalities, a meter shall provide functions for government’s regulation and supervision. They are shown in Table 3.

Table 3. Requirements for Regulation and Supervision

| Prevention of Cheating Codes |
|------------------------------|
| Records of Critical Events and Data |
| Software Version             |

1. Software Version. Software version is used to identify whether the software in a meters is the version which is declared in type approval. In IR46 it is term “3.6.2”.

2. Records of Critical Events and Data. For arbitration for possible dispute, a meter shall automatic store essential data, and record critical events such as setting parameters. Term “3.6.4.4”, term “3.6.7.2” and term “3.6.9” are for this issue.

3. Prevention of Cheating Codes. It is the highest sublevel of requirements. It prevents the existence of intentional “backdoor”. Term “3.6.3.2.2” is on this issue.

3. Validation Method

Roughly speaking, software design starts from concept model design stage, then to programming language implementation stage, finally to executable software. For the three stages we can have the corresponding validation methods listed in Table 4. In IR46, only the first one and the third one are used. The first one is denoted by “AD” (Analysis of the Documentation and validation of the design). The third one is denoted by VFTSw (Validation by Functional Testing of Software functions).

Table 4. Validation Methods

| Software Design Process | Validation Methods       |
|-------------------------|--------------------------|
| Executable Software     | Functional Testing       |
| Programming Language    | Analysis of Source Codes |
| Concept Model           | Analysis of Design       |
4. Conclusion
In this paper, we analyze the logical relation of the software protection requirements in IR46 to form a logical hierarchy. The base for requirements is the prevention of intervention of hardware and software. Besides we should also consider functionality, as well as regulation and supervision.

References
[1] OIML IR46. *Active Electrical Energy Meters*.
[2] OIML D31. *General Requirements for Software Controlled Measuring Instruments*.
[3] Li Xiangfeng, Chen Jun, Liu Ying. Life evaluation on electrical energy meters and application for long-life meters[J]. Hubei Electric Power, 2008, 32(S2):22-24,37.
[4] GAO Bo. Research of smart meter design scheme based on IR46[J]. Electrical Measurement & Instrumentation, 2017, 54(17):77-81.
[5] WELMEC 7.2. *Software Guide (Measuring Instruments Directive 2004/22/EC)*.