Formalization of Smart Metering Requirements

Andreas Fuchs
Fraunhofer Institute for Secure Information Technology
Rheinstrasse 75
Darmstadt, Germany
fuchs@sit.fraunhofer.de

Sigrid Gürgens
Fraunhofer Institute for Secure Information Technology
Rheinstrasse 75
Darmstadt, Germany
guergens@sit.fraunhofer.de

Donatus Weber
University of Siegen
Hölderlinstrasse 3
Siegen, Germany
donatus.weber@uni-siegen.de

Christian Bodenstedt
University of Siegen
Hölderlinstrasse 3
Siegen, Germany
christian.bodenstedt@uni-siegen.de

Christoph Ruland
University of Siegen
Hölderlinstrasse 3
Siegen, Germany
christoph.ruland@uni-siegen.de

ABSTRACT

Today's industries and households have a growing need for resources like electricity obtained and measured via house connections. For the future, european and national regulations require providers to enable additional services such as accurate monthly bills and consumption information regarding the actual time of use. Those regulations and future use cases like charging stations for E-Mobility or the need for accurate real-time consumption information in Smart Grids require the classical meters to be replaced by *Smart Meters* that utilize embedded systems providing network connectivity to the backend.
These functionalities raise additional security concerns that are addressed by regulations and laws of the metrology domain. However, the current directives in this area rely only on verbal definitions of these security requirements. This paper analyses security requirements from the Measuring Instruments Directive 2004/22/EC (MID) and demonstrates a possible formal representation ruling out the drawbacks of textual descriptions and enabling formal reasoning and proving of lawful requirements.

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements Specifications—*Methodologies*; D.2.4 [**Software Engineering**]: Software/Program Verification—*Formal Methods*

General Terms

Security, Legal Aspects

Keywords

Smart Metering, Formal Methods, Security Requirements

S&D4RCES 2010 September 14, 2010, Vienna, Austria.

1. INTRODUCTION

Many technical products have to comply to standards or fulfill special lawful requirements to be allowed for legal use. Typical examples are metering devices which have to pass assessment procedures to ensure conformity. The basis for the assessment procedures are laws which formulate the requirements using a verbal description. However, these descriptions are often not precise enough and can be interpreted in different ways.

This results in problems, especially regarding lawful requirements covering security and dependability aspects of metering devices. An even bigger issue exists in the procedure of checking compliance to lawful requirements of a newly developed device before being placed on the market. Based on verbally described requirements, the check cannot be automatized. The automatic validation of the device against the requirements would imply to have them formalized. A possible approach for the formalization of security and dependability requirements for metering devices is the use of the Security Modelling Framework SeMF [9]. This paper describes first results of using this framework to formalize a typical security requirement for metering devices. It is organized as follows: In the next section we give a brief description of the metering domain, Section 3 describes today's type approval procedures for metering devices and existing approaches for requirement formalizations. Section 4 then gives a brief introduction to the Security Modeling Framework SeMF, based on which we formalize some exemplary metering device requirements of the Measuring Instruments Directive 2004/22/EC (MID) in Section 5. Finally, we close with our conclusions and future work in Section 6.

2. MOTIVATION FOR SMART METERS

Today's industries and households have a growing need for resources like electricity, water, gas or heat. The infrastructure of modern countries offers customers the possibility to easily obtain these resources via the house connection.

For billing, the consumed resources have to be measured with special-purpose meters. The area in research and jurisdiction responsible for the necessary instruments and measurements is the metrology domain. In the case of an electricity meter for example, the metrology domain covers the

complete process from measuring of the consumption over transferring the meter readouts to the energy provider up to billing the energy customer for this measured consumption.

The measurement process is subject to strict lawful requirements in terms of accuracy, dependability and security. The European metrology domain is regulated by the Measuring Instruments Directive [7]. The MID is an European Union directive that has been published 2004-05-31. It had to be transferred to national law in each of the European Union member countries until 2006-01-04 and should be applied since 2006-10-30. The contents of the MID cover various types of devices and systems with a measuring function like electricity, water, gas or heat meters. The document is structured into a main part describing how this directive has to be applied, and several annexes which contain the actually requirements to the metering devices. Annex I contains requirements that apply to all classes of metering devices. The special requirements to the different classes of metering devices can be found in annexes MI-001 to MI-010. The remaining annexes A to H1 contain information about different conformity assessment procedures.

Modern metering devices are highly integrated embedded systems with device specific sensors for measuring current, flow of gas or water, or heat quantities. Some meters are also equipped with units for sending data via communication networks (PLC, GPRS/UMTS, DSL etc.).

Metering devices based on embedded systems are so called Smart Meters. Compared to conventional electricity meters, Smart Meters offer a large range of new functionalities. They enable frequent billing by using remote readout of measurement data. They make the current energy consumption visible to the user and they provide administrative access for the measuring point operator or the network operator. Furthermore smart meters can help to support smart grid mechanisms that will be used to automatically adjust the amount of produced power by distributed energy providers like block power stations to the actual demand.

Led by the liberalization of the European energy market, the Smart Meter sector is expected to be a growing market in the next years. Animating spirits for the use of Smart Meters in Germany is §40 of the German ENWG which states that every customer has the right to demand a monthly electricity bill. The trend of a growing market is further intensified by increasing fields of application for Smart Meters. One of the future uses could be the area of E-Mobility, generating a need for charging stations to large-scale operate electro mobiles.

The deployment of smart meters is additionally facilitated by the EU-Directive 2006/32/EC [8]. This directive states that "Member States shall ensure that [...] final customers [...] are provided with [...] meters that accurately reflect the final customer's actual energy consumption and that provide information on actual time of use".

3. STATE OF THE ART

Equipment approval. The engineering process and subsequent production of Smart Meters comprises multiple steps. The first step is the development and design of the hardware and software of the system. This step is usually followed by building a first prototype which is refined until the design is confirmed and the device is running without errors. The next step is the mass production of the new meter type followed by placing it on the market. Before the action of placing the meter on the market can take place, it has to be approved by a so called notified body. This is usually the Metrology Institute of the country itself (e. g. the Physical Technical Institute in Germany) or a person in law that is allowed to represent the Metrology Institute.

The Measuring Instruments Directive (MID) offers three different procedures to gain the obligatory type approval for a new metering device. These procedures are segmented into several modules. The first one consists of a combination of modules B and F. Module B requires a type examination at the prototype stadium. This type examination comprises a time consuming check of the whole system which is done by a notified body of the member state. This is followed by module F, the first basic calibration of every single device at the time of production. The basic calibration is also done by the notified body. As required by law, every Smart Meter has to pass the conformity declaration and tagging which is the last step.

Another option is offered by the combination of the modules B and D. Here the steps of module B are the same as described above. However, then module B is followed by module D which requires a Quality Management System for production. So the first basic calibration of every single meter is now part of the production line and no more performed by the notified body. As before, the last step is the conformity declaration and tagging.

The third and completely different type approval procedure is the use of module H1. Here, no type examination of any prototype is necessary. The main step is the check of the system design by the notified body before a prototype even exists. This step is followed by an overall Quality Management System including the phases of prototyping, mass production and first basic calibration. As for the other two type approval procedures the last step consists in conformity declaration and tagging.

Nowadays type approval of a new metering device in Germany is a very time consuming process because nearly all Smart Meter producers use the type approval procedures consisting of modules B+F or B+D. By now the notified bodies in Germany doing the type examination (module B) are completely overburdened because the individual type examination procedures cannot be automatized.
An intensified use of module H1 for type approval could unburden the notified bodies, provided a common design process for metering devices is developed. The actions of checking the device's conformity in regard to the security and dependability requirements of the MID could be automatized. However, a necessary condition is the formalization of the metering device requirements.
By now no overall design process exists especially addressing security and dependability requirements. This is caused by the fact that the engineers in companies developing metering devices are mostly experts for sensors and measurement techniques, but they are no experts in the field of security and dependability.

Formalization of Requirements. In this paper we introduce a formal representation of some MID security requirements. This constitutes a first step towards automated type approval of metering devices and overcomes the drawbacks of a verbal description of security requirements such as the possibility of misunderstandings and the lack of compari-

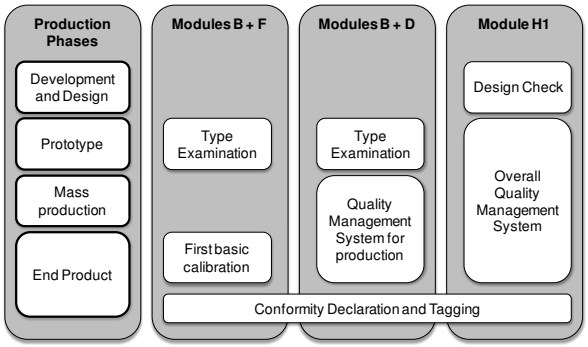

Figure 1: MID Assessment procedures

son and validation techniques. There exist many different approaches for security requirements formalization such as ISO 15408-2 Evaluation criteria for IT security – Part 2: Security functional requirements[11] or RFC4949 Internet Security Glossary[14]. However these two formalization approaches are based solely on textual descriptions of a security relevant vocabulary that – despite the advantage of consistent definitions among many people – does not remedy the drawbacks of textual descriptions. Further, there exists the class of BAN [3] and other authentication protocol logics or algebras [12]. Even though these approaches provide a formalism that can be used for comparison and validation, they rely on a big set of axiomatic deduction rules. This can be a drawback as it is impossible to verify the framework itself, rather, domain experts have to agree on its correctness. Finally, there exists the class of formal security frameworks that are based on formal language and automaton theory. Representatives of this domain, such as AVISPA [2], express most requirements and properties in terms of *safety and lifeness* [1] properties, or *non-interference*, respectively *hypersafety and hyperlifeness* properties [4]. The approach used for the rest of this paper – the Security Modelling Framework SeMF [9] – is based on the same foundation of formal language theory, though it uses more fine-grained concepts to model the requirements and properties of a system. Furthermore it does not primarily focus on an attack-based view of the system, but rather models the requirements as the assurances of a system that could be target of attacks.

4. THE SECURITY MODELLING FRAMEWORK SEMF

In this section we briefly introduce our Security Modeling Framework SeMF and, based on this, the formal definition of the security requirements that will be needed to exemplarily formalize some of the MID requirements.

The behaviour B of a discrete system S can be formally described by the set of its possible sequences of actions (traces). Therefore $B \subseteq \Sigma^*$ holds, where Σ (called the alphabet) is the set of all actions of the system, Σ^* is the set of all finite sequences (called words) of elements of Σ, including the empty sequence denoted by ε. Subsets of Σ^* are called formal languages. Words can be composed: if u and v are words, then uv is also a word. For a word $x \in \Sigma^*$, we denote the set of actions of x by $alph(x)$. For more details on the theory of formal languages we refer the reader to [5].

We further extend the system specification by two components: *agents' initial knowledges* about the global system behaviour and *agents' local views*. The initial knowledge $W_P \subseteq \Sigma^*$ of agent P about the system consists of all traces P initially considers possible, i.e. all traces that do not violate any of P's assumptions about the system. Every trace that is not explicitly forbidden can happen in the system. An agent P may assume for example that a message that was received must have been sent before. Thus the agent's W_P will contain only those sequences of actions in which a message is first sent and then received. Further we can assume $B \subseteq W_P$, as reasoning within SeMF primarily targets the validation and verification of security properties in terms of positive formulations, i.e. assurances the agents of the system may have. Other approaches that deal with malfunction, misassumptions and attacker models cannot rely on this assumption.

In a running system P can learn from actions that have occurred. Satisfaction of security properties obviously also depends on what agents are able to learn. After a sequence of actions $\omega \in B$ has happened, every agent P can use its *local view* λ_P – an alphabetic language homomorphism $\lambda_P : \Sigma^* \to \Sigma_P^*$ – of ω to determine the sequences of actions it considers to have possibly happened. Examples of an agent's local view are that an agent can see only its own actions, or that an agent P can see that an action $send(sender, message)$ occurred but cannot see the message, in which case $\lambda_P(send(sender, message)) = send(sender)$, or an agent may see a message on a network bus and is not able to determine the sender $\lambda_P(send(sender, message)) = send(message)$.

For a sequence of actions $\omega \in B$ and agent $P \in \mathbb{P}$ ($\mathbb{P}$ denoting the set of all agents), $\lambda_P^{-1}(\lambda_P(\omega)) \subseteq \Sigma^*$ is the set of all sequences that look exactly the same from P's local view after ω has happened. Depending on its knowledge about the system S, including underlying security mechanisms and system assumptions, P does not consider all sequences in $\lambda_P^{-1}(\lambda_P(\omega))$ possible. Thus it can use its initial knowledge to reduce this set: $\lambda_P^{-1}(\lambda_P(\omega)) \cap W_P$ describes all sequences of actions P considers to have possibly happened when ω has happened.

Security properties can now be defined in terms of the agents' initial knowledges and local views. In [10] we have introduced a variety of definitions of security properties (e.g. authenticity, proof of authenticity, confidentiality). For the formalization of the requirements to be discussed in Section 5 we need the concepts of authenticity and precedence of actions.

We call a particular action a authentic for an agent P if in all sequences that P considers to have possibly happened after a sequence of actions ω has happened, some time in the past a must have happened. By extending this definition to a set of actions Γ being authentic for P if one of the actions in Γ is authentic for P we gain the flexibility that P does not necessarily need to know all parameters of the authentic action. For example, a message may consist of one part protected by a digital signature and another irrelevant part without protection. Then, the recipient can know that the signer has authentically sent a message containing the signature, but the rest of the message is not authentic.

Therefore, in this case, Γ comprises all messages containing the relevant signature and arbitrary other message parts.

DEFINITION 1. *A set of actions $\Gamma \subseteq \Sigma$ is authentic for $P \in \mathbb{P}$ after a sequence of actions $\omega \in B$ with respect to W_P if $alph(x) \cap \Gamma \neq \emptyset$ for all $x \in \lambda_P^{-1}(\lambda_P(\omega)) \cap W_P$.*

We define the following instantiation of this property that states that whenever an action b has happened in a sequence of actions ω, it must be authentic for agent P that action a has happened as well. Note that in most cases, action b is in P's local view.

DEFINITION 2. *For a system S with behaviour $B \subseteq \Sigma^*$, agent $P \subseteq \mathbb{P}$, and actions $a, b \in \Sigma$, $auth(a, b, P)$ holds in S if for all $\omega \in B$, whenever $b \in alph(\omega)$, the action a is authentic for P.*

Finally we need to express that whenever an action b has happened, an action a must have happened before:

DEFINITION 3. *For a system with behaviour $B \subseteq \Sigma^*$, $a, b \in \Sigma$, $precede(a, b)$ holds if for all $\omega \in B$, if $b \in alph(\omega)$ then $a \in alph(\omega)$.*

Note that since the property needs to hold for all ω in B, it holds in particular for an ω with action b as the last action. Hence action a must happen before action b.

This property can be used for example to formalize the assumption that a message that is received by some agent must have been sent before.

5. REQUIREMENTS FORMALIZATION

In this section we introduce the formal model of an example system and, based on this, formalize and discuss some exemplary MID requirements.

5.1 Sample Requirement

The Measuring Instruments Directive (MID) of the European Union lists a couple of security requirements that address properties of the metering devices. We start our work on formalizing requirements with the following:

MID Requirement Annex I 8.4: *Measurement data, software that is critical for measurement characteristics and metrologically important parameters stored or transmitted shall be adequately protected against accidental or intentional corruption.*

This formulation leaves some space for interpretation. First, we note that while for stored data it might be possible to actually prohibit intentional corruption by means of some access control mechanism, this is usually not possible when it comes to communication over networks using technologies such as PLC, GPRS/UMTS, DSL, etc. It must always be assumed that the messages can be manipulated, blocked or destroyed. So what is presumably meant to be required is that it must be possible to detect any kind of manipulation. Having this in mind, we conclude that the above requirement can be reformulated to:

1. Authenticity and integrity of measurement data, of the software that is critical for measurement characteristics, and of important parameters *stored* on the device have to be ensured.
2. Authenticity and integrity of measurement data, of the software that is critical for measurement characteristics, and of important parameters, when *transmitted* over communication networks have to be ensured.

In the following we will address these requirements formally.

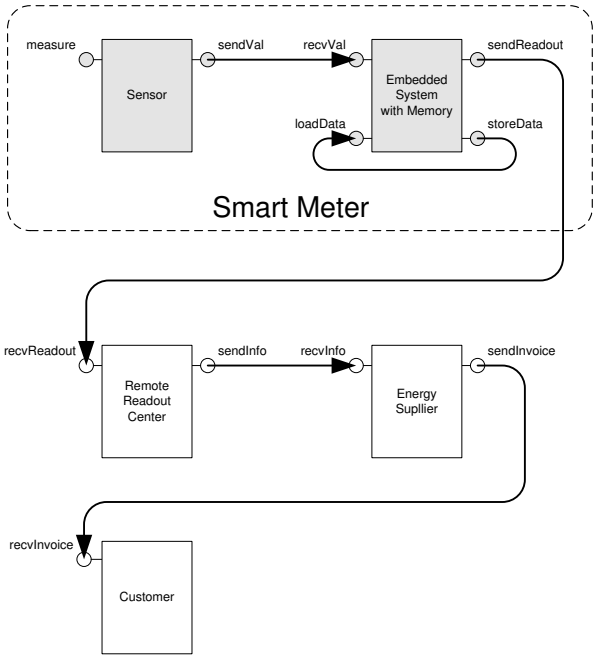

Figure 2: System model

5.2 The Example System Modelled Formally

Our example system consists of two Smart Meters, each with a Sensor, an Embedded System and equipped with some memory. Hence we identify as agents acting in the system the sensors $Sensor_i$ and the embedded systems ES_i ($i \in \{1, 2\}$). Further agents of the system are one Remote Readout Center RRC gathering the readouts from many Smart Meters, one Energy Supplier $Suppl$ selling energy to Customers, and one Customer C.

For formalizing the actions depicted in Figure 2, for simplicity we abstract from the exact way that Embedded System, Remote Readout Center, etc., process the value measured by the sensor. Hence we use the following actions:

$measure(Sensor_i, v_j, t_k)$ Sensor$_i$ measures value v_j in the time interval t_k.

$sendVal(Sensor_i, ES_l, v_j, t_k)$ Sensor$_i$ sends the measured measurement value and the measurement time interval to the Embedded System ES$_l$.

$recvVal(ES_i, Sensor_l, v_j, t_k)$ Embedded System ES_i receives some value, presumably from Sensor$_l$, that was presumably measured in time interval t_k.

$storeData(ES_i, Mem_i, v_j, t_k)$ The Embedded System stores some value and its presumed measurement time interval in its memory.

$loadData(ES_i, Mem_i, v_j, t_k)$ The Embedded System loads some value and its presumed measurement time interval from its memory.

$sendReadout(ES_i, RRC, v_j, t_k)$ The Embedded System sends some value and its presumed measurement time interval to the Remote Readout Center (RRC).

$recvReadout(RRC, ES_i, v_j, t_k)$ The Remote Readout Center receives some value and its presumed measurement time interval from an Embedded System, presumably from ES$_i$.

$sendInfo(RRC, Suppl, v_j, t_k)$ The Remote Readout Center

sends some value and its presumed measurement time interval to the Energy Supplier.

recvInfo$(Suppl, RRC, v_j, t_k)$ The Energy Supplier receives some value and its presumed measurement time interval, presumably from the Remote Readout Center.

sendInvoice$(Suppl, C, invoice(v_j, t_k))$ The Energy Supplier sends the invoice based on the value and its presumed measurement time interval to the Customer.

recvInvoice$(C, Suppl, invoice(v_j, t_k))$ The Customer receives the invoice of some value and its presumed measurement interval time, presumably sent by the Supplier.

It is important to note that the recipient of a message is not necessarily able to know its actual sender. The agent named in the receiving action as the sender is only the one that *presumably* has sent the message, it can be viewed as part of the message itself or as part of some underlying transport protocol. Note also that by differentiating between sending actions of different agents (e.g. sendReadout vs. sendInfo) we implicitly model that in a real system these actions cannot be confounded. This corresponds to the different communication techniques (PLC, GPRS/UMTS, DSL, etc.) that can be used in the metering domain. However, usage of the same network for specific send and receive actions should be reflected in identical action names.

5.3 Requirements Formalized

We start with the requirement regarding transmission of data between Sensor and Embedded System. MID requires transmitted data to be "adequately protected against accidental or intentional corruption". In terms of the formal framework SeMF, this is an authenticity requirement. Each time the Embedded System ES_l receives some data and time interval, presumably from a specific Sensor, it must be authentic for the Embedded System that this data and time interval was indeed sent by the Sensor. In other words, in all action sequences that ES_l considers to have possibly happened, according to its local view and initial knowledge, Sensor_i must have sent this measurement data and time interval.

$$auth(sendVal(Sensor_i, ES_l, v_j, t_k), recvVal(ES_l, Sensor_i, v_j, t_k), ES_l) \quad (1)$$

Equivalent authenticity requirements can be derived for the transmission of value and time interval from Embedded System to RRC, from RRC to the Supplier, and finally from the Supplier to the Customer, thus addressing the above required protection of transmission.

MID further requires stored data to be "adequately protected against accidental or intentional corruption". This leads again to an authenticity requirement stating that each time the Embedded System loads some data together with a measurement time interval, it must have stored this data and time interval before:

$$auth(storeData(ES_i, Mem_l, v_j, t_k), loadData(ES_i, Mem_l, v_j, t_k), ES_i) \quad (2)$$

A system that satisfies the above security requirements can still not be assumed to provide the desired results. What we want such a system to provide is for example that whenever measurement value is sent, received or stored, the sensor indeed measured this value in the named time interval. MID provides a couple of requirements addressing hardware components, e.g. requirement 8.2:

A hardware component that is critical for metrological characteristics shall be designed so that it can be secured. Security measures foreseen shall provide for evidence of an intervention.

However, it is not clear whether any of them was really meant to cover what is needed. In the SeMF framework we can formulate a security requirement that explicitly states that whenever the sensor sends a measurement value together with a time interval, it must have measured this value within this time interval:

$$precede(measure(Sensor_i, v_j, t_k), sendVal(Sensor_i, ES_l, v_j, t_k)) \quad (3)$$

In a system satisfying this requirement, a sensor will never send measurement data and time interval without ever having measured the respective value in this time interval. Explicitly formulating this requirement supports the development of a concrete system since it emphasizes the necessity to verify whether or not the requirement is satisfied. If the sensor is a simple device with restricted capacities, it just measures and automatically transmits the measurement to some fixed recipient. In this case the security requirement can be assumed to be satisfied. If however the sensor has more intelligence, additional actions have to be taken in order to ensure that the sensor acts correctly.

In the same manner we can address the correct behaviour of the Embedded System, i.e. the correct functioning of the Embedded System's software by requiring the following two properties to hold:

$$precede(recvValue(ES_i, Sensor_l, v_j, t_k), storeData(ES_i, Mem_l, v_j, t_k)) \quad (4)$$

$$precede(loadData(ES_i, Mem_l, v_j, t_k) sendReadout(ES_i, RRC, v_j, t_k)) \quad (5)$$

This indirectly addresses another MID requirement:

Software that is critical for metrological characteristics shall be identified as such and shall be secured.

In order to ensure the respective properties, measures must be taken that the initial ES_i software that we assume to provide these properties can not be manipulated unnoticed.

We can proceed and specify equivalent precedence requirements in order to demand the correct behaviour of RRC, Supplier, etc. A system that meets all these requirements still does not provide all that is needed, because so far we have not taken into account the requirements of agents involved in the system. The Customer for example has the requirement that each time he/she receives an invoice, the amount of used electricity named in v_j and the time interval named in t_k correspond to what the Sensor of his/her Smart Meter actually measured in the named time interval. Hence in order to address all of the properties we want the overall system to provide, the MID requirements have to be seen in a larger context that also takes these personal views on the system into account.

In order to adequately model the requirements of the agents involved in the system, we need to formulate trust requirements and assumptions, respectively. Since the Customer for example has no means to actually verify that the

Sensor works correctly, he/she must trust into the sensor's correctness. We can for example assume the Customer to trust into the Supplier's trust in that the Sensor works correctly, and further assume mechanisms in place that allow the Supplier to verify the Sensor's correctness, thus substantiating the Supplier's trust.

Our security framework SeMF provides an adequate formal definition of trust of an agent into a property holding in a system. It further provides formally proven relations between security properties such as authenticity, precedence, confidentiality, and trust. Applying this framework allows to precisely specify the security requirements that need to hold for a metering system, thus substantiating and stating more precisely security requirements formulated in documents such as the European Measurements Instruments Directive.

6. CONCLUSIONS

In this paper we have introduced the formal model of a system for transmission of measurement data in the metering domain in terms of our Security Modeling Framework SeMF. We have further interpreted the verbal description of some MID requirements addressing security aspects of metering devices and then formalized these requirements within our formal model. Our results substantiate the possibility to formalize verbal descriptions of lawful security and dependability requirements for metering devices by using the Secure Modelling Framework SeMF. They further highlight the importance of such formalization, as some of the MID requirements were found to leave room for misinterpretation.

One of the main challenges is to correctly interpret the verbal descriptions of the MID. It is very important to reflect what the requirement should presumably express. This also includes to think about the security properties addressed. The process of reflection and interpretation of the verbally described requirements induces the effect that eventually existing security flaws in the lawful texts can be found.

When regarding the growing need for Smart Meters and the necessary assessment procedures to have them put into market, formalized requirements can help to automatically verify new meter designs against the existing lawful requirements. The time consuming process of assessment could be automatized, thus saving time and money. Further, other domains like the automotive or aeronautic industry can benefit from the possibility to formalize verbally described requirements based on laws or standards like ISO or IEEE in other domains.

The results presented in this paper are just a first step towards an automated type approval of metering devices. Future work will include to add formal specifications of trust relations in order to capture trust assumptions that must hold in order for a metering system to provide the desired security properties.

7. REFERENCES

[1] B. Alpern and F. B. Schneider. Defining liveness. *Information Processing Letters*, 21(4):181–185, 7 Oct. 1985.

[2] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuéllar, P. Drielsma, P. Heám, O. Kouchnarenko, J. Mantovani, et al. The AVISPA tool for the automated validation of internet security protocols and applications. In *Computer Aided Verification*, volume 3576 of *LNCS*, pages 281–285. Springer, 2005.

[3] M. Burrows, M. Abadi, and R. Needham. A Logic of Authentication. *ACM Transactions on Computer Systems*, 8, 1990.

[4] M. Clarkson and F. Schneider. Hyperproperties. In *Proceedings of the 2008 21st IEEE Computer Security Foundations Symposium*, pages 51–65. IEEE Computer Society, 2008.

[5] S. Eilenberg. *Automata, Languages and Machines.* Academic Press, New York, 1974.

[6] European Council. Council directive 76/891/eec of 4 november 1976 on the approximation of the laws of the member states relating to electrical energy meters, 11 1976.

[7] European Parliament. Directive 2004/22/ec of the european parliament and of the council of 31 march 2004 on measuring instruments. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?-uri=OJ:L:2004:135:0001:0080:EN:PDF, 3 2004.

[8] European Parliament. Directive 2006/32/ec of the european parliament and of the council of 5 april 2006 on energy end-use efficiency and energy services and repealing council directive 93/76/eec. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?-uri=OJ:L:2006:114:0064:0064:EN:PDF, 4 2006.

[9] A. Fuchs, S. Gürgens, and C. Rudolph. A Formal Notion of Trust – Enabling Reasoning about Security Properties. In M. Nishigaki, A. Josang, Y. Murayama, and S. Marsh, editors, *Trust Management IV: 4th IFIP WG 11.11 International Conference, IFIPTM 2010, Morioka, Japan, June 16-18, 2010, Proceedings*, pages 200–215. Springer-Verlag GmbH, 2010.

[10] S. Gürgens, P. Ochsenschläger, and C. Rudolph. On a formal framework for security properties. *International Computer Standards & Interface Journal (CSI), Special issue on formal methods, techniques and tools for secure and reliable applications*, 27(5):457–466, June 2005.

[11] ISO/IEC. Information technology – security techniques – evaluation criteria for it security – part 2: Security functional requirements. ISO/IEC 15408-2, Oct. 2005.

[12] U. Maurer. Abstraction in cryptography. In S. Halevi, editor, *Advances in Cryptology — CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, page 459. Springer-Verlag, Aug. 2009.

[13] Physical-Technical Federal Institute of Germany (PTB). Requirements for software-based metering devices. http://www.ptb.de.

[14] R. Shirey. Internet Security Glossary, Version 2. RFC 4949 (Informational), Aug. 2007.

[15] TERESA consortium. Teresa project. http://www.teresa-project.org, 11 2009.