Reconciling Security Protection and Monitoring Requirements in Advanced Metering Infrastructures

Robin Berthier*, Jorjeta G. Jetcheva†, Daisuke Mashima†, Jun Ho Huh*, David Grochocki*, Rakesh B. Bobba*, Alvaro A. Cárdenas‡, and William H. Sanders*
*University of Illinois at Urbana-Champaign, †Fujitsu Laboratories of America, and ‡University of Texas at Dallas
{rgb,jhhuh,dgrocho2,rbobba,whs}@illinois.edu
{jjetcheva,dmashima}@us.fujitsu.com, alvaro.cardenas@utdallas.edu

***Abstract*—Making Advanced Metering Infrastructures (AMIs) resilient to availability and privacy compromises is an important aspect of grid reliability and consumer trust. Lessons learned from traditional cyber systems indicate that achieving resiliency requires the joint deployment of strong protective measures on the one hand, and monitoring and response mechanisms on the other. One major challenge in this endeavor is that of finding the sweet spot between the confidentiality requirements for protecting sensitive AMI traffic through encryption, and the monitoring necessary for full inspection of this traffic. The way to reconcile those conflicting requirements needs to be considered carefully. We review current approaches for protecting and monitoring AMIs and discuss a number of solutions for their productive co-existence.**

I. Introduction

Advanced Metering Infrastructures (AMIs) are designed to allow efficient and reliable communication between utilities and end users of electricity, and are a key enabling technology in the transition to a smarter grid. In typical designs, each home connects to an AMI mesh network through its resident smart meter, allowing it to receive information from the local utility company.

Cyber security is of critical importance for AMI networks [18], [19], [13]. On one hand, AMIs transfer sensitive information, such as details on electricity usage of customers, which, if disclosed, can negatively impact the privacy of users [2]. AMIs thus require *confidentiality* measures. On the other hand, AMIs are used by utilities to send electricity pricing signals, for direct load control, and to send remote disconnect messages, with which attackers might try to tamper in order to disrupt utility operations. For instance, remote disconnect attacks might be attempted [13], where the attacker impersonates the utility and sends a command to one or more smart meters such that their associated customers are no longer subscribed to electricity services, causing power outages at targeted customer premises. Similarly, malicious users might try to tamper with their own usage information to avoid paying for their consumption. Therefore, it is important to *authenticate* and protect the *integrity* of messages exchanged over an AMI.

The large size of typical AMI networks and the embedded nature of smart meters make the distribution and management of cryptographic keys in AMIs challenging. Potential information security threats against AMIs have been discussed in the literature (e.g., [11], [22], [16]), and defenses have been proposed based on existing cryptographic mechanisms. The industry has been moving forward on the implementation of protective measures, and authentication and encryption of all communications have become a de facto requirement.

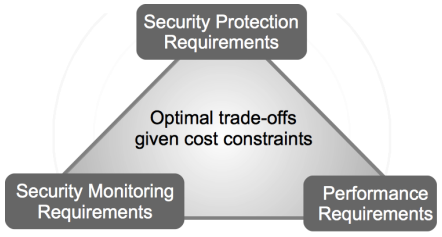

Figure 1. Requirement trade-offs

While encryption of network communications at different layers provides strong confidentiality guarantees, it is also at odds with network visibility, which is of vital importance for network troubleshooting, resource management, and intrusion detection. In particular, encrypted messages preclude the use of most intrusion detection approaches, and might prevent operators from having sufficient visibility over network traffic to identify malicious activity. The goal of this paper is to study those trade-offs in the specific context of an AMI and to identify a set of solutions. Security, performance, and monitoring requirements may not be achievable simultaneously (Figure 1), and need to be reconciled based on use cases, technology capabilities, and cost constraints.

In this paper, we consider AMIs that are equipped with different types of intrusion detection systems (IDSes), e.g., a centralized or a distributed IDS [13], and discuss how encryption requirements and key management schemes may have to be adapted in each scenario. We then introduce the design of a comprehensive monitoring solution that can assist utilities and vendors with AMI deployments and protection strategies. To our knowledge, we are the first to analyze IDS-friendly encryption and key management requirements for AMIs.

II. AMI Networks and Availability Requirements

A utility's AMI network may include up to several million smart meters, and is typically subdivided into smaller Neighborhood Area Networks (NANs), each of which has up to a few thousand smart meters. Each NAN is connected to one or more head-end gateways, which provide connectivity between the NAN and the utility company's network, typically over a WAN. Smart meters are capable of bidirectional communication, using either wireless or PLC technology. In the case of wireless smart meters, each meter may be directly connected to a cellular network provider endpoint (e.g., a Verizon 3G or LTE base station), or meters may form a multihop wireless mesh network using proprietary, or standards-based, mesh protocol stacks. Smart meters themselves not only are endpoints of communication, but also serve to forward traffic on behalf of other attached networks and devices, including devices belonging to Home Area Networks (HANs), Building Area Networks (BANs), Distribution Automation (DA) devices, and gas and water meters.

Communication within an AMI network includes both unicast and multicast applications intended to query individual devices or groups of devices. While today most communication is between a smart meter and the utility network, peer-to-peer (P2P) traffic patterns are emerging as well, e.g., for purposes of DA device coordination. AMI applications have response time requirements ranging from real-time to hours. For example, firmware upgrades have response time requirements on the order of hours. Meter readings, outages, and recovery notifications are examples of applications with response time needs on the order of minutes. Demand response for ancillary services, balancing of supply and demand in the presence of renewable generation resources, and DA device coordination are examples of applications that require real-time communications.

III. Confidentiality and Integrity Requirements

In addition to ensuring timely delivery of communication messages (availability requirement), the confidentiality and the integrity of messages are critical in AMIs. Use of appropriate cryptographic techniques is an effective way to reduce opportunities for exploit and raise the cost and complexity of attacks.

A. Current Approaches

Through the use of public-key cryptography (or asymmetric-key cryptography), such as RSA and Elliptic Curve Cryptography (ECC), confidentiality can be attained by encrypting messages with a recipient's public key, while integrity and non-repudiation can be secured through digital signatures. Alternatively, symmetric-key cryptography, e.g, based on AES, allows the encryption of messages using shared secret keys and enables message and sender authentication through a message authentication code (MAC), such as HMAC. The notable advantage of symmetric-key schemes is that they are less computationally intensive than public-key ones and thus incur shorter processing delays. A hybrid approach in which public-key cryptography is used to establish symmetric session keys between peers is also widely used.

B. Key Management within the AMI Network Stack

In AMI settings, because devices are often resource-constrained, symmetric-key-based approaches are often preferred. In such cases, a typical key management strategy is to use a network-wide shared secret key for broadcast or for exchange of routing information, and/or to have each pair of communicating devices establish pairwise secret keys for unicast communication. Secure multicast can be implemented using a group key, which is shared among a certain subset of devices in the network or may utilize a network-wide shared key for simplicity. Keys may be preconfigured upon installation, distributed by a central node, or established through key agreement protocols [34] [33].

When available, PKI (public-key infrastructure) may be desirable because of its well-established framework for key management and the smaller number of keys involved. Each device is assigned a private and public key pair certified by a trusted authority (a Certificate Authorite (CA)). In a network with n devices, PKI will require the use of at most n keys, whereas a symmetric-key encryption scheme could require $O(n^2)$ pairwise shared keys [26]. Thus, PKI not only lowers the storage requirement on each device, but also can potentially facilitate secure key storage, since only one private key per device needs protection.

Table I shows example protocols within a standard AMI network stack, as well as the typical encryption schemes used at each communication layer. Typically, layer-2 encryption includes pairwise secret keys used to encrypt unicast data between two neighboring nodes, and a network-wide shared key to enable broadcast communication. Layer-3 encryption schemes are intended to protect the routing protocol in operation, and a network-wide shared key is often needed to process encrypted routing headers. For example, [15] suggests the use of 802.1x to obtain a network-wide shared key at bootstrap; the key is then used to encrypt all layer-3 packet headers. In many cases, meters in an AMI network communicate with a Meter Data Management (MDM) server at the utility company's network, and require end-to-end protection of application-layer data. Public-key-based techniques like TLS are sometimes used. The keys may be based on PKI, or may be obtained at bootstrap time, e.g., through 802.1x mechanisms. Notably, encryption schemes are typically used at multiple, and often all, layers at the same time.

IV. Solutions to Satisfy Monitoring and Protection Requirements

A. The Need for Intrusion Detection Systems

While protective measures (such as the ones described in the previous section) are a necessary step to prevent attacks, they are not sufficient to ensure the resiliency of a system. Deploying the most robust protections may initially guarantee

Table I
STANDARDS-BASED AMI STACK

Layer	*Example Protocols*	*Example Encryption*
2	802.15.4, 802.11	Pairwise secret keys between neighboring nodes for unicast (hop-by-hop keys), network-wide secret key for broadcast
3	IPv4, IPv6	Network-wide secret key
3/4	RPL [31], RPL P2P [17], LOAD [8], DFF [7]	Network-wide secret key
4	UDP, TCP	Pairwise secret keys between application endpoints (end-to-end keys)
7	C12.22 [27], COAP [25], DLMS/COSEM [9]	Pairwise secret keys between application endpoints (end-to-end keys), public/private key pair

that known exploits will not succeed, but over time, conditions may change, and new attack vectors could be discovered. That is why protective measures need to be periodically revised, and, more importantly, why networks and systems need to be continuously monitored to detect attack attempts.

A variety of IDS technologies have been introduced over the past two decades. In the context of an AMI, it is recommended that a specification-based detection technology be used in order to implement white-listed networks on which only known activity is allowed [6]. The reason is that an AMI, unlike traditional enterprise networks, carries a limited set of application traffic, and one can precisely specify the expected behavior of nodes. Specification-based analysis requires monitoring of traffic at multiple layers, including the application layer. That means that IDSes should have access to payload information, a feature also known as *deep-packet inspection*, in order to compare traffic against known specifications.

Obviously, full end-to-end payload encryption prevents deep-packet inspection unless decryption keys are shared with monitoring sensors. We now study solutions to this challenge based on different IDS architectures.

B. Sharing Keys with a Centralized IDS

In the case of end-to-end traffic encryption between devices in the field and a utility server (e.g., the MDM server), decryption occurs within the MDM and can be performed either within the MDM software or by dedicated security appliances. Use of dedicated security appliances makes it possible to send traffic that is flowing between the MDM application and the security appliances in the clear, so it can be fully monitored by a centralized IDS with deep-packet inspection techniques. If decryption occurs within the MDM software, then decryption keys have to be shared with the IDS. The central location of both the IDS and the key management server (e.g., certificate authority) eases that operation and also offers a limited risk of keys being exposed or stolen.

The drawbacks of a centralized IDS architecture with either key sharing or access to traffic in the clear are scalability and visibility to traffic at the edge of the network. The scalability challenge is due to the high volume of decryption and/or detection operations in a central location. We note that security appliances in the utility network are often load-balanced when they have to process traffic for millions of meters. The same solution could be used for the centralized IDS. The second challenge is that the central location does not provide visibility over traffic among meters in the field (P2P traffic). With a centralized architecture, only adversarial actions that reach the utility network can be detected. That is why it is important to consider deploying distributed IDS sensors.

C. Sharing Keys with a Distributed IDS

When IDS sensors are distributed across an AMI network, the way to enable them to analyze encrypted traffic is to give them access to decryption keys. Sharing keys with the IDS is not without risk and adds a new level of complexity. Its feasibility depends both on the key management system used and the IDS architecture. In particular, we consider in this section a *dedicated* and an *embedded* IDS architecture. The former consists of dedicated sensors distributed within the area of the AMI network, while the latter consists of sensors embedded within AMI network devices, for example, through addition of network intrusion detection features to meter firmware. We note that an embedded sensor would not need access to decryption keys for traffic destined for the device on which it is installed, because it already has access to clear system operations between the communication module and the core system. However, devices may be used as routers (e.g., in a mesh network), so embedded sensors would need access to decryption keys in order to analyze traffic from nodes whose traffic they have to forward. In the case of an embedded IDS in an AMI mesh network, allowing an intermediate meter/router to decrypt payloads being forwarded may cause a privacy risk. That can be addressed by applying the key-sharing strategy to low-layer encryption only, while using an additional layer of encryption at the application layer to ensure that only the MDM can decrypt the sensitive part of the payloads.

The risk involved with sharing of keys grows with the expected lifetime of the keys. For short-term keys, the compromise of an IDS sensor would have little impact on the security of the infrastructure. However, long-term keys and, in particular, private keys embedded in devices by manufacturers should never be shared.

The key-sharing operation would be required each time a key is initialized or updated. Those two events are infrequent in the context of symmetric group keys, when a single key is already used by a large number of devices. However, the operation becomes expensive if pairwise symmetric keys are used, because a given IDS sensor would need to receive every set of symmetric pairwise keys deployed in the AMI network. That challenge can be partially addressed by strategic deployment of decryption keys to IDS sensors based on the traffic that is expected to pass through them. That would limit not only the number of keys on each sensor, but also the impact

of the compromise of a meter or sensor. We propose methods for selective key distribution in Section IV-G.

D. Advanced Multi-party Cryptosystems

Another approach to preserving data confidentiality in AMI networks while allowing IDS sensors to inspect traffic is to use multi-party cryptosystems that allow a message to be encrypted for multiple receivers (e.g., [32]). In such schemes, while multiple receivers are able to decrypt messages, they do not share the same key, and thus compromise of a single or even multiple IDS nodes does not require rekeying of other nodes. However, such compromises will allow adversaries to violate confidentiality of data. To address that problem, broadcast and/or attribute-based cryptosystems with revocation capabilities should be chosen so that compromised IDS sensors may be revoked. While such cryptosystems provide the necessary functionality to meet protection and monitoring requirements, they typically use pairing-based [1] computations that are computationally expensive. Likewise, solutions that rely on secret sharing [24], such as [12], also suffer from added complexity and performance overheads, making them less suitable for AMIs.

E. Using Traffic Analysis

If the risks associated with sharing of keys with distributed IDSes are considered unacceptable, then IDSes could perform network traffic analysis on the encrypted traffic: available information includes packet sizes, timing, and (in most cases) header information. Traffic analysis without deep-packet inspection is a well-studied field in traditional wired networks [20]. There are two main types of traffic analysis: passive and active. *Passive* analysis simply monitors network flows and collects statistics about them, while *active* analysis places the IDS as one of the devices through which the traffic is routed, and is able to affect characteristics of the flow (usually the timing between packets).

Applications of traffic analysis have ranged from passive detection of stepping stones (or relays) [10], [28] to active detection [23], [29], inference of sentences from encrypted VoIP conversations [30], traffic classification of different flows [21], [4], [5], [3], and exploratory analysis of network traffic [14]. AMI networks, however, present a new and fundamentally different environment for traffic analysis, and most previous results are not applicable to AMI systems. For example, most of the work done in classifying traffic flows assumes TCP communications, but TCP traffic is minimal to nonexistent in most large AMI networks. Similarly, AMI communications can be polling-based, periodic, or event-driven. Sending of alarms to the utility is a typical event-driven communication event: a meter will send an alarm report only if an event is detected. Therefore, an IDS can identify an event based on the observation that a sensor node has sent a packet outside of its normal transmission schedules.

Traffic analysis is a new area of research in AMIs, and further research is needed to understand the limits of this approach with current (and future) AMI applications.

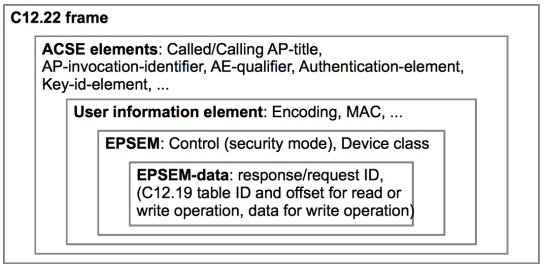

Figure 2. Internal structure of a C12.22 frame

F. Leveraging Partial Encryption

Another option consists of selective encryption of communications based on content. For example, a message containing personal customer information can be signed and encrypted before being sent back to the utility, but other non-identifying pieces of information can be sent in the clear, having only been signed by the source. That allows both the dedicated and embedded infrastructures to monitor the majority of traffic flowing in the AMI network, while protecting customer information from being sent out in the clear. With that approach, the IDS sensors do not need to have access to decryption keys, eliminating the need to share a large volume of keys and the need to protect the keys from being compromised. However, the downside is that any encrypted part of a message is not examinable until the message has reached the utility data center, where it can be safely decrypted by a centralized sensor.

The most obvious candidates for exclusion from the encryption requirement are parts of a message that represent protocol stacks below the application layer. As we explained in Section III-B, encryption schemes used in layers 2 and 3 encrypt routing headers and unicast data between neighboring nodes. We contend that such information does not have strong confidentiality or privacy requirements, and can be sent in clear text if it is properly signed and authenticated.

The application layer needs further scrutiny to identify information that is subject to confidentiality or privacy requirements. We now look at C12.22 [27] in detail. The same approach can be applied to other protocols, like DLMS/COSEM [9]. Figure 2 shows the main internal elements of a C12.22 frame. The Association Control Service Element (ACSE) contains control information about the association between communicating entities, such as caller and called identifiers and authentication information. Each frame can contain one or more Extended Protocol Specifications for Electric Metering (EPSEM) elements. Each carries a response or a request identifier, and the actual C12.19 table identifier and payload data in the case of a read or write operation. The data portion of the EPSEM element can be sent in clear text, authenticated clear text, or authenticated cipher text. EAX' is used for authentication, while EAX'-AES is used to encrypt a portion of the user information.

There are 13 different EPSEM request and related response services. `Identification` gets information about

C12.19 device functionality. `Read` and `Write` transfer table data to and from devices. `Logon` establishes a session. `Security` sets table access permissions. `Logoff` and `Terminate` end a session. `Disconnect` removes a node from a network segment. `Wait` maintains an established session. `Registration` and `Deregistration` add and remove entries in the routing table. `Resolve` looks up native network addresses. Finally, `Trace` gets the list of relays used to reach a given node.

Among those operations, the `Read` response and `Write` request operations need to be encrypted, since they directly affect the C12.19 tables, e.g., the time-of-use tables, security tables, history and event logs tables, and load control and pricing tables. The `Resolve` response operations can identify a smart meter, and should also be encrypted. The `Identification` and `Security` request and response operations can provide adversaries with information that would be useful in an attack on a meter. Hence, they are subject to the confidentiality requirement as well. None of the other operations, such `Logon/Logoff` or `Terminate`, have strong confidentiality or privacy requirements. Thus, among a total of 26 request and response types, 18 can be sent in authenticated clear text and do not have to be encrypted. As a result, distributed IDSes can perform deep packet inspection on those C12.22 messages without having access to any extra decryption keys. Requests and responses carrying encrypted data can be further analyzed within the utility network by the central IDS sensor.

G. Summary and Example Solution

Based on the above options for reconciling protection and monitoring requirements, we believe that traffic analysis and multi-party cryptosystems add too much complexity and performance overhead to be practical. However, we believe that the right combination of selective encryption and key sharing has the potential to preserve confidentiality requirements while allowing deep-packet inspection in the field. Moreover, it leverages technologies that are already available and deployed. Table II presents an example of such a combination. Lower layers that use secret keys can be monitored in the field through distribution of network keys to IDS sensors. Cryptosystems used at the application layer likely involve private keys that are unique to each device. As a result, key sharing should be avoided, and selective encryption should be favored. By encrypting only sensitive data inside payloads and leaving header information and non-sensitive data as authenticated cleartext, an IDS in the field can monitor most of the traffic and let the central IDS complete the monitoring of the encrypted part of the payload.

One issue remains: sharing of pairwise symmetric keys with sensors in the field (shown as unicast traffic at layer 2 in Table II). We outline two possible modes for selective key distribution to minimize the number of keys to share.

First, we can configure sensors in a *lazy mode*, such that decryption keys are shared only after traffic has been received. For instance, if a set of meters communicate periodically with the MDM server and are monitored by a single IDS sensor, then initially, the sensor would have no key and would wait for traffic to pass through before requesting the decryption keys it needs. This on-demand strategy guarantees that only the necessary minimum set of decryption keys is shared with sensors.

Table II
EXAMPLE OF AN IDS-FRIENDLY PROTECTED AMI STACK

Layer	*Attribute*	*Encryption*	*Monitoring*
2	Unicast	Pairwise secret keys	Sharing keys with field IDS
2	Broadcast	Network-wide secret key	Sharing keys with field IDS
3/4	-	Authenticated cleartext	Fully monitored
7	Non-sensitive	Authenticated cleartext	Fully monitored
7	Sensitive	Authenticated ciphertext	Header monitored by field IDS. Payload analyzed by central IDS.

Second, a *prefetch mode* in which keys are prefetched based on *expected* communication patterns can be used. Such patterns can be derived from routing protocol control information. In particular, key-prefetching hints can be derived from source route information. For example, in an AMI network using the RPL routing protocol [31], upon receiving a packet from an RPL root node that contains a source route (which is the case in the RPL non-storing mode), a node can obtain keys for all nodes in the source route following the index at which its own address is located in the source route. Those nodes would likely be forwarding traffic to the RPL root node through this node. The RPL root node is typically co-located with the AMI gateway towards the Internet and is therefore a likely destination for all AMI nodes. Note that we do not recommend that nodes that implement embedded IDSes prefetch keys for nodes listed in RPL P2P source routes [17], as most AMI nodes will not participate in P2P traffic, and prefetching of keys in this case is likely to constitute unnecessary overhead. Nodes in an AMI network may have tens to hundreds of network neighbors. In a dedicated IDS, the sensor could prefetch keys for all nodes within wireless range. However, in an embedded IDS scenario, resource constraints render that approach undesirable. Routing information can help embedded IDS sensors prefetch keys only for nodes that are likely to send traffic through the node at which the sensor resides. For example, in an AMI network running RPL, nodes would prefetch keys for neighbor nodes that have recently sent them RPL DAO packets, but not for other neighbors.

V. CONCLUSIONS

As large cyber-physical infrastructures evolve, so does our understanding of their requirements and thus our ability to adapt security solutions to match their unique characteristics. AMIs have reached a stage where the need for resiliency has

become critical and requires that the cybersecurity community go beyond strong authentication and encryption mechanisms, and offer solutions that would enable use of advanced intrusion detection technologies for continuous monitoring of network infrastrucuture and devices.

This paper offers the first study of solutions for reconciling security protection and monitoring requirements. After reviewing existing approaches for encrypting AMI traffic and managing keys, we presented a variety of options for resolving the conflicting requirements of protection and monitoring, and showed how combining key sharing and selective encryption can help achieve the objective of an IDS-friendly protected AMI stack.

Important challenges remain, such as determination of how best to send short-term pairwise symmetric keys to IDS sensors securely without taxing the limited bandwidth and computational resources within the AMI network, and without exposing the monitoring infrastructure to single-point-of-failure scenarios. We plan to explore solutions to those problems in future work.

ACKNOWLEDGMENTS

This material is based upon work supported in part by the Department of Energy under Award Number DE-OE0000097 and by Fujitsu Laboratories of America. The opinions expressed are those of the authors alone. The authors would like to thank Ed Beroset for his review and insightful feedback and Jenny Applequist for her editorial assistance.

REFERENCES

[1] P. Barreto, H. Kim, B. Lynn, and M. Scott. Efficient algorithms for pairing-based cryptosystems. In M. Yung, editor, *Advances in Cryptology — CRYPTO 2002*, volume 2442. Springer Berlin Heidelberg, 2002.

[2] D. C. Bergman, D. Jin, J. P. Juen, N. Tanaka, C. A. Gunter, and A. K. Wright. Distributed non-intrusive load monitoring. In *Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES Proc.*, pages 1–8. IEEE, 2011.

[3] L. Bernaille and R. Teixeira. Early recognition of encrypted applications. In *Passive and Active Measurement Conference (PAM), Proc.*, pages 165–175, Louvain-la-Neuve, Belgium, April 2007.

[4] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian. Traffic classification on the fly. *SIGCOMM Comput. Commun. Rev.*, 36(2):23–26, 2006.

[5] L. Bernaille, R. Teixeira, and K. Salamatian. Early application identification. In *Conference on Future Networking Technologies (CONEXT 2006), Proc.*, page 6, 2006.

[6] R. Berthier and W. H. Sanders. Specification-Based Intrusion Detection for Advanced Metering Infrastructures. In *17th Pacific Rim International Symposium on Dependable Computing (PRDC), IEEE Proc.*, pages 184–193. IEEE, 2011.

[7] S. L. Cespedes, U. Herberg, A. A. Cardenas, T. Iwao, and M. L. Dow. Depth-first forwarding in unreliable networks (DFF). http://tools.ietf.org/html/draft-cardenas-dff-14, 2013.

[8] T. Clausen, A. C. de Verdiere, J. Yi, A. Niktash, Y. Igarashi, H. Satoh, U. Herberg, C. Lavenu, T. Lys, C. Perkins, et al. The lightweight on-demand ad hoc distance-vector routing protocol-next generation (LOADng). http://tools.ietf.org/html/draft-clausen-lln-loadng-08, 2013.

[9] DLMS User Association. DLMS/COSEM. *Architecture and Protocols*, 2007.

[10] D. Donoho, A. Flesia, U. Shankar, V. Paxson, J. Coit, and S. Staniford. Multiscale stepping-stone detection: Detecting pairs of jittered interactive streams by exploiting maximum tolerable delay. In *Proceedings of the International Symposium on Recent Advances in Intrusion Detection (RAID)*, pages 17–35, 2002.

[11] S. Ganguly, S. Panda, A. C. Dhandapani, and G. Mallappan. Efficient encryption and key management in advanced metering infrastructure. http://morse.colorado.edu/~tlen5710/11s/11AMIKeys.pdf, 2011.

[12] V. T. Goh, J. Zimmermann, and M. Looi. Experimenting with an intrusion detection system for encrypted networks. *International Journal of Business Intelligence and Data Mining*, 5(2):172–191, 2010.

[13] D. Grochocki, J. Huh, R. Berthier, R. Bobba, W. Sanders, A. Cardenas, and J. Jetcheva. AMI threats, intrusion detection requirements and deployment recommendations. In *Third International Conference on Smart Grid Communications (SmartGridComm), IEEE Proc.*, pages 395–400, 2012.

[14] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. BLINC: Multilevel traffic classification in the dark. *SIGCOMM Comput. Commun. Rev.*, 35(4):229–240, 2005.

[15] R. e. a. Kopmeiners. A standardized and flexible IPv6 architecture for field area networks. www.cisco.com/web/strategy/docs/energy/ip_arch_sg_wp.pdf, 2011.

[16] A. Lee. Cryptographic key management (CKM) design principles for the advanced metering infrastructure (AMI). http://www.epri.com/abstracts/Pages/ProductAbstract.aspx?ProductId=000000000001024431, 2012.

[17] J. Martocci, M. Goyal, M. Philipp, A. Brandt, and E. Baccelli. Reactive discovery of point-to-point routes in low power and lossy networks. http://tools.ietf.org/wg/roll/draft-ietf-roll-p2p-rpl/, 2012.

[18] S. McLaughlin, D. Podkuiko, and P. McDaniel. Energy theft in the advanced metering infrastructure. In *4th international conference on Critical Information Infrastructures Security (CRITIS), Proc.*, pages 176–187, Berlin, Heidelberg, 2009. Springer-Verlag.

[19] S. McLaughlin, D. Podkuiko, S. Miadzvezhanka, A. Delozier, and P. McDaniel. Multi-vendor penetration testing in the advanced metering infrastructure. In *26th Annual Computer Security Applications Conference (ACSAC), Proc.*, pages 107–116, New York, NY, USA, 2010. ACM.

[20] A. D. Montigny-Leboeuf. Flow attributes for use in traffic characterization. Technical Report CRC-TN-2005-003, Communications Research Centre, Canada, December 2005.

[21] A. W. Moore and D. Zuev. Internet traffic classification using Bayesian analysis techniques. In *SIGMETRICS '05: Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 50–60, New York, NY, USA, 2005. ACM Press.

[22] M. Nabeel, S. Kerr, X. Ding, and E. Bertino. Authentication and key management for advanced metering infrastructures utilizing physically unclonable functions. In *Third International Conference on Smart Grid Communications (SmartGridComm), IEEE Proc.*, pages 324–329, 2012.

[23] P. Peng, P. Ning, and D. S. Reeves. On the secrecy of timing-based active watermarking trace-back techniques. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P 06)*, pages 334–349, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[24] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[25] Z. Shelby, K. Hartke, and C. Bormann. Constrained application protocol (CoAP). http://tools.ietf.org/html/draft-ietf-core-coap-16, 2013.

[26] S. W. Smith. Cryptographic scalability challenges in the smart grid. In *Innovative Smart Grid Technologies (ISGT), IEEE PES Proc.*, pages 1–3. IEEE, 2012.

[27] A. Snyder and M. Stuber. The ANSI C12 protocol suite - updated and now with network capabilities. In *Power Systems Conference: Advanced Metering, Protection, Control, Communication, and Distributed Resources, 2007. PSC 2007*, pages 117–122, 2007.

[28] K. Suh, D. R. Figueiredo, J. Kurose, and D. Towsley. Characterizing and detecting skype-relayed traffic. In *INFOCOM 2006, Proceedings of the 25th IEEE International Conference on Computer Communications*, pages 1–12, April 2006.

[29] X. Wang, S. Chen, and S. Jajodia. Tracking anonymous peer-to-peer VoIP calls on the internet. In *12th ACM Conference on Computer and Communications Security (CCS), Proc.*, pages 81–91, New York, NY, USA, 2005. ACM Press.

[30] A. M. White, A. R. Matthews, K. Z. Snow, and F. Monrose. Phonotactic reconstruction of encrypted VoIP conversations: Hookt on fon-iks. In *Symposium on Security and Privacy (SP), IEEE Proc.*, pages 3–18. IEEE, 2011.

[31] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. http://www.ietf.org/rfc/rfc6550.txt, RFC 6550 (Proposed Standard), Internet Engineering Task Force (IETF), Mar. 2012.

[32] S. Yu, C. Wang, K. Ren, and W. Lou. Attribute based data sharing with attribute revocation. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '10, pages 261–270, New York, NY, USA, 2010. ACM.

[33] E. Yüksel. Analysing zigbee key establishment protocols. *arXiv preprint arXiv:1205.6678*, 2012.

[34] E. Yüksel, H. R. Nielson, and F. Nielson. Zigbee-2007 security essentials. In *Proc. 13th Nordic Workshop on Secure IT-systems*, pages 65–82, 2008.