Hierarchical Key Management for Multi-resolution Load Data Representation

Christian D. Peer
Josef Ressel Center for
User-Centric Smart Grid Privacy,
Security and Control
Salzburg University of
Applied Sciences, Austria
Email: christian.peer@en-trust.at

Dominik Engel
Josef Ressel Center for
User-Centric Smart Grid Privacy,
Security and Control
Salzburg University of
Applied Sciences, Austria
Email: dominik.engel@en-trust.at

Stephen B. Wicker
School of Electrical and
Computer Engineering
Cornell University
Ithaca, New York
Email: wicker@ece.cornell.edu

***Abstract*—It has been shown that information about a consumer's actions, beliefs and preferences can be extracted from high resolution load data. This information can be used in ways that violate consumer privacy. In order to increase consumer control over this information, it has been suggested that load data be represented in multiple resolutions, with each resolution secured with a different key. To make this approach work in the real-world, a suitable key management needs to be employed. In this paper, we consider a combination of multi-resolution load data representation with hierarchical key management. Emphasis is placed on a privacy-aware design that gives the end-user the freedom to decide which entity is allowed to access user related data and at what granularity.**

I. Introduction

Increasing energy needs accompanied by an emphasis on alternative energy production creates a need for efficient power grid management and regulated power consumption. This so-called Smart Grid enables load balancing and forecasting within the power grid. In addition it is able to influence the consumer's energy consumption by offering real-time pricing information. Based on this information, consumers can decide when to use devices so as to manage energy costs. Studies show that Smart Grid Infrastructure can reduce peak load during summertime by as much as 20% [1]. To fulfill this task, the Smart Grid relies on Advanced Metering Infrastructure (AMI), a sensor network collecting fine-grained power consumption data. Smart Meters form the core component of an AMI. These devices collect fine-grained consumption data, so-called load data, from a single household. While this data plays an essential part in load balancing and real-time pricing, its collection also creates serious privacy concerns.

It has been shown that apart from information needed for grid operation, other pieces of information can be obtained from fine-grained load data that are sensitive and private to the end user [2]–[4]. Occupancy or sleeping patterns can be determined and certain appliances within the household can be identified and a usage pattern can be drawn. This information can be valuable for targeted marketing as well as criminal purposes. With regard to the former, techniques for matching appliance signatures to load data are called non-intrusive load monitoring (NILM) or non-intrusive appliance load monitoring (NALM) [3].

Acting on privacy concerns, customers and governments are rejecting the deployment of Smart Meters and therefore blocking the deployment of the Smart Grid [5]. To address this issue, privacy preserving methods have to be implemented. Two types of approaches show great potential for ensuring privacy within the smart grid: (i) Secure aggregation of encrypted load data and (ii) consumer control over load data in multiple resolutions, each resolution associated with different access levels. In terms of secure aggregation, Erkin et al. give an overview of the recent development in [5].

This paper will focus on the representation and securement of load data in multiple resolutions. NILM/NALM techniques need high resolution load data to gain accurate results. By lowering the resolution of the load data, NILM/NALM techniques can only achieve limited results. While a low resolution on a daily average is sufficient for accounting purposes, applications like load forecasting or energy saving tools require high resolution load data to achieve useful results. This is where multi-resolution load data representation is needed. Each resolution is encrypted with a different key. Trusted services or third parties are only granted access to the resolution level necessary to fulfill their role. Access can be controlled by a trusted authority, or better, by the user. This adds a new degree of freedom, as the user can decide which party gains access to which data.

An approach on how to represent load data in multiple resolutions can be found in [6]. While this approach describes how to split load data in multiple resolutions, it leaves the question about suitable key generation and management unanswered. In this paper, a key management system suitable for accessing multi-resolution load data within the Smart Grid Infrastructure will be introduced. Furthermore this paper will suggest the use of hierarchical keys to keep key management efforts as low as possible.

This paper also proposes a general communication infrastructure fulfilling the requirements within the Smart Grid Infrastructure. It allows secure communication between entities and third party entities without exposing the Smart Meters

978-1-4799-4934-2/14/$31.00 ©2014 IEEE

to a public network. Hence, it minimizes the risk of possible attacks on the Smart Grid Infrastructure.

The rest of the paper is organized as follows: Related Work and the state of the art are discussed in Section II. The proposed key management system is discussed in detail in Section III. Section IV introduces the idea of hierarchical key management and generation. Finally Section V summarizes the most important findings.

II. Related Work

The following section gives a short overview of the technologies on which this paper is based.

A. Privacy Preserving Architecture

There are different possibilities to enforce privacy protection. One is by regulation and law. While this basic idea is essential for a modern society, it still offers the potential to violate privacy using legal or illegal means. As long as system design and architecture offer the possibility to collect personally identifying information, there is a possibility to violate privacy protection. Therefore a better approach is to ensure privacy protection by design. In [7], Wicker et al. propose a framework for privacy aware design tailored to the development of demand response architectures. They suggest five major elements:

1) *Provide Full Disclosure of Data Collection:* Information on which data is collected, collection purpose and duration of storage has to be provided to the consumer
2) *Require Consent to Data Collection:* User must agree to data collection
3) *Minimize Collection of Personal Data:* Only collect data necessary for functionality of technology, use data as close as possible to the point of collection
4) *Minimize Identification of Data with Individuals:* Anonymize data wherever possible, separate functional records and personally identifying records.
5) *Minimize and Secure Data Retention:* Store data only if necessary and in a way that is not useful in any other context. Notify user if data is lost or stolen.

The system proposed in this paper will take these design principals into account.

B. Multi-resolution load data representation

To preserve users' privacy, the resolution of load data generated by a Smart Meter can be reduced. As different use cases within the Smart Grid require different resolutions, it is difficult to determine a resolution suitable for all use cases. In addition, according to the framework for privacy aware design proposed by Wicker et al. in [7], there is no need for entities to get access to load data in a higher resolution than actually needed. To solve this problem, Engel [6] proposes to provide a Smart Meter's load data in multiple resolutions. Access to a certain resolution is only granted according to an entity's need. Furthermore, the user can decide, if access to a certain resolution is granted or revoked. Engel [6] suggests to use the wavelet transform based on the Haar wavelet and lifting scheme. The Haar wavelet calculates averages and deltas recursively, therefore adding only low computational costs. In addition, transformation is lossless and preserves the aggregate, meaning the whole consumption can be calculated using any resolution.

C. Key Management

To ensure message integrity and prevent eavesdropping, a secure way for communication between the single nodes is required within a Smart Grid. A system guaranteeing both, integrity and confidentiality for the communication channel and authentication and authorization for accessing provided services has to be implemented. A key management system can be seen as the base of such a system.

In the literature, there are different approaches on how to design a key management scheme for a secure communication within a Smart Grid.

Long et al. [8] propose an encryption scheme based on shared secrets. They divide the Smart Grid control architecture into two levels, each with its own key management system, tailored to the computational resources of the devices. While, at a first glance, shared keys seem to be an easy solution, within a growing infrastrucutre, the number of keys is growing rapidly. Every entity has to maintain one key per secure connection to another entity. Hence, causing high efforts for key management, renewal and distribution.

To solve this key management issue and to keep the number of secret keys to a minimum, the use of public keys is recommended. As Smith points out in [9], due to the use of digital signatures enabled by public key cryptography, the secret known by each device cuts down to exactly one, its own private key. Public key cryptography needs a Public Key Infrastructure (PKI) used for establishing, maintaining and distributing the public/private key pair and its assignment to a certain identity. According to Smith, a PKI doesn't have good scalability properties. Therefore, deploying a PKI within a Smart Grid Infrastructure can raise serious issues on how to manage a vast amount of Certification Authorities (CA), maintain the trust path and on how to revoke already issued certificates.

To address these scalability issues, in [10], Baumeister proposes a PKI using multiple CAs, including a CA acting as a bridge between different PKIs. Baumeister also pointed out that several PKIs have been standardized and widely accepted for many years, hence guaranteeing reliability, stability and security.

The same CA topology is also recommended by the United States National Institute of Standards and Technology in [11]. It suggests that every Grid Operator maintains its own PKI based on a hierarchical CA topology. Compatibility, communication and policy enforcement between different PKIs are ensured using bridges.

Through compromising the private key or changing certificate information, a certificate can become invalid before its lifetime is over, in which case it must be revoked. A PKI can publish revoked certificates in a Certificate Revocation

List (CRL). During the verification of a certificate, each entity has to download the CRL to check if the certificate is listed and therefore being revoked. CRLs tend to be large files generating high overhead and hence are hard to process for low resource entities. [12]

A better solution is the implementation of the Online Certificate Status Protocol (OCSP). During certificate validation, the entity sends a query about the revocation status of the certificate to a OCSP server. The provided information is up to date and communication overhead is reduced. The accessibility of the OCSP server can result in a high availability issue. OCSP stapling[1] can be used to solve this problem. An entity obtains a OCSP response for its own certificate and provides the cached response to any entity requesting the certificate. [11], [12]

D. Hierarchical Key Generation

Already in 1981, Lamport [14] suggested to use a hash chain generating a series of One Time Passwords (OTP) to address the problem of identification by sending a secret password over an insecure communication channel. To construct a hash chain of length N, a one-way hash function F is applied to an initial seed value s N-times.

$$F^2(s) = F(F(s)) \tag{1}$$

$$F^N(s) = F(F^{N-1}(s)) \tag{2}$$

F^N is used as the initial value and therefore sent to the server in a secure way. The remaining OTPs $F^1 ... F^{N-1}$ are stored in a secure manner on the client. The client can use F^{N-1} as the next OTP. Knowing F^N, the server can verify the OTP by calculating $F^N = F(F^{N-1})$, but neither the server nor any eavesdropper can determine the next valid OTP as F is a one-way hash function. After a successful authentication, the server stores F^{N-1} as the next value to compare with and F^{N-2} is used for the next authentication attempt. The S/KEY One-Time Password System is one example on how to use OTP for authentication [15].

The idea of hash chains can be found in many security systems [16]. Hash chains or hash trees are also used for access control to JPEG2000 coded images or H.264/scalable coded video (H.264/SVC) [17]–[19].

Imaizumi et al. propose a scheme for hierarchical access control to JPEG2000 coded images in [17]. Image properties are encrypted with different keys. According to the keys gained, a certain resolution or property can be decrypted. To minimize the number of managed keys, a hierarchical key management is introduced. All keys used are derived from one managed master key using hash chains and cyclic shifts. For decryption, the key for the highest resolution, is used. As the used hash function is no secret, the keys needed to decrypt the requested resolution can be derived from the one key provided. It is impossible to decrypt the image in a higher resolution, as the needed keys can't be derived from the one provided. In [18], Wu et al. propose a similar system for access control to JPEG2000 coded images.

In [19], Asghar et al. suggest to use key derivation for encrypting multi-layered coded video (H.264/SCV). The aim is the same as with Imaizumi et al. [17]. A user should be able to watch his/her subscribed layer data with just holding one key. For key generation and distribution, Asghar et al. use the Multimedia Internet Keying Protocol (MIKEY) [20]. Key derivation is done within the MIKEY key generation process. After key generation and distribution, an Advanced Encryption Standard - Counter Mode (AES-CM) Cipher algorithm is used for encryption.

Access control to a multi-resolution representation of load data has similar requirements as for JPEG2000 coded images or H.264/SCV encoded videos. Techniques used for these use cases can be adopted to the Smart Grid. As many successful security systems build on hash chains and one-way hash functions, they can be seen as well-established and secure.

III. Smart Grid Communication Infrastructure

To preserve privacy and to ensure secure communication, a system guaranteeing integrity, confidentiality, and authentication is needed within the smart grid. Encrypted communication between two entities must be confidential, therefore no other entity should be capable of decrypting this communication channel. In addition, third party entities should also be able to use provided services if access is granted to them. It is essential that the system is designed following the framework for privacy aware design proposed in [7]. Each entity should only have access to services and resources on a need to know basis. Information is only stored as long as needed and the user has to be informed how his/her data is being used. Access should be granted on an opt-in basis as opposed to the more prevalent (and less privacy-enabling) opt-out basis.

Possible attacks on the Smart Grid Communication Infrastructure can come from many different sides, namely the user or neighbor, the Grid Operator, Utility or any third party with or without intended access to the Grid. Independent of their origin, attacks can be classified into following groups: altering/forging messages, eavesdropping, data misusage, altering firmware or theft of private keys and denial of service. The approach proposed in this paper addresses these attacks by relying on well-established techniques for content and communication encryption. Hence these techniques can be assumed to be safe.

In Section II, different approaches on designing a suitable key management system for the Smart Grid have been discussed. A PKI is the only suitable key management system with the capability to manage a big infrastructure with a vast amount of issued certificates. The approach proposed in this paper relies on a certificate based Public Key Infrastructure (PKI). Several PKIs are standardized and well-established, therefore guaranteeing reliability and security. This approach also allows third parties to access the Smart Grid Infrastructure in a secure manner.

[1] see RFC 4366 [13]

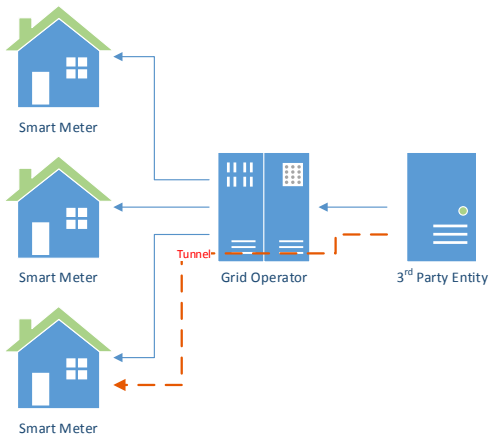

Fig. 1. Smart Meters are connected directly to the Smart Grid Operator. Third Party Entities can access a Smart Meter only via the Smart Grid Operator

The proposed PKI is managed by the Grid Operator and uses bidges to enable communication with other PKIs, therefore simplifying the certificate management as well as the trust path. Each entity acting within the Smart Grid needs to have a valid certificate proving its identity.

The Smart Meter plays a main role in the proposed system and is therefore a trusted device. A Smart Meter must be capable to generate strong keys and store these keys in a manner, that they can't be read or altered from outside. In addition, a Smart Meter must be able to compute cryptographic functions. Like suggested by the United States National Institute of Standards and Technology (NIST) [11] and Wicker et al. [7], a Hardware Security Module (HSM) or a Trusted Platform Module (TPM) can be used to fulfill these requirements. Another requirement is tamper resistance. It must be guaranteed that nobody can intrude or tamper with the Smart Meter without authorization. This embraces changes in hardware as well as in software/firmware. For identification and content encryption, each Smart Meter holds a valid certificate including a private/public key pair.

The assumed Smart Grid Infrastructure is shown in Figure 1. Smart Meters are connected directly to the Grid Operator. Third parties can access the Smart Meter via an Application Programming Interface (API) provided by the Grid Operator. This approach has two benefits: On the one hand, the Grid Operator can choose the technology on how to communicate with the Smart Meters. On the other hand, not exposing Smart Meters directly to a public network improves security as the Grid Operator can act as firewall only allowing authorized entities to communicate with the Smart Meters. Smart Meters are devices with low computational power, vulnerable to Denial of Service Attacks (DoS Attacks). An attack can result in serious issues on grid balancing and pricing. Monitoring and blocking unauthorized traffic by the Grid Operator is an essential part of increasing reliability and availability within the Smart Grid Infrastructure.

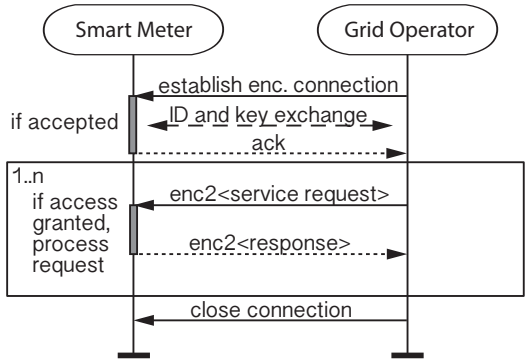

Fig. 2. The Grid Operator can communicate with the Smart Meter using an encrypted connection.

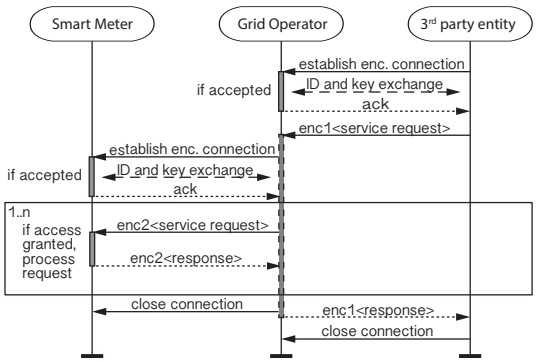

Fig. 3. To establish a connection with a Smart Meter, a third party has to send the request to the Grid Operator. If access is granted by the Grid Operator, it requests the resource from the Smart Meter. If the request is accepted by the Smart Meter, too, the Smart Meter processes the request and sends the response back to the Grid Operator. The Grid Operator forwards the response to the third party entity. Note: encrypted communication is established between third party entity and Grid Operator (enc1) and Grid Operator and Smart Meter (enc2). Hence, the Grid Operator can read the response. Content encryption has do be applied in addition, if necessary.

Figure 2 shows the communication sequence for establishing a connection between the Smart Grid Operator and a Smart Meter. First, the Grid Operator establishes an encrypted connection to the Smart Meter using Transport Layer Security (TLS)[2]. The Smart Meter accepts the connection if the Grid Operator provides a valid certificate. As soon as the encrypted connection is established successfully, the Grid Operator can use the Smart Meter's API to place a service request. If the Grid Operator has permission to access the service, the Smart Meter processes the request and sends the result back to the Grid Operator. The Grid Operator can place multiple service requests. The Grid Operator closes the connection as soon as the connection is not needed any more.

Whereas the Grid Operator can connect directly to a Smart Meter, third party entities must connect via the Grid Operator's API with the Grid Operator acting as a proxy. As shown in Figure 3, first the third party entity establishes an

[2]see IETF RFC 5246 [21]

encrypted connection to the Grid Operator and identifies itself. If the third party entity hast permission to access the Smart Grid Infrastructure, the Grid Operator accepts the connection. Now, using the encrypted channel, the third party sends a service request including the target Smart Meter ID to the Grid Operator. After verifying the service request, the Grid Operator establishes an encrypted connection to the Smart Meter and forwards the service request. Based on the third party entity's certificate, the Smart Meter grants or denies access to the requested service. If access is granted, the Smart Meter processes the request and sends the response back to the Grid Operator. The Grid Operator then forwards the response to the third party entity. The third party can place multiple service requests. As soon as the connection is not needed any more, the Grid Operator closes the encrypted connection to the Smart Meter and the third party entity closes the encrypted connection to the Grid Operator. Note that an encrypted communication is established between the third party entity and the Grid Operator as well as between the Grid Operator and the Smart Meter. Since these two connections are independent, the Grid Operator can read the whole communication between third party entity and Smart Meter. The proposed sequence only guarantees communication encryption preventing eavesdropping. For content encryption and hence privacy protection, the Smart Meter can encrypt the response using the third party entity's public key. An example for content encryption is given in Section IV. It is necessary for grid stability and reliability to differ between communication and content encryption. Within the Smart Grid, there are multiple data flows used for load balancing and controlling/managing the grid. Intruding and altering these data flows can cause severe damage to the grid. Hence, it is necessary that the Grid Operator can monitor and control the data flows within the Smart Grid Infrastructure, requiring the Grid Operator to read the sent messages. For data flows containing private information, content encryption has to be applied, preventing the Grid Operator from reading these data flows. However, it must be ensured, that these data flows can not harm the grid.

The Smart Grid is aiming to alter consumption behavior by providing fine-grained pricing information to the consumer encouraging the consumer to use energy when it is cheapest. Therefore, Wicker et al. [7] propose to broadcast real-time pricing information to the Smart Meters. Each Smart Meter is therefore accumulating price-weighted consumption data. The Electricity Provider can than access the aggregate on a daily, weekly or monthly basis. This proposal ensures protection of consumers' privacy and also fits perfectly in the scheme, proposed in this paper.

Access to and encryption of load data is discussed in Section IV.

IV. Load data encryption and distribution

As discussed in Section II, Engel et al. propose a multi-resolution representation of load data to increase privacy [6], [22]. Access to a certain resolution is based on the conditional access paradigm. A given entity is granted access to a resolution necessary to fulfill its role. As a NILM or NALM algorithm needs high resolution data to achieve accurate results, reducing the resolution of the provided load data reduces the potential for abuse. In addition, the consumer can decide, which entity is granted access to a certain resolution. This adds another degree of freedom as entities have to explain their data usage to gain users' trust.

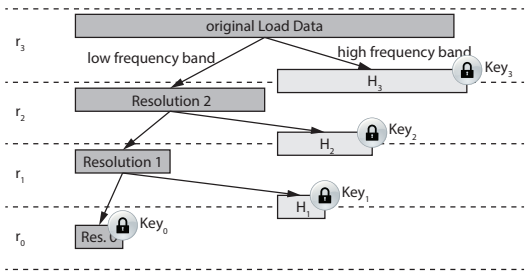

Fig. 4. The Wavelet transform splits load data into high and low frequency bands. The low frequency band equals load data with reduced resolution.

Load data can be represented in multiple resolutions using a suitable wavelet transform, as suggested by Engel et al. in [6]. The Haar wavelet transform suits the requirements best. It consists of calculating averages and deltas, therefore needing few computational resources. The Haar wavelet is a lossless transform; under each resolution, the total consumption over the whole timespan can be derived.

The wavelet transform splits load data in a high and low frequency band recursively up to a certain level. Where the low frequency band is used for the next recursive operation, the high frequency band is preserved. The low frequency band represents the data at a certain resolution with half the number of samples of the next higher resolution. The high frequency band represents the delta of a sample to the according sample of the low frequency band. The values from the high frequency band and the remaining value from the low frequency band are called wavelet coefficients. The wavelet coefficients are needed to do the inverse wavelet transform and restore the load data to a certain resolution. The described steps can be seen in Figure 4.

To restore a certain resolution, the inverse wavelet transform is performed on the low frequency band and its according high frequency band. The inverse starts with the coefficients of the lowest resolution and works its way up to the desired resolution.

To fulfill the conditional access paradigm introduced prior in this section, wavelet coefficients have to be encrypted with a different key for each resolution (from now on resolution key). Granting access to a certain resolution means to distribute the resolution keys for the certain resolution and for all lower resolutions to the requesting entity. A high number of resolution keys has to be managed and distributed, therefore introducing significant overhead for key management and storage.

To address the problem of high key management costs, Hierarchical Keys are introduced. Hierarchical Keys allow the decryption of multiple ciphertexts with a single key although the messages were encrypted with different keys. For example encrypting three messages m_1, m_2, m_3 each with a different hierarchical key k_1, k_2, k_3. In terms of decryption, using k_1 just decrypts m_1, but k_2 or k_3 can be used to decrypt m_1, m_2 or m_1, m_2, m_3, respectively. Hierarchical Keys therefore simplify key management, as less keys have to be known to decrypt multiple messages. Key generation and sample use cases have been already discussed in Section II.

As the use case of multi-resolution representation of load data is quite similar to H.264/SVC and JPEG2000 encryption, techniques proposed in [17]–[19] can be adopted. A hierarchical resolution key is generated for each level of resolution. Resolution keys are derived from a master key using hash chains. Any appropriate one-way hash function can be used. Resolution key renewal can be done within a certain time period, e.g., daily. Wavelet coefficients are encrypted using the appropriate resolution key. The wavelet transform itself is performed on a cyclic basis, e.g., hourly, covering a fixed time span, e.g., the last 24 hours. The wavelet coefficients are packed into a single stream (see Figure 5) and transfered to any entity requesting it. According to the entity's resolution key, the entity is only capable to decrypt the wavelet coefficients of the resolution, to which access was granted to. As the one-way hash function is no secret, the entity can derive the resolution keys to encrypt the wavelet coefficients of a lower encryption but it can't encrypt any wavelet coefficients of a higher resolution.

Figure 6 shows the service requests needed for obtaining load data. This sequence is based on the communication sequence shown in Figure 3. Before sending a service request to the Smart Meter, the entity has to establish a connection via the Grid Operator, as described in section III. To obtain load data, the entity has to go through two steps, (i) obtaining a suitable resolution key and (ii) retrieving the load data. To obtain the resolution key, the entity has to request access for a certain resolution. Therefore, it sends a service request including the certificate and the requested resolution to the Smart Meter. The Smart Meter has to decide, if access is granted. If this is the entity's first access request, the Smart Meter forwards the request to the consumer as he/she can decide, if access for a certain resolution is granted to a certain entity. If the entity is known by the Smart Meter, access can be granted/denied based on previous consumer decision. In case access is granted, the Smart Meter encrypts the resolution key using the entities public key and sends it to the entity. In a second step, the entity sends a load data request to the Smart Meter. The Smart Meter returns a stream containing the encrypted wavelet coefficients, as shown in Figure 5. There is no additional authentication process needed, as the stream is worthless without the resolution key obtained in step one. By decrypting the wavelet coefficients and performing an inverse wavelet transform, the entity can now restore load data up to the resolution, to which access was granted. Load data can

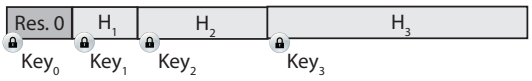

Fig. 5. All wavelet coefficients needed for the inverse transformation are encrypted with different keys and transmitted as a single stream.

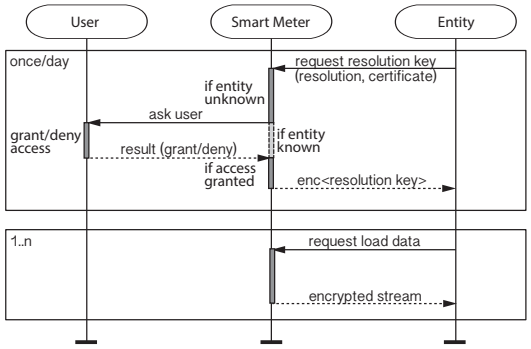

Fig. 6. To access load data, the entity has to request the resolution key for the desired resolution. The user has to decide, if access is granted or denied. After receiving a valid resolution key, the entity can request load data as long as the resolution key is valid. To guarantee content security, the resolution key is encrypted using the entity's public key.

be obtained as long as the issued resolution key is valid. To ensure content security, the resolution key is encrypted using the requesting entity's public key. As only the entity knows it's private key, the resolution key cannot be decrypted by the Grid Operator working as a proxy.

V. Conclusion

Secure communication plays an important role within the Smart Grid. It is essential to ensure authentication, authorization and integrity to prevent unauthorized parties from eavesdropping or altering communication. As consumer related data is collected and transferred, privacy protection is another important issue to address.

In this paper, a secure way of communication, suitable to be used within a Smart Grid Infrastructure, has been introduced. The approach uses a PKI to ensure a secure communication between Smart Meters, the Grid Operator and third party entities. For communication between third party entities and Smart Meters, the Grid Operator acts as a proxy. Hence, the Grid Operator protects the Smart Grid Infrastructure from possible attacks.

To preserve privacy, load data is represented in multiple resolutions. The consumer can decide which entity can access data and at which specific resolution. For multi-resolution representation, the wavelet transform is used, as it adds just a small computational overhead and the transformation process is lossless. Each resolution is encrypted using a different resolution key. Key management efforts are reduced by introducing a hierarchical key management using one-way hash functions for key derivation.

The proposed scheme offers a secure way of communication within the Smart Grid. Methods are used to preserve con-

sumer's privacy. A new degree of consumer freedom is added, as the consumer can decide to whom and at what level his or her personal data can be provided.

ACKNOWLEDGMENT

Financial support by the Austrian Marshall Plan Foundation as well as by the Austrian Federal Ministry of Economy, Family and Youth and the Austrian National Foundation for Research, Technology and Development is gratefully acknowledged. Also, funding by the Austrian Federal Ministry for Transport, Innovation and Technology and the Austrian Research Promotion Agency (FFG) under Project 838793, "INTEGRA", is gratefully acknowledged. Thanks are due to Thomas Stütz for his clarifying comments. Thanks are also due to the School of Electrical and Computer Engineering at Cornell University for hosting and supporting the first author.

REFERENCES

[1] "A national assessment of demand response potential," in *Staff Report, Federal Energy Regulatory Commission*, June 2009. [Online]. Available: http://www.ferc.gov/legal/staff-reports/06-09-demand-response.pdf

[2] M. Lisovich and S. Wicker, "Privacy concerns in upcoming residential and commercial demand-response systems," in *Clemson University Power Systems Conference 2008*. Clemson University, March 2008. [Online]. Available: http://www.truststc.org/pubs/332.html

[3] G. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, Dec 1992.

[4] G. Eibl and D. Engel, "Influence of data granularity on nonintrusive appliance load monitoring," in *Proceedings of the Second ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec '14)*. Salzburg, Austria: ACM, 2014, pp. 147–151. [Online]. Available: http://doi.acm.org/10.1145/2600918.2600920

[5] Z. Erkin, J. Troncoso-Pastoriza, R. Lagendijk, and F. Perez-Gonzalez, "Privacy-preserving data aggregation in smart metering systems: an overview," *Signal Processing Magazine, IEEE*, vol. 30, no. 2, pp. 75–86, 2013.

[6] D. Engel, "Wavelet-based load profile representation for smart meter privacy," in *Proceedings IEEE PES Innovative Smart Grid Technologies (ISGT'13)*, Washington, D.C., USA, Feb. 2013, pp. 1–6.

[7] S. Wicker and R. Thomas, "A privacy-aware architecture for demand response systems," in *Proceedings 44th Hawaii International Conference on System Sciences (HICSS'11)*, Jan 2011, pp. 1–9.

[8] X. Long, D. Tipper, and Y. Qian, "An advanced key management scheme for secure smart grid communications," in *Proceedings IEEE International Conference on Smart Grid Communications (SmartGridComm'13)*, Oct 2013, pp. 504–509.

[9] S. Smith, "Cryptographic scalability challenges in the smart grid (extended abstract)," in *Proceedings IEEE PES Innovative Smart Grid Technologies (ISGT'12)*, Jan 2012, pp. 1–3.

[10] T. Baumeister, "Adapting PKI for the smart grid," in *Proceedings IEEE International Conference on Smart Grid Communications (SmartGridComm'11)*, Oct 2011, pp. 249–254.

[11] NIST, "Smart grid cybersecurity strategy, architecture, and high-level requirements," in *NISTIR 7628 Guidelines for Smart Grid Cybersecurity*. National Institute of Standards and Technology, U.S. Department of Commerce, 2013, vol. 1.

[12] J. Buchmann, E. Karatsiolis, and A. Wiesmaier, *Introduction to Public Key Infrastructures*. Springer Berlin Heidelberg, 2013. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40657-7_5

[13] S. Blake-Wilson, M. Nystrom, D. Hopwood, J. Mikkelsen, and T. Wright, "Transport Layer Security (TLS) Extensions," in *IETF Request for Comments*, no. 4366, April 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4366.txt

[14] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, no. 11, pp. 770–772, Nov. 1981. [Online]. Available: http://doi.acm.org/10.1145/358790.358797

[15] N. Haller, "The S/KEY One-Time Password System," in *IETF Request for Comments*, no. 1760, 1995. [Online]. Available: http://www.ietf.org/rfc/rfc1760.txt

[16] V. Goyal, "How to re-initialize a hash chain," Cryptology ePrint Archive, Report 2004/097, 2004.

[17] S. Imaizumi, M. Fujiyoshi, H. Kiya, N. Aoki, and H. Kobayashi, "A key derivation scheme for hierarchical access control to JPEG2000 coded images," in *Advances in Image and Video Technology*, ser. Lecture Notes in Computer Science, Y.-S. Ho, Ed. Springer Berlin Heidelberg, 2012, vol. 7088, pp. 180–191. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-25346-1_17

[18] Y. Wu, D. Ma, and R.-H. Deng, "Progressive protection of JPEG2000 codestreams," in *International Conference on Image Processing (ICIP '04)*, vol. 5, Oct 2004, pp. 3447–3450 Vol. 5.

[19] M. Asghar and M. Ghanbari, "Cryptographic keys management for H.264/scalable coded video security," in *8th International ISC Conference on Information Security and Cryptology (ISCISC'11)*, Sept 2011, pp. 83–86.

[20] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and K. Norrman, "MIKEY: Multimedia Internet KEYing," in *IETF Request for Comments*, no. 3830, 2004. [Online]. Available: http://tools.ietf.org/html/rfc3830

[21] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," in *IETF Request for Comments*, August 2008.

[22] D. Engel and G. Eibl, "Multi-resolution load curve representation with privacy-preserving aggregation," in *Proceedings of IEEE Innovative Smart Grid Technologies (ISGT) 2013*. Copenhagen, Denmark: IEEE, Oct. 2013, pp. 1–5.