Interoperable device Identification in Smart-Grid Environments

Nicolai Kuntze*, Carsten Rudolph*, Ingo Bente†, Joerg Vieweg† and Josef von Helden†
*Fraunhofer Institute for Secure Information Technology (SIT),
Rheinstrasse 75, 64295 Darmstadt, Germany
Email: {nicolai.kuntze|carsten.rudolph}@sit.fraunhofer.de
†Fachhochschule Hannover
University of Applied Sciences and Arts, Hannover, Germany
Email: {ingo.bente|joerg.vieweg|josef.vonhelden}@fh-hannover.de

***Abstract*—Concepts for future energy networks envision the distribution of measurement and control infrastructures to the customers to allow for improved reactions to certain events in the energy grid and to ease the measurement process. Such a distribution of control functionalities requires a corresponding device on the customer side that performs or mediates between the energy grid requirements and the customer infrastructure. These Smart Energy Gateways (SEGs) are owned by the energy network operator and but enforce the contracts between network operator and customer for both parties. They can also support additional value-added services. Due to the different uses, SEGs will be exposed to more and other attacks than current end-user devices such as DSL or WLAN routers. The impact of attacks to SEGs is also a peril to the overall operation of the energy grid.**

This paper provides an approach to the reliable identification of SEGs based on already established industry standards, namely Trusted Computing and Trusted Network Connect.

I. Introduction

The growing integration of information and communication technology (ICT) with international energy infrastructures is recently shaping the overall vision of converging energy and information networks to a new infrastructure called the *Smart Grid*. The Smart Grid describes the vision and enhancement strategy to address shortcomings of today's energy networks, e.g. reliability and environmental sustainability of supply, fine-grained control of transmission and distribution of electricity or enhanced interconnection with distributed and alternative power sources. It basically envisions the integration of power transmission and power distribution elements with information and communication infrastructures. I.e. the Smart Grid vision includes electricity systems characterized by a two-way flow of electricity and information across distributed generation, distribution and operation resources and more reliable energy supply. In a Smart Grid demand peaks can be reduced through intelligent demand response and demand side management, for instance by a modulation of customer's energy demand and usage based on flexible pricing schemes and the management of assets on customer's side. Additional detailed information on the status of the energy grid will also enable quick and efficient reaction to problems in the grid and therefore lower operating costs for grid operators. Further, these capabilities additionally can provide incentives increasing customers' interest to automatically manage their consumption patterns and to take opportunities in de-regulated energy markets.

One of the key components of Smart Grids is Demand Side Management (DSM) [16] (in conjunction with Smart Metering and Home Area Networks) that will help the end consumers to be more aware of their energy consumption behaviour and control the in-home appliances in an easier way to reduce their energy usage. Additionally, energy network operators can influence the demand on the customer side either based on incentives with respect to the load situation or by directly controlling the status of particular appliances such as control of air condition or loading and unloading electrical vehicles. For these purposes, the so called Smart Energy Gateway (SEG) has been widely proposed [7], [12] to enable communications and interactions between in-home appliances, electrical vehicles and SEGs as well as between the SEG and higher level instances of the power grid. SEG and their use cases deal with sensitive data about consumers like their consumption profiles. Consequently, security and privacy issues have to be considered cautiously. Secure identification and authentication is one essential factor. In-home appliances need to be identified by the SEG and also identification and authenticity of the SEG itself towards higher instances are required as the foundation for secure communication and dependable control structures. Value-added services based on the toehold established by the SEGs are also a promising business case for the energy grid operators. Third-party applications running on the SEG can provide a variety of services to the customers.

As discussed in [8] dependability and security issues in Smart Grids from the perspective of the SEG can be derived from the control requirements for Smart Grids and the operation of the SEGs in malicious environments. The distribution of control functionalities into the physical vicinity of the customer introduces new opportunities as introduced above but also allows for new (cyber) attack vectors previously unknown to the energy sector. The cornerstone for all trust relations to the individual SEG is the secure and reliable authentication of the individual SEG to allow for authorisation decisions in the various work flows a SEG is involved in (as for example control systems or real time measurements of the energy grid load). Each SEG has to proof its identity towards the energy

grid on the one hand and to the attached components in the vicinity of the end customer on the other as the SEG mediates between the energy grid and the customer environments.

The energy grid operator establishes by the deployment of the SEG a toehold at the side of the customer to allow for certain metering and control functionalities interfering with the customers behaviour but also providing for certain value added services. Aside the well know metering of consumption and supply of energy, control issues arise to allow for energy-efficient demand control. This demand control is based on the distribution of control messages to the customer. On the side of the customer the control messages have to be interpreted by the SEG and transferred to commands transmitted to the attached devices like microCHPs [3] or the air conditioning.

Identification of devices is done in many cases based on MAC- or IP-addresses or simple software tokens as used e.g. in Kerberos [11] approaches. Those current mechanisms are insufficient for the reliable authentication of SEGs as they are not using a strong identity of a device that is protected against forgery but an arbitrarily assigned address. This address might need to be changed under several circumstances. Besides that, even if no change is intended, this address-based identification approach is easy to spoof and therefore not secure enough. Those issues render this idea unusable within a smart grid scenario.

This paper focuses on the identification and authorisation of SEGs and proposes to use the capabilities of the Trusted Platform Module (TPM) in conjunction with Trusted Network Connect (TNC) to provide interoperable and secure device authentication in smart grid environments. The presented approach introduces a special Integrity Measurement Collector/Verifier (IMC/IMV) that handles device authentication. During a normal TNC handshake, the device authentication is done via a challenge response protocol that involves the signature of a non-migratable asymmetric TPM signing key deployed on the SEG. Such a key is generated by the TPM itself and the private part will never leave the TPM in clear text. By using standardized protocols, the approach ensures maximum interoperability. Furthermore, it is far more secure than existing mechanisms for device authentication due to the hardware protection and security capabilities of the TPM.

Sections 2 introduces Trusted Computing, Section 3 gives a short overview about Trusted Network Connect. In Section 4, potential problems that can arise when using current device identification and authentication approaches for SEGs are emphasized and the concept of using a non-migratable, TPM-protected key within Trusted Network Connect for device authentication is presented. Additionally the security of this contribution compared against common security issues is discussed. The implementation of the approach is discussed in Section 5. The paper concludes in Section 6 sketching out planned next steps towards hardware extended device authentication for critical infrastructures.

II. Trusted Computing in a Nutshell

Trusted Computing technology [10] as defined by the Trusted Computing Group is a technology implementing consistently behaving computer systems. This consistent behaviour is enforced by providing methods for reliably checking a system's integrity and identifying anomalous and/or unwanted characteristics. These methods depict a trusted system's base of trust and thus are implemented in hardware, as it is less susceptible for attacks than software pendants.

To successfully realize stringently reliable modules, several cryptographic mechanisms are implemented on a hardware chip, namely Trusted Platform Module (TPM). This chip incorporates strong asymmetric key cryptography, cryptographic hash functions and a random number generator, that is capable of producing true random numbers instead of pseudo random ones. Additionally each trusted system is equipped with a unique key pair whose private key is securely and irrevocably stored inside the chip. The chip itself is the only entity to read and use this key for e.g. signing or encryption. This concept builds a powerful basement for approving and establishing system integrity since it allows to truly trustworthy let a trusted system sign data and to securely encrypt data for one specific trusted system. This is commonly used to measure system integrity and to ensure a system is and remains in a predictable and trustworthy state that produces only accurate results.

A. Trust for Measurement

The key concept of Trusted Computing is the establishing and extension of trust from an initially trusted security anchor up to further used components of a system while boot-up. Each component loaded while booting up the system is measured before execution by computing a SHA-1 digest value of it. The first component of this cycle acts as security anchor and has to be initially trusted, since it's integrity is not measured. This anchor is called *Core Root of Trust for Measurement* (CRTM) and is implemented as Firmware extension. It is executed after the very start of a system before any other Firmware code thus enabling to measure the Firmware and the platform's firmware. Each subsequent component involved in the boot-up process thereupon measures its successive component. Each measurement is stored in *Platform Configuration Registers* (PCR) on the TPM chip. These 160-bit registers are in the volatile storage on the chip and can exclusively be updated by calling the TPM command `TPM_EXTEND`. This command includes the old value of a register in the calculation of its new value thus preventing manipulation of registers.

$$\mathrm{PCR}_i = \text{SHA-1}(\mathrm{PCR}_i \mid \text{new value}) \tag{1}$$

This basically implements a non-commutative one-way function preventing from deleting and/or overwriting digest values in a PCR and enabling tracking of the chronological sequence values were applied to the register. This allows to analyze a system's state and furthermore prove its integrity by verifying integrity of any component loaded upon boot-up. This type of boot-up is also called *Trusted Boot Process*.

Moreover the successive process of extending trust with each measure is commonly referenced to build up a *Chain of Trust*. To reproduce and verify a platform register's value in hindsight, every `TPM_EXTEND` command executed has to be tracked in a log. In the case of runtime measurement this has to be done by the operating system [14] resulting in a log called *Stored Measurement Log* (SML). Since PCRs are located in the volatile storage of the TPM chip, each PCR is initialized with zeros upon system start and from there on is filled with measured data.

B. Trust for Reporting

Another main concept of Trusted Computing is Remote Attestation, a process to prove trustworthiness of a Trusted Platform to an external party. Similar concepts are also covered in [9] in the domain of smart meters. To verify a platform's integrity, a subset of PCRs together with the above-mentioned SML is sent to the external party. The PCRs values are then re-calculated using the chronological order of measured components logged in the SML and a software implementation of the `TPM_EXTEND` command. In order to ensure integrity of the submitted PCR subset, it is signed by a unique TPM key pair, the *Attestation Identity Key* (AIK) which is created on the chip. Its private key is furthermore secured from being read from outside the chip thus representing the so-called *Root of Trust for Reporting*.

C. Trust for Storage

TPM chips are equipped with several cryptographic modules providing access to en-/decryption, hashing and key generation. This allows to securely generate cryptographic keys inside the TPM. Therefore a *RSA key pair generator* makes use of a *True Random Number Generator* that is capable of producing true random numbers hence generating true random RSA key pairs. These are thereupon stored outside the chip in a shielded storage. This storage is protected using a hierarchical encryption structure. Each private key of a generated key pair is encrypted with a parent key. The root of this tree-like key structure is represented by the last of three security anchors, the *Storage Root Key* (SRK). The SRK is a 2048-bit RSA key pair, that is created on the chip while setting up the TPM for a new owner. This is done using the `TPM_TakeOwnership` command. Like the EK it is unmodifiable and stored in the non-volatile storage on the chip, restricting the private key from being read from outside the chip. The SRK represents the *Root of Trust for Storage* (RTS) since it is used to securely store data and other keys outside the chip.

D. AIK Certification

Since each TPM is globally unique and thus identifiable and traceable, privacy issues arise when attesting a platform's state to external parties using Remote Attestation. In order to avoid this security issue, TPM chips provide for pseudonymity by allowing to generate temporary keys for attestation. These *Attestation Identity Keys* (AIK) can be created at any time using the `TPM_MakeIdentity` command and may be certified by a *Trusted Third Party* (TTP) to allow external parties to verify, that an AIK belongs to a TCG conform platform. AIKs can only be associated to their platform's EK by the TTP thus providing the platform with pseudonymity towards other entities. To issue an AIK credential, the platform has to send the EK-signed public key of a generated AIK key pair together with several credentials declaring the platform's TCG conformance to the TTP. After successful verification of the AIK and the platform's credentials, a particular data structure is sent to the platform. This structure contains the AIK credential and can be securely loaded only into the TPM that signed the initial request using the `TPM_ActivateIdentity` command.

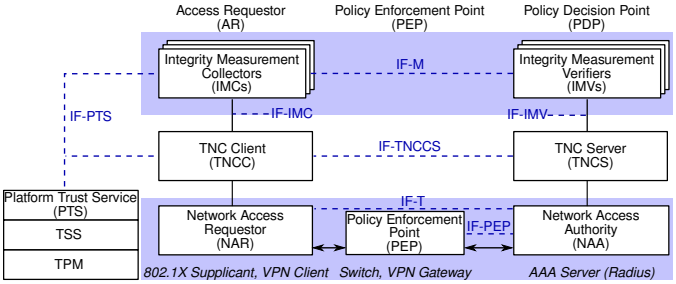

Fig. 1. Simplified TNC Architecture [5].

III. Trusted Network Connect in a Nutshell

Network Access Control (NAC) [2] approaches like TNC control to what extent endpoints are allowed to access a protected network. By defining policies, network operators specify preconditions that endpoints must fulfil before network access is granted. Those policies primarily consist of entries addressing the integrity state of an endpoint, like the installed Operating System, its Patch Level or the presence of security tools like Anti Virus Software or Personal Firewalls. Furthermore, policies can include entries for users, their associated roles within an enterprise and the endpoint itself, stating further requirements that must be fulfilled. An enterprise that uses a NAC solution could enforce that any endpoint that wishes to connect to the corporate network must have a specific Anti Virus software running with virus signatures that are up-to-date. Special software components are responsible for gathering necessary data reflecting the integrity state of an endpoint and for communicating this data to the protected network. This process is referred to as *assessment* in context of TNC. If integrity checks fail, several access restrictions can be enforced. In addition to being an open standard, TNC distinguishes itself from other NAC solutions by leveraging the capabilities of Trusted Platforms as proposed by the TCG [4]. In contrast to simple reporting TNC uses the TPM to provide cryptographically signed reports, a process called *Remote Attestation*. Thus, TNC can be considered as an interoperable instantiation of Remote Attestation.

A simplified version of the TNC architecture is depicted in figure 1. It consists of three entities:

1) An Access Requestor (AR) depicted on the left side. It represents an endpoint that wants to get access to a TNC

protected network.

2) A Policy Decision Point (PDP) depicted on the right side. It is located within the protected network. The PDP is responsible for deriving an access decision based upon the AR's current integrity state.
3) The entity in the middle is called Policy Enforcement Point (PEP), usually an edge switch or a VPN Gateway enforcing the access decision made by the PDP.

Both the AR and the PDP consist of multiple software components. Each one of them performs a specific task within the TNC framework. Important for the further discussion are especially the Integrity Measurement Collectors (IMCs) on the AR and the Integrity Measurement Verifiers (IMVs) on the PDP. As their names imply, IMCs are responsible for gathering integrity data on the AR during the assessment phase. By using the other components of the TNC framework, the gathered data is communicated to the IMVs on the PDP where it is evaluated against a policy specified by the operator of the TNC protected network. The IF-T protocol of the TNC framework (see figure 1) provides an authenticated channel that cryptographically protects all data that is communicated between the AR and the PDP, normally by using SSL/TLS. Optionally, the capabilities of Trusted Platforms can be used during the TNC assessment. Especially IMCs are able to request TPM based measurements of system components (like requesting a TPM_quote command that outputs a signature for certain Platform Configuration Registers [10]) via the IF-PTS interface.

IV. SEG Authentication in Smart Grids

Current device identification mechanisms commonly rely either on an address or some kind of credential. As an address is not necessarily bound to the real device identity and may change under certain circumstances (intentionally or not), it is not usable for the authentication of a SEG. Credentials like certificates may be a good approach, as long as they can be strongly bound to the SEG. Such a strong relation requires a sufficiently high protection of the identity and its relation to the system. Trusted Computing, in particular the Trusted Platform Module and the related deployment concepts defined by the Trusted Computing Group provide for one example of technology. If credentials are not strongly bound to the SEG they may be copied onto another, possible rogue SEG. This rogue SEG would then be able to use them for malicious activities.

Some work has already be done in the area of using Trusted Network Connect for the mentioned device identification purposes. [15] proposes a two step approach, part of the so called Trustcube. The Trustcube infrastructure is an approach which defines an infrastructure for measuring properties of a client. It enables arbitrary service providers to base decisions concerning this clients on the measured values. Besides platform properties of the client, measurement include also the identity of the platform itself as well as the user. The verification of the platforms identity is done via TNC and consists of two steps: First, a registration phase, which runs offline and only happens once, followed by an authentication phase which happens each time the identity of a platform is verified. While this approach sounds similar to ours, Trustcube includes the user into the authentication process and thus cannot only rely on the platforms identity. Additionally, it uses biometric information of the user, like fingerprints or eye retina scans. As an SEG does not has the owner directly operating the device an approach needs to rely on hardware features alone. One possible solution hereby is to use a non-migratable TPM key, which identifies the device and is independent from a particular user. Such a key can not be transferred between two TPMs and can therefore be used as an identity for the SEG.

In the proposed scenario of a SEG being the interface between the Smart Grid and the home networks, the in-home appliances could be considered the access requestors which need to be authenticated at SEG. The SEG is a PDP regarding connections to appliances and at the same time an AR while facing the data collector which is then the PDP.

As already stated, a secure and interoperable device authentication is required for SEGs. The presented approach uses Trusted Computing mechanisms to achieve this goal by using of a special IMC/IMV-pair that leverages the capabilities of Trusted Platforms within the context of Trusted Network Connect in order to authenticate such SEGs. While such a SEG connects to a high-level infrastructure of the smart grid, provided by the network operator, this special IMC/IMV-pair ensures the identity of the SEG. This process is done within the TNC handshake performed between the SEG and a verifier provided and controlled by the network operator. Besides the identity of the device, there may be more measurements taking place as part of this handshake. The idea of the identification itself is based upon the usage of a non migratable TPM signing key, which is bound to the platform (the SEG). The private part (secret key) of this key is stored on the SEG's TPM and cannot be retrieved from it. The public part will be used for verification purposes. The general procedure is to setup the SEG within an initial first step (creation of the key etc.) and to perform the authentication within the second step. The first step needs to be done only once while the second step will be performed each time a SEG authentication is necessary. In the first step, the key will be created and the public part is stored in a certificate. This certificate is made available to the IMV on the verifiers side. When the second step is carried out, the IMV sends a challenge to the IMC which will be signed using the key stored in the TPM. The IMV is able to verify the signatures correctness. In case it matches the expected values, the platform where the IMC is running on must be known. This implies that the connecting SEG is also known. As the key is generated by using the TPM, attackers are unable to retrieve the private part of it thus making it impossible to use it on another device, i.e. it is impossible to copy it onto another SEG.

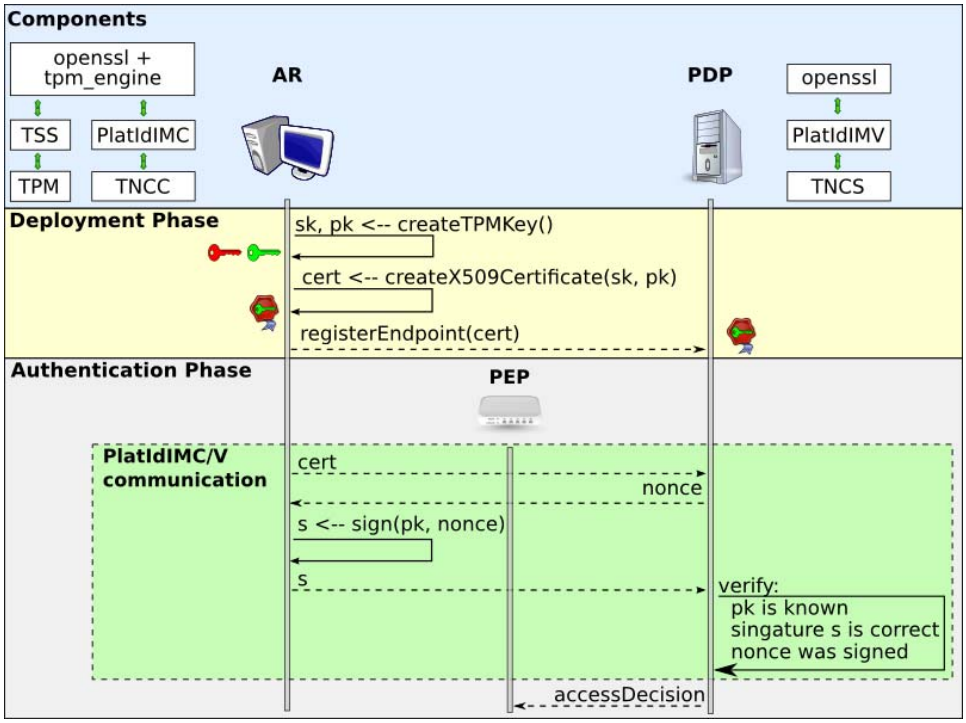

Fig. 2. Platform Authentication Protocol.

A. Protocol

The proposed IMC/IMV pair uses a simple protocol to exchange and verify the needed authentication data. Figure 2 gives an overview about this protocol. The protocol is divided into the two separate phases of the deployment phase and the authentication phase.

a) Deployment Phase: The deployment phase realises the initial setup of the platform. The phase runs solely offline and should be done when setting up the SEG. The phase consists of the following two steps:

1) Creation of a non-migratable key and an appropriate certificate on the platform which is to be authenticated later and
2) registration of this certificate within the key infrastructure of the Smart Grid resp. the Policy Decision Point.

The first step, creation of the key, is performed on the platform itself. The capabilities of the TPM are used to do this, resulting in a key where the private part is bound to the TPM of this and only this platform. That is, using software mechanisms (i.e. not tampering the hardware itself) one is unable to use the private part of this key on another platform. This is ensured by using a non-migratable TPM-key. Besides that, the public part of the generated key can be retrieved from the TPM. This feature is used to generate a X.509 certificate [6], which may optionally be signed by some trusted party (e.g. the Smart Grid Operator). The certificate is used to include the key into an arbitrary key infrastructure on the network operators side. The key can then be used later to authenticate the platform while the certificate holds information of the device, i.e. the SEG, trying to authenticate, which is done in the second phase.

b) Authentication Phase: This phase runs online, i.e. a connection to the Network Operators PDP and the key infrastructure is needed. The goal is to ensure, that the SEG trying to connect to the network owns an appropriate (and thereby well-known) key. That is, this phase ensures that (1) the identity of the connecting device is known by the network and (2) that the device is allowed to connect to the network. Four steps are carried out to achieve this:

1) The IMC on the Access Requestor sends the certificate, which was build within the deployment phase, to the IMV on the Policy Decision Point. Although the PDP already possesses the certificate, this step is needed to identify the Access Requestor. The proof of this identification is done in one of the following steps.
2) After receiving the certificate from the Access Requestor, the PDP now sends a challenge to the AR. For this challenge, a nonce is used to prevent replay attacks.
3) The IMC on the Access Requestor now signs the received nonce by using the secret key. This key was generated within the deployment phase and is stored within the TPM. By using a non-migratable key, it is ensured that only this platform (i.e. only the TPM on this platform) can use it for signing purposes. The signed nonce is then transmitted to the IMV on the Policy Decision Point.
4) The PDP can now verify the signature by using the public key (which was included in the certificate). If the signature is correct, the AR must possess the secret key-part of the certificate. As the secret key is bound to the TPM, it must be the platform the certificate belongs

to. This means by having a correct signature, the identity of the Access Requestor is verified. The PDP can now issue an access decision based on the verification results.

V. Implementation

We have successfully implemented the approach described in section IV. The DeviceIdIMC/V pair that handles the device authentication is based upon TNC@FHH's framework[1]. The OpenSSL TPM Engine, that encapsulates the communication with the TPM, is part of the TrouSerS project[2] and was only slightly modified.

During the deployment phase, the device must be registered on the PDP. In the experimental implementation, this process consists of three steps:

1) A non-migratable TPM signing key is created on the device by using the OpenSSL TPM Engine's createKey command. The command writes the created key as TSS_KEY_BLOB to a file.
2) In order to obtain a X.509 certificate for the non-migratable key, the default openssl command line tool can be used. The only requirement is to mention the OpenSSL TPM Engine appropriately via the options `-keyform engine -engine tpm` during the creation of the certificate request.
3) The signed certificate must be registered on the PDP. In our experimental implementation, this is done manually by just copying the certificate file and adjusting the DeviceIdIMV configuration accordingly.

This process assumes that one trustworthy stakeholder controls both the PDP and the corresponding device. Otherwise, there would be no assurance that the certificate belongs to a non-migratable TPM signing key.

Subsequently, the authentication phase can take place. The implementation uses wpa_supplicant as TNC client and TNC@FHH as TNC server. But since the IMC/V pair complies to TCG IF-IMC/V specification, any other third party software that supports those interfaces could be used as well. The device authentication is carried out during an ordinary TNC handshake. The message flow that takes place is depicted in figure 3.

In step 1, all necessary components are initialized during the startup of the TNC client and the TNC server. This includes the parsing of the IMC/V configuration files and the initialization of the OpenSSL TPM Engine.

At the beginning of a new TNC handshake, the TNC client notifies the DeviceIdIMC via the `beginHandshake()` function (step 2). This causes the DeviceIdIMC to forward the device certificate (obtained in the deployment phase) via the `sendMessage()` function to the TNC client. The TNC client transmits the certificate over the network to the TNC server. The network traffic is encapsulated according to the IF-T protocol.

On the PDP, the TNC server forwards the received certificate via the `receiveMessage()` function to the DeviceIdIMV (step 4), which then generates a nonce (step 5) by using OpenSSL. Note that there is no OpenSSL TPM Engine, TSS or TPM on the PDP. This nonce is forwarded to the TNC server via the `sendMessage()` (step 6), transported over the network to the TNC client, which forwards it to the DeviceIdIMC via the `receiveMessage()` function (step 7).

The DeviceIdIMC then uses the OpenSSL TPM Engine to sign the received nonce (step 8). For the IMC, this is a simple call to a standard OpenSSL function[3]. The OpenSSL TPM Engine handles all the necessary communication with the TSS (and the TPM respectively) in order to use the TPM protected signing key that was obtained during the deployment phase.

Next, the signature is communicated to the DeviceIdIMV (step 9 and 10) via the same functions as in steps 3 and 4. In step 11, the IMV uses OpenSSL to verify the signature. If the signature is valid and if the corresponding public key is known, the client was successfully authenticated (which means that the client has been properly registered during the deployment phase). In this case, the IMV provides an ALLOW recommendation to the TNC server, stating that network access should be granted.

VI. Conclusion

Secure identification of devices is one essential aspect within the envisioned smart grid environments. Without such identification as a step to establish a trust relation to the SEG, the concept of smart grids based on distributed control requires physically enforced means to establish the trust relation. Such a physical protection seems to be infeasible in the context of the different communication interfaces envisioned for the SEGs. In this paper, an approach to securely identify devices in smart grids using Trusted Computing mechanisms was presented based on already establish industry standards, namely Trusted Computing and Trusted Network Connect. The proposed IMC/IMV-pair verifies the identity of the smart grid device based on a key stored within the devices TPM. This key cannot be retrieved, thus making it impossible to fake an identity. While this idea renders a good approach for device identification, there needs to be more work done to implement it into an arbitrary smart grid device.

While the presented approach provides a solution for a secure identification of SEGs within a smart grid environment, there are some open question where more research needs to be done. The first problem which needs to be addressed is to show how the solution can scale within the Smart Grid with expected millions of nodes. Further, one open issue is concerned with the exact properties of the SEG to be covered by a possible attestation in addition to the mere identification.Finally, all SEGs which need to be identified securely must provide capabilities of a Trusted Platform. Most importantly, they need

[1] http://trust.inform.fh-hannover.de/

[2] http://trousers.sourceforge.net/

[3] `RSA_private_encrypt()`

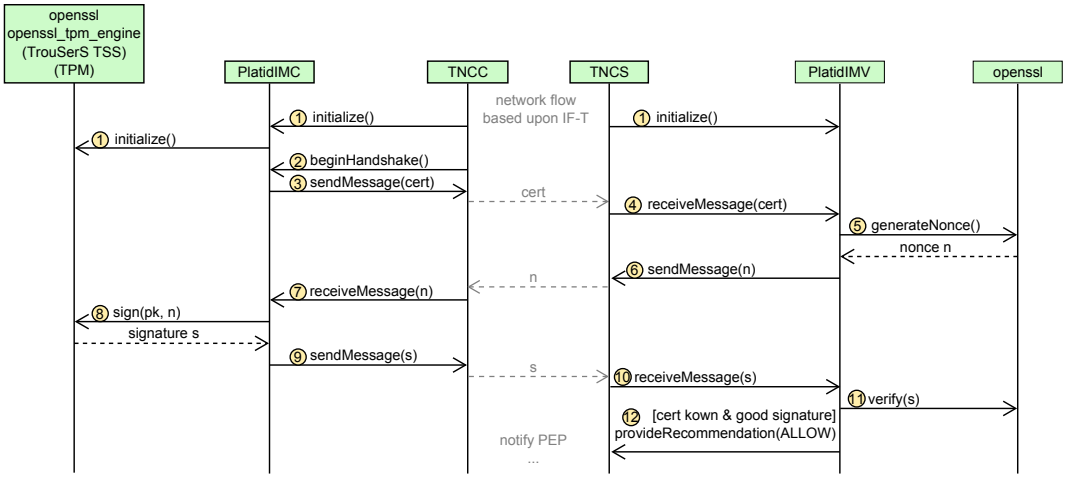

Fig. 3. Sequence diagram.

to be equipped with a TPM or some other hardware security anchor and must be able to perform a TNC handshake.

If those SEGs provide the capabilities of a Trusted Platform, there are more possibilities for increasing the security. First one, it would be possible to perform a remote attestation of the device, ensuring the integrity. Trusted Network Connect would be a promising approach within this scope: the platform identification IMC/IMV-pair could be one part of this integrity measurement. Another possible idea would be the enhancement of a smart grid device to perform a trusted or secure boot without the need for a remote verification.

When using remote attestation, privacy issues arise. Status information can provide too much information about the device and the environment and thus violate privacy requirements of the end user. These privacy issues can be approached by using an already proposed idea: the so-called Privacy Enhanced Trusted Network Connect [1]. This idea allows the usage of policies which allow or deny the access to a certain device property.

Data derived by sufficiently protected SEGs as presented in this paper can support security and event management within Smart Grid infrastructures. First steps in this direction were already made [13], [8] and are to be developed. Further research will be directed to establish self-healing properties based on the trust assertions given by TPM. Automated detection and isolation of malicious SEGs either by neighbouring SEGs or central infrastructure components seems to be a worthwhile approach to the perils that stem from the introduction of ICT components.

REFERENCES

[1] Ingo Bente, Josef von Helden, and Joerg Vieweg. Privacy enhanced trusted network connect. In *INTRUST 2009: Proceedings of the 1st International Conference on Trusted Computing*, pages 129–145, Berlin, Heidelberg, 2010. Springer-Verlag.

[2] P. Congdon. IEEE 802.1 x Overview Port Based Network Access Control. *Internet article available at: http://grouper. ieee. org/-groups/802/1/mirror/8021/docs2000/P8021XOverview. PDF*, 2000.

[3] M. De Paepe, P. D'Herdt, and D. Mertens. Micro-CHP systems for residential applications. *Energy conversion and management*, 47(18-19):3435–3446, 2006.

[4] TCG Infrastructure Work Group. Reference Architecture for Interoperability (Part I). http://www.trustedcomputinggroup.org/resources/infrastructure_work_group_reference_architecture_for_interoperability_specification_part_1_version_10, June 2005. Specification Version 1.0 Revision 1.

[5] TCG Trusted Network Connect Work Group. TNC Architecture for Interoperability. http://www.trustedcomputinggroup.org/resources/tnc_architecture_for_interoperability_version_13, April 2008. Specification Version 1.3 Revision 6.

[6] R. Housley, W. Ford, W. Polk, and D. Solo. Internet X. 509 public key infrastructure certificate and CRL profile, 1999.

[7] S. Karnouskos, O. Terzidis, and P. Karnouskos. An advanced metering infrastructure for future energy networks. *New Technologies, Mobility and Security*, pages 597–606.

[8] N. Kuntze, C. Rudolph, M. Cupelli, J. Liu, and A. Monti. Trust infrastructures for future energy networks. In *Power and Energy Society General Meeting - Power Systems Engineering in Challenging Times*, 2010.

[9] M. LeMay and C. Gunter. Cumulative attestation kernels for embedded systems. *Computer Security–ESORICS 2009*, pages 655–670, 2010.

[10] C. Mitchell et al. Trusted Computing. *Trusted computing*, page 1, 2005.

[11] BC Neuman and T. Ts'o. Kerberos: An authentication service for computer networks. *IEEE Communications magazine*, 32(9):33–38, 1994.

[12] M. Pipattanasomporn, H. Feroze, and S. Rahman. Multi-agent systems in a distributed smart grid: design and implementation. In *Proc. Proc. IEEE PES 2009 Power Systems Conference and Exposition (PSCE'09)*, 2009.

[13] Jennifer Richter, Nicolai Kuntze, and Carsten Rudolph. Securing digital evidence. In *Fifth International Workshop on Systematic Approaches to Digital Forensic Engeneering*, pages 119–130, 2010.

[14] R. Sailer, X. Zhang, T. Jaeger, and L. Van Doorn. Design and implementation of a TCG-based integrity measurement architecture. In *Proceedings of the 13th conference on USENIX Security Symposium-Volume 13*, page 16. USENIX Association, 2004.

[15] Zhexuan Song, Jesus Molina, Sung Lee, Houcheng Lee, Seigo Kotani, and Ryusuke Masuoka. Trustcube: An infrastructure that builds trust in client. In *Future of Trust in Computing*, pages 68–79. Vieweg+Teubner, 2008.

[16] LH Tsoukalas and R. Gao. From smart grids to an energy internet: Assumptions, architectures and requirements. In *Electric Utility Deregulation and Restructuring and Power Technologies, 2008. DRPT 2008. Third International Conference on*, pages 94–98, 2008.