The Pennsylvania State University

The Graduate School

Department of Computer Science and Engineering

PROTOCOLS FOR SECURE COMMUNICATION AND TRAITOR TRACING IN ADVANCED METERING INFRASTRUCTURE

A Thesis in

Computer Science and Engineering

by

Siarhei Miadzvezhanka

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2011

The thesis of Siarhei Miadzvezhanka was reviewed and approved* by the following:

Patrick McDaniel
Associate Professor of Computer Science and Engineering
Thesis Adviser

Trent Jaeger
Associate Professor of Computer Science and Engineering

Raj Acharya
Professor of Computer Science and Engineering
Head of the Department of Computer Science and Engineering

*Signatures are on file in the Graduate School.

ABSTRACT

Electrical smart grid is rapidly growing worldwide. Advances in smart metering infrastructure have led to the creation of networks for bidirectional communication between neighborhood electric smart meters and utility management services. Current security solutions for such networks are proprietary, ad-hoc, and often built with a limited understanding of new security threats and risks faced by modern metering infrastructure. Recent research showed existing vulnerabilities in Advanced Metering Infrastructure (AMI) apparatuses, and motivated the need for a new publicly open suite of protocols that would be tailored for security needs of modern AMI systems. We propose a suite of secure protocols for communication between smart meters and smart grid infrastructure. Along with enforcing message integrity and secrecy through typical means, our protocols provide non-repudiation and traitor tracing guarantees. Proposed protocols help to mitigate several types of attacks on smart grid by identifying the origin of attacks against AMI. We implement and evaluate security, scalability, and performance of the proposed protocols. The protocols can support up to thirty thousand smart meters per data collector with an average delay of 22ms per data deposit transaction. We estimate scalability and limitation of the protocols on different processor architectures including Intel i386, ARMv5, and Atom N290.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The emerging smart electric grid is a complex network of sensors and signaling built on top of existing electric grid to enhance distributed electricity generation, transmission, storage, distribution, and consumption. At the consumer end of the smart grid, traditional electromechanical electric meters are being replaced with *smart meters*. Smart meters record and remotely report high-resolution electricity usage data to support new capabilities such as time-of-use (TOU) pricing [14], outage management [12], and demand response load curtailment [6]. Data is transmitted from smart meters to utility-end services through various networks connected by data *collectors*. Together, this network of smart meters, collectors, and utility services is known as the Advanced Metering Infrastructure (AMI).

AMI serves two main functions: (1) to collect and deliver electricity usage from household smart meters to the utility provider and (2) to provide signaling between the utility provider and smart meters for remote management. Examples of remote management include requesting load curtailment, remotely disconnecting or reconnecting power, and upgrading smart meter firmware. The usage data collected from meters is used primarily for billing, but may also be used for load forecasting and outage management. If anyone of these functions comes under the control of a malicious third party, there are severe consequences for the integrity of power delivery and measurement.

Currently deployed AMI systems often utilize ad-hoc security solutions. Such systems assume that the utility provider detects a compromised device immediately. In reality, however, even the minimal set of security guaranties such as confidentiality and authenticity is not always provided [8, 19, 17]. Our penetration tests and other works have discovered collectors and meters have a set of vulnerabilities and insecure engineering solutions that lead to feasible exploitations [21, 20, 32, 5, 7]. While known practices such as the use of public key cryptography can address some of these challenges by maintaining the integrity of AMI communications, one key problem remains: *Given the complexity of AMI networks, it can be difficult to determine the origin of attacks against data collection and signaling.* This is important for several reasons including isolating faulty or malicious equipment and determining the full impact of a compromise.

In this thesis we focus on designing a suite of protocols for secure data collection and signaling in neighborhood level AMI. We define threat and trust models, and the set of the requirements for AMI where smart meters and collectors are considered to be mutually distrusted. Our contributions in this paper include:

- We propose a new smart grid communication suite of protocols aimed at securing communication between smart meters, gateways, and the utility providers. Along with enforcing message integrity and secrecy through typical means, our protocols provide non-repudiation and traitor tracing guarantees. We show how proposed protocols help to mitigate several types of attacks on smart grid by identifying the origin of attacks against AMI.

- We implement and evaluate security, scalability, and performance of the proposed protocols. Our experiments show that protocols can scale up to thirty thousand smart meters per data collector with an average delay of 22ms per data deposit transaction. Since typical smart meters and collectors run on slower than tested hardware, we estimate scalability of protocols on different platforms. We show how protocols scale on ARMv5, Atom N270, and Intel i386ex architectures.

The remainder of this thesis is organized as follows. Chapter 2 explains the development and security concerns of AMI. Chapter 3 describes our protocols. We show implementation details in Chapter 4. We evaluate proposed protocols on a sampling of AMI hardware in Chapter 5, and conclude in Chapter 6.

Chapter 2

AMI Background and Security Motivation

AMI is rapidly growing worldwide. There were more than 30 million smart meters installed in Italy alone [23]. About 10 million meters were installed in recent years in the US, with plans to achieve 43 percent of homes having smart meters by 2014. To better understand the security of these deployments, we would discuss how the smart grid has evolved on top of the existing electric power grid, and then delve into the limitations of the existing smart grid and AMI security. This includes a review of the basic architectures of currently deployed AMI systems as drawn from our two years of experience in smart meter vulnerability testing [20, 21, 19].

2.1 Background

Energy meters have existed for more than hundred years prior to the development of AMI [27] and have evolved through several generations. First generation electromechanical induction electricity meters lacked any communication with the utility provider. A utility employee had to be present on site to check the readings of meters manually. Although very inefficient, such metering infrastructure was reasonably secure, with attacks mostly aimed at circumventing physical security of energy meters. Second generation Automatic Meter Reading Systems (AMR) [31] provided one-way data communication enabling the utility provider to read meters remotely without the need to dispatch an

on-site employee. Most recently, third generation AMI introduced two way communication that facilitates both remote meter reading, and the capability of meters to execute commands on behalf of the utility through signaling. This is the enabling technology behind time-of-use (TOU) pricing schemes, in which the meter itself is aware of the cost of power [25, 14], outage management, which enables utilities to better map out the area affected by an outage [12], and demand response systems, in which smart meters may request that appliances reduce their load [6].

The architecture of a typical AMI network consists of three key components: smart meters, data collectors, and the utility providers (Figure 2.1). The utility provider interacts with meters and collectors through a server computer. Utilities communicate with the rest of the AMI over the *backhaul network*, typically a wide-area public network such as the Internet, cellular, or public switched telephone network (PSTN). Due to their sheer number, smart meters rarely connect directly to the backhaul network. Instead, the collectors act as a gateway from the backhaul to a local area network of meters. Collectors usually sit either on a distribution pole as shown in Figure 2.1-c or on a wall of an individual residence [28]. The meter LAN communicates with an associated collector using a wireless mesh network (Figure 2.1-a), a power line carrier (PLC) (Figure 2.1-b), or other protocol.

Depending on a density of a neighborhood, a single data collector serves several to a thousand of smart meters. In case of wireless mesh, every meter acts as a node for a mesh network. Data propagates through a local area mesh network from the most outbound meter to a data collector using wireless ad-hoc network protocols.

2.2 Security Concerns for Data Collectors

Recently, concerns have been raised on the security of AMI [28, 1, 16, 18]. However, past research has not developed end-to-end protocols tailored to security critical operations of smart meters, collectors, and utility providers they serve. Here, we observe a fundamental limitation in the existing work on AMI security: *very little research has been done towards security solutions that identify misbehaving parties in AMI systems.*

Because they form the gateway between backhaul and LAN networks, data collectors are a single point of failure of AMI systems. Through our efforts in smart meter vulnerability testing [21] it was shown that collectors exhibit vulnerabilities that can be practically exploited to adversely affect both meters and utilities. For example, an attacker can exploit vulnerabilities in the gateway's network protocol stack to upload a malicious firmware or use a flash chip writer to physically[1] tamper with memory. Having taken control over the collector, an adversary can modify all data and signaling passing through with little chance of being detected. In case forged or modified profiles are detected by the utility provider, a collector may simply claim it received corrupted data from the smart meter, or did not receive any data at all. Similarly, smart meters can accuse collectors of modifying electricity usage profiles.

[1] In contrast with smart meters, collectors rarely have tamper protection cases. Thus, physically opening a gateway for malicious activities will not trigger alarms. Being installed on polls in unmonitored areas and having security of a locked enclosure, they are an easy target for an adversary.

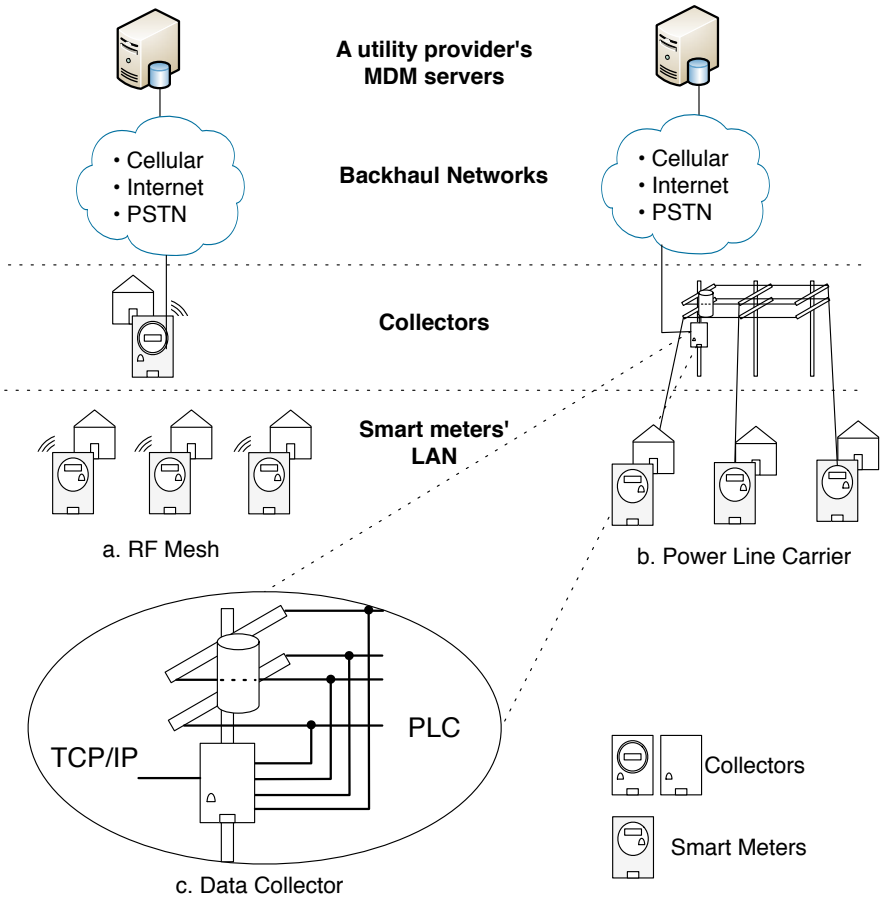

Fig. 2.1. Network types on the neighborhood side of the AMI.

Chapter 3

Protocol Design

In this section we describe trust and threat models, assumptions, requirements, and the design of the communication protocol between smart meters, collectors, and the utility provider.

3.1 Trust, Threat Models, and System Requirements

A typical neighborhood AMI consists of the following entities: a set of smart meters, where individual smart meter SM can be identified by identity ID, a collector identified by C, and typically a unique utility provider U.

AMI systems have two data flows between individual smart meters and the utility provider as shown on Figure 3.1. Usage data is deposited from smart meters to collectors (1), where data may be stored until retrieved by the utility (2). The collector also acts as a forwarding agent for signals from the utility to individual smart meters (3 & 4). Signaling is used for uploading time-of-use (TOU) pricing schemes, outage management, and load balancing. In the event that one of these four links is broken, our protocols can detect which collector or meter was responsible. Note that we assume the utility is universally trusted.

There are two main goals for shown data flows:

- Smart meter's electricity usage profiles must be delivered to the utility with integrity, authenticity, confidentiality, and non-repudiation guarantees.
- Any signaling data delivered from the utility to meters must satisfy confidentiality, authenticity, integrity, and non-repudiation guarantees.

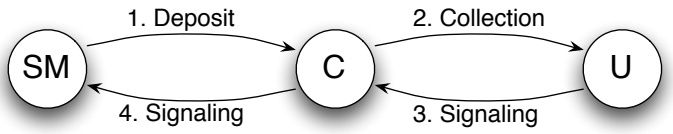

Fig. 3.1. Data flows in AMI.

In this paper, we assume the following security model. A meter is only trusted to produce accurate measurements, but not to always deposit those measurements to the collector, or to execute commands signaled by the utility. Note that work has already been done on detecting meters that misreport usage data [11, 2]. Smart meters and collectors are mutually distrusted. Meters assume the collector can modify, drop, forge, and eavesdrop on transmitted profiles or signaling. Smart meters trust the utility provider, and allow authenticated remote management.

Similarly, the collector is not trusted to deliver deposited usage data to the utility or forward signals to the meter. Since collectors work as proxies for all communication, communication protocols must be secure against arbitrarily manipulations by the collector.

As the utility aims to locate errors caused by meters, collectors, or the network, the utility is always trusted to operate correctly. The utility is entrusted to collect data from the collector, manage smart meters and encryption keys for AMI infrastructure. The utility trusts smart meters only to produce valid electricity usage profiles, but do not trust either networks or collectors.

Networks are not trusted by smart meters, collectors, or the utility provider. We assume that an adversary can listen, modify, insert, and delete packets in both LAN and back-haul networks.

3.2 Data Assumptions

Based on an implementation, the utility may collect active and reactive power usage, voltage deviations, power outages, currents, etc. The correctness of the protocols does not depend on measured data. For the clarity of presentation, in this paper we assume that utility collects only active power usage profiles on a predefined schedule. Power usage profiles include total active power (in kW·h) measurements and corresponding time stamps. An example of a typical profile is shown in Table 3.1.

Time stamp, s	Total active power, kW·h
1300280942	10.82
1300281842	10.91
1300282742	11.03
1300283642	11.12

Table 3.1. An example of active power usage profile.

3.3 Key Distribution

We assume all participating parties have a unique pair of private and public keys, signed by a certificate authority and embedded by the manufacturer. Typically, an AMI deployment is owned and managed by the utility, making it an appropriate certificate authority. We assume the following distribution of keys. A smart meter with identity ID has public and private keys k^+_{ID} and k^-_{ID}. A collector has the key pair k^+_c and k^-_c, and similarly, the utility has the key pair k^+_u and k^-_u. The smart meter holds a copy of both the collector's and utility's public keys in its memory. The collector and utility maintain each other's public keys as well as the public keys for all smart meters within their respective domains. In summary, each entity in the AMI can communicate securely with every other entity, except for inter-meter communication. Assumptions of our model do not introduce a key distribution overhead compared to existing AMI systems that have embedded either shared secret, or public and private keys into every smart meter and collector.

3.4 Protocol Design

The protocols consist of three basic functions: (1) *data deposit*, (2) *data collection*, and (3) *signaling* as shown in Figure 3.2.[1] Both data deposit and data collection are necessary to propagate usage data from the meter to the utility provider, but they may happen asynchronously with the collector acting as a store and forward point. Data deposit defines how smart meters upload energy usage profiles to collectors on a predefined schedule (typically on 15 minutes to 1 hour intervals). Data collection defines how energy usage profiles are downloaded by the utility from the collector on a daily basis. Finally, signaling is used to propagate control commands from the utility to smart meters. Both hops needed for signaling occur immediately after each other. Although not shown, smart meters can also use signaling to inform the utility of power outages. The details of each protocol are shown in Table 3.2, and explained further here.

[1] Extended diagram can be found in appendix A, on Fig. A.1

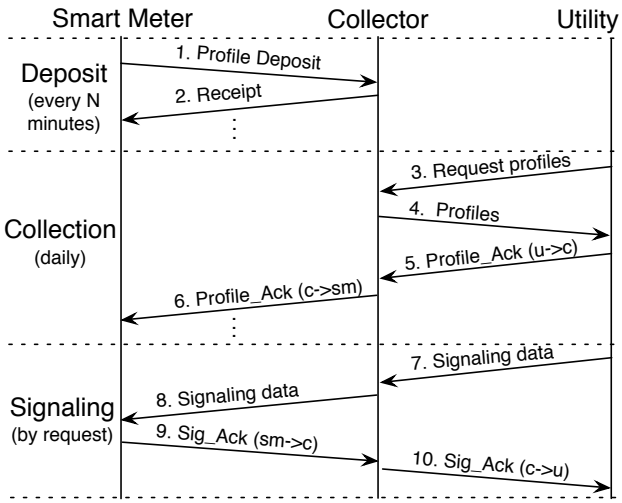

Fig. 3.2. Deposit, Collection, and Signaling protocols.

Protocol Messages	Content of messages
	Data Deposit
1. Profile deposit	$SM \rightarrow C : ID\|C\|E(k_u^+, k_s)\|E(k_s, ts\|kW \cdot h)\|time\|Sign_{ID}(k_{ID}^-, ID\|C\|E(k_u^+, k_s)\|E(k_s, ts\|kW \cdot h)\|time)$.
	in terms of $S_{ID} = ID\|C\|E(k_u^+, k_s)\|E(k_s, ts\|kW \cdot h)\|time$, the message can be simplified:
	$SM \rightarrow C : S_{ID}\|Sign_{id}(k_{ID}^-, S_{ID})$
2. Receipt	a) C verifies time stamp and signature created by SM, knowing smart meter's public key k_{ID}^+.
	b) $C \rightarrow SM : R = Sign_c(k_c^-, S_{ID}\|time)$.
	c) SM verifies received time stamp and receipt R, knowing collector's public key k_c^+.
	Data Collection
3. Request profiles	$U \rightarrow C : data\ request = U\|C\|ID\|time\|Sign_u(k_u^-, U\|C\|ID\|time)$.
4. Profiles	a) C verifies data request by checking digital signature.
	b) $C \rightarrow U : C\|U\|S_{ID}\|Sign_{ID}(k_{ID}^-, S_{ID})\|time\|Sign_c(k_c^-, C\|U\|S_{id}\|Sign_{ID}(k_{ID}^-, S_{ID})\|time)$.
	c) U verifies correctness of data by checking both digital signatures and corresponding time stamps.
5. Profile Ack(u→c)	$U \rightarrow C : U\|C\|ID\|Hash(S_{ID})\|time\|Sign_u(k_u^-, U\|C\|ID\|Hash(S_{ID})\|time)$.
	in terms of $Profile\_Ack = U\|C\|ID\|Hash(S_{ID})\|time$, the message can be simplified:
	$U \rightarrow C : Profile\_Ack\|Sign_u(k_u^-, Profile\_Ack)$.
6. Profile Ack(c→sm)	a) C checks the validity of the Profile_Ack.
	b) If valid, C deletes local copies of $S_{ID}, Sign_{ID}, R_c$. Acknowledgements are piggybacked to the next receipt sent to SM:
	c) $C \rightarrow SM : R = Sign_c(k_c^-, S_{ID}\|time)\|Profile\_Ack1\|Sign_u(k_u^-, Profile\_Ack1)\|Profile\_Ack2...$
	d) SM verifies Ack by checking identities, time, and signatures corresponding to acknowledged data.
	e) SM deletes local copies of triples $S_{ID}, Sign_{ID}, R_c$, and corresponding Ack.
	Signaling
7. Signaling data(u→c)	$U \rightarrow C : U\|C\|ID\|time\|E(k_{ID}^+, signaling\_data)\|Sign_u(k_u^-, U\|C\|ID\|time\|E(k_{ID}^+, signaling\_data))$.
	in terms of $Signal\_data = U\|C\|ID\|time\|E(k_{ID}^+, signaling\_data)$, the message can be simplified:
	$U \rightarrow C : Signal\_data\|Sign_u(k_u^-, Signal\_data)$
8. Signaling data(c→sm)	a) C verifies signaling data by checking digital signature and time stamp.
	b) $C \rightarrow SM : Signal\_data\|Sign_u(k_u^-, Signal\_data)$
9. Signaling Ack(sm→c)	a) SM verifies time stamps, digital signatures.
	b) $SM \rightarrow C : Ack = ID\|C\|U\|Hash(signaling\_data)\|time\|Sign_{ID}(k_{ID}^-, ID\|C\|U\|Hash(signaling\_data)\|time)$
	c) in terms of $Ack = ID\|C\|U\|Hash(signaling\_data)\|time$, the message can be simplified:
	d) $SM \rightarrow C : Ack\|Sign_{ID}(k_{ID}^-, Ack)$
10. Signaling Ack(c→u)	a) C verifies an Ack received from SM, and deletes locally stored Ack copy.
	b) $C \rightarrow U : Ack\|Sign_{ID}(k_{ID}^-, Ack)$

Table 3.2. Protocol messages and their content

3.4.1 Data Deposit

SM starts with creating a sample of the electricity usage profile, from a collected set of $ts|kW \cdot h$ pairs[2] as shown in Table 3.1, where ts is a time stamp, $kW \cdot h$ is a corresponding total active power usage, and symbol | represents concatenation. SM then generates an ephemeral secret key k_s which is unique per data deposit transaction. The profile sample is encrypted with secret key: $E(k_s, ID|ts|kW \cdot h)$. The ephemeral secret key itself is encrypted with utility's public key $E(k_u^+, k_s)$. The identities, encrypted profile sample, encrypted ephemeral key, and time stamp are digitally signed with smart meter's private key k_{ID}^- and sent from SM to C as shown on Figure 3.3. In terms of transmitted data samples S_{ID}, data deposit message is simplified as $SM \ \rightarrow \ C: \ S_{ID} \mid Sign_{ID} \ (k_{ID}^-, \ S_{ID})$.

$SM \rightarrow C : ID|C|E(k_u^+, k_s)|E(k_s, ts|kW \cdot h)|time|Sign_{id}(k_{id}^-, ID|C|E(k_u^+, k_s)|$
$|E(k_s, ts|kW \cdot h)|time)$.
In terms of $S_{id} = ID|C|E(k_u^+, k_s)|E(k_s, ts|kW \cdot h)|time$, the message can be simplified:
$SM \rightarrow C : S_{ID}|Sign_{ID}(k_{ID}^-, S_{ID})$

Fig. 3.3. Data-deposit message from smart meter SM to collector C.

[2]Based on an implementation, the utility may collect active and reactive power usages, voltage deviations, power outages, currents etc. The correctness of the protocols does not depend on measured data. For simplicity, in this paper we show smart meters collecting only active power usage.

Upon receiving a usage profile from a smart meter, C verifies the signature created by SM and checks attached and signed time stamp against an internal clock (Table 3.2-2a). Should the time stamp be old, or digital signature invalid, C reports such behavior to U. If the time stamp is fresh and the signature is valid, C stores received usage profile and calculates a receipt R to be send back to SM. The correctness of receipt R must be verifiable by both SM and U. SM must not be able to replay or create counterfeit receipts. Four steps of the data deposit protocol are shown on Fig 3.4.

Although there are many ways to construct a receipt R, we chose to use digital signature scheme $R = Sign_c(k_c^-, S_{ID}|time)$ to provide described above guarantees. The receipt message is sent back to smart meter, where it is verified using collector's public key (Table 3.2-2b and 3.2-2c). Although original profile samples already include time stamps and provide protection against replay attacks, it is important to note that C cannot verify such time stamps without revealing confidential data. Therefore, additional time stamps that can be verified by C were added into every data deposit message. Added time stamps create two important improvements to the protocol. First, SM can prove to U that C received profile data at a specific time by providing U with corresponding data-receipt pairs. Second, since C checks time stamps on received from SM data, SM cannot longer trick C into issuing the receipt for old data. For the protocol to work correctly, clocks between SM, C, and U have to be loosely synchronized.

C is not able to view or modify any transmitted profile samples for any particular customer, since data is encrypted with unknown to C secret key k_s. Digital signatures over encrypted data and time stamps provide data authenticity and integrity guarantees. Non-repudiation guarantees are provided by receipts generated at C and digital signatures

1. $SM \rightarrow C : S_{ID}|Sign_{ID}(k^{-}_{ID}, S_{ID})$
2. C verifies time stamp and signature created by SM, knowing meter's public key k^{+}_{ID}.
3. $C \rightarrow SM : R = Sign_{c}(k^{-}_{c}, S_{ID}|time)$
4. SM verifies received time stamp and receipt R, knowing collector's public key k^{+}_{c}.

Fig. 3.4. Data deposit protocol.

at SM. Data for which SM has a receipt cannot be repudiated by C. Similarly, SM cannot repudiate data that has been sent to and verified by C. Whenever usage profile is lost or corrupted as a result of collector's misbehavior, SM can prove to U that C has received profile by showing the original usage profile and corresponding receipt R.

3.4.2 Data Collection

Data collection part of the protocol describes the way utility collects profile samples from the collector. Typically, data collection happens on a daily basis. However, shorter intervals can be used depending on the utility needs, the size of smart meter profiles, and the capacity of the collector. On a predetermined schedule, U requests profile data from C with a request profile message (Figure 3.5-1). Data request includes source U and destination C identities, a set of smart meters $\{ID\}$ for which data is requested, a time stamp, and a digital signature. After verifying the request (Figure 3.5-2), C sends requested electricity usage profiles to U (Figure 3.5-3). When received, U has to ensure integrity and authenticity of received data (Figure 3.5-4). Integrity and authenticity properties derive from public key cryptography and time stamps. Upon successful decryption of the message, U will learn the ephemeral key k_s and be able to decrypt the

1. $U \rightarrow C : data\ request = U|C|ID|time|Sign_u(k_u^-, U|C|ID|time)$
2. C verifies data request by checking digital signature.
3. $C|U|S_{ID}|Sign_{ID}(k_{ID}^-, S_{ID})|time|Sign_c(k_c^-, C|U|S_{ID}|Sign_{ID}(k_{ID}^-, S_{ID})|time)$
4. U verifies correctness of data by checking both digital signatures with corresponding time stamps.
5. $U \rightarrow C : U|C|ID|Hash(S_{ID})|time|Sign_u(k_u^-, U|C|ID|Hash(S_{ID})|time)$,
in terms of Profile_Ack = $U|C|ID|Hash(S_{ID})|time$, the message can be simplified:
$U \rightarrow C$: Profile_Ack|$Sign_u(k_u^-$,Profile_Ack).
6. C checks the validity of the Ack. If valid, C deletes local copies of $S_{ID}, Sign_{ID}, and R_c$.

Fig. 3.5. Data collection protocol.

profile sample. After verifying authenticity and integrity of the profile sample, the utility creates an acknowledgement *Ack* that is sent back to smart meter through collector (Figure 3.5-5).

To achieve data availability, the utility acknowledges all usage profiles before they are deleted from smart meters. Acknowledgements received from U are piggybacked to receipts for new data. Whenever C receives usage profiles from SM for whom it has an *Ack*, it sends the *Ack* along with computed receipt (Table 3.2-6c). When *Ack* received from C is successfully verified, SM erases usage profile and corresponding receipt of the profile that has been acknowledged (Table 3.2-6d,e). A complete mapping of messages from Figure 3.2 to their content is shown in Table 3.2.

3.4.3 Signaling

Signaling allows the utility provider to remotely manage smart grid devices. For instance, the utility can remotely connect and disconnect devices, update firmware images,

upload TOU profiles, change authentication and encryption keys, etc. Power outages can also be identified much quicker with help of signaling from smart meters back to the utility. Note that signaling is an online protocol, and therefore all failures in communication will be visible in near real time, compared to data deposit and collection that work on predefined schedule.

The utility provider starts with encrypting signaling data $signaling\_data$ with public key of a destined smart meter. The encrypted signaling data and a time stamp are digitally signed and sent to a collector (Figure 3.6-1). Upon successful verification of the digital signature (Figure 3.6-2), the collector retransmits signaling message to the smart meter (Figure 3.6-3). When message is received by the smart meter (Figure 3.6-4), it is verified, decrypted, and processed. Ack is then created and sent back to C (Figure 3.6-5). After verification, the collector deletes locally stored $Signal\_data$, and retransmits Ack to the utility provider (Figure 3.6-7).

1. $U \rightarrow C : U|C|ID|time|E(k^{+}_{ID}, signaling\_data)|Sign_u(k^{-}_{u}, U|C|ID|time|E(k^{+}_{ID}, signaling\_data))$.
In terms of $Signal\_data = U|C|ID|time|E(k^{+}_{ID}, signaling\_data)$, the message can be simplified:
$U \rightarrow C : Signal\_data|Sign_u(k^{-}_{u}, Signal\_data)$
2. C verifies signaling data by checking digital signature and time stamp.
3. $C \rightarrow SM : Signal\_data|Sign_u(k^{-}_{u}, Signal\_data)$
4. SM verifies time stamps, digital signatures.
5. $SM \rightarrow C : Ack = ID|C|U|Hash(signaling\_data)|time|Sign_{ID}(k^{-}_{id}, ID|C|U|Hash(signaling\_data)|time)$.
In terms of $Ack = ID|C|U|Hash(signaling\_data)|time$, the message can be simplified:
$SM \rightarrow C : Ack|Sign_{ID}(k^{-}_{ID}, Ack)$.
6. C verifies Ack, and deletes locally stored $Signal\_data$, and $Sign_c$
7. $C \rightarrow U : Ack|Sign_{ID}(k^{-}_{ID}, Ack)$

Fig. 3.6. Signaling protocol.

In the signaling protocol, the data integrity and authenticity are achieved by using public key cryptography and time stamps. Since smart meter's private key is known only to a unique smart meter, collector will not be able to view or modify any signaling data without corrupting it. Time stamps prevent replay attacks. Digital signatures provide non-repudiation guarantees for signaling data.

3.5 Security Evaluation

The proposed suite of protocols helps to identify the origin of attacks against AMI infrastructure. Devices originating corrupted, forged, or malicious data or signaling will be detected. Such detection is possible due to cryptographic properties of receipts and digital signatures that are stored by collectors and smart meters. For instance, whenever data is lost as a result of collector's misbehavior, a smart meter can prove to the utility provider that the collector received profile data at a specific time. This can be done by providing the utility with corresponding usage profile-receipt pairs. Similarly, a malicious smart meter will be detected if it tries to accuse a honest collector in modification of the profile data. Since collectors check and sign time stamps corresponding to every profile deposit, a malicious smart meter trying to trick collector into issuing the receipt for forged or old data will also be detected by utility.

The protocols, however, do not allow to remotely identify the device that implements a denial of service attack by simply dropping all traffic. In case a collector drops data for a particular smart meter, solutions such as onion routing can be used to route packets through several smart meters before they reach the final destination. In general though, manual inspection is required to identify devices that drop traffic. Similarly,

the protocol design does not address availability requirements. Not much can be done against a trivial physical attack, such as an adversary cutting the wires in PLC network, or physically breaking a collector. However, redundant systems can help to improve system's availability in the event of accidental damages of AMI devices or networks. For example, wireless mesh networks shown in section 2.1 can find an alternative path and reroute traffic in case the original path becomes unavailable.

Chapter 4

Implementation

The protocol was implemented in C++. We use OpenSSL Crypto [29] library for cryptographic constructions. Specifically, RSA (2048 bit) and Blowfish (128 bit) algorithms are used for public key and secret key cryptography respectively; SHA1 (160 bit) is used as a hash function. Details of implementation environment are shown in Table 4.1.

The implementation has three components corresponding to smart meter, collector, and the utility. Smart meter's component includes a data generator to simulate realistic energy usage profiles. The data generator produces uniformly distributed active power consumption data with matching time stamps over tested period of time.

Operating System	Linux 2.6.32
Programming language	C++
Compiler	GCC 4.4.3
Crypto	OpenSSL Crypto library
Asymmetric cipher	RSA (2048 bit)
Symmetric cipher	Blowfish (128 bit)
Hash function	SHA1 (160 bit)
Digital Signature	RSA

Table 4.1. Implementation environment

Chapter 5

Evaluation

In this chapter we focus on an experimental evaluation of the proposed protocol. The objective of experiments is to evaluate scalability and performance of the protocol. We describe the implementation details, experimental setup, performed tests, and analyze the results.

5.1 Experimental Setup

We use a set of virtual machines shown on Figure. 5.1 to setup experiments. Each virtual machine (VM) has an Intel 3.2GHz CPU[1], 512MB RAM and runs Linux 2.6.32. Separate VMs are used for a set of smart meters (VM1), the collector (VM2), and the utility (VM3). We consider different sizes of neighborhood in the experiments, with number of smart meters ranging from 1 up to 100,000. While the actual AMI systems have each smart meter running on a dedicated hardware, simulation of thousands of smart meters running on a single virtual machine can lead to CPU contention and socket exhaustion. To avoid such conflicts, we separate 1% of smart meters into virtual machine VM4.

In the following experiments we focus on performance on the collector, mostly ignoring bandwidth and network latency limitations. Our analysis shows that even for

[1]In our experiments the host for virtual machines had 4 cores 3.2GHz CPU. It allowed us to run each VM on a separate core.

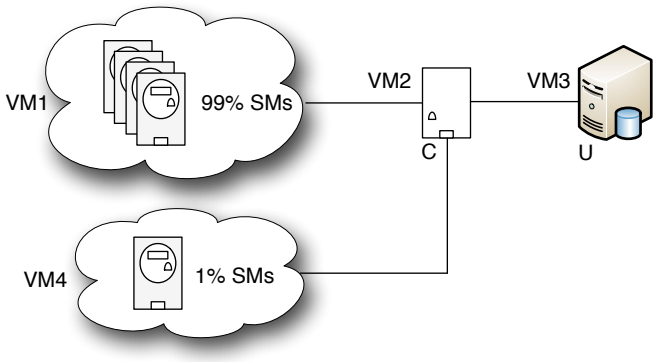

Fig. 5.1. Experimental setup with four virtual machines.

large number of smart meters, the link bandwidth will not be an issue for any modern AMI system. In the data deposit, collection, and signaling parts of the protocol, most of the messages are roughly of size 1kB. Data deposit operation (Figure. 3.1) requires 2 messages (2kB) to be transmitted. Assuming data deposits are done once every 15 minutes, it will require minimum bandwidth of 0.01778 kbps for one smart meter. A large-scale system with 100 000 smart meters will require 1.778 Mbps bandwidth, which is well below limitations of current PLC, BPLC, or wireless mesh networks. In the experiments we assume one-hop distance from smart meters to the collector, and therefore neglect network latency. However, in a wireless mesh networks (Figure 2.1-a), the collector is typically several hops away from the majority of the smart meters. In a large AMI system with thousands of smart meters it is nearly unavoidable to have network diameter of several hops, which may result in considerable network latency. However, such network latency would bring a fairly constant error to results of our experiment, and for small

diameter AMI systems can be neglected. The same analysis can be given for networks with 30 minutes deposit intervals that would result in reducing load on the collector roughly in half, improving the performance.

5.2 Latency Evaluation

We measure data deposit transaction latencies for a set of loads starting with low (100 smart meters) and finishing with extremely heavy load of 100,000 smart meters. For every load level, the meters are set to initiate the profile deposits every 5 minutes. The launch of smart meters is uniformly distributed among data deposit interval. We measure the latency on every smart meter in VM4 from the time the connection with collector is established until receipt is received and verified. Such latency includes four components: data transmission from a smart meter to the collector, signature verification and receipt creation on the collector, and the transmission of the receipt back to the smart meter. The experiments last for 15 minutes of simulated time for every load level.

The average latencies and standard deviations for different loads are shown in Table 5.1. For a low load (100 meters) the latency does not exceed 11.1ms, averaging at 10.104ms. The latencies over time for a low load are shown on Fig. 5.2. Higher loads of 1,000 and 10,000 smart meters produce average latencies of 10.259ms and 13.950ms respectively. Corresponding standard deviations do not exceed 5.86 and reflect the relative stability of transactions' latencies.

Number of meters	Avg. latency (ms)	Std. Dev.
1	9.978	0.35
10	10.008	0.27
100	10.104	0.31
1000	10.259	2.34
5000	11.695	5.29
10000	13.950	5.86
30000	19.960	12.03
100000	50.227	38.01

Table 5.1. Average latency and standard deviation for varied number of smart meters.

5.3 Throughput Evaluation

To recall from Chapter 3, smart meters initiate data deposit operation on predefined intervals, typically from 15 minutes to an hour. Collectors, on the other hand, constantly work with hundreds or potentially thousands of meters. Such imbalance creates a bottleneck of the AMI at the collector. For throughput consideration we analyze the limitation of the proposed protocols on number of smart meters.

Figure. 5.3 shows the experimental latency function of the number of smart meters on a logarithmic scale from Table 5.1. A major part of the latency in the plotted graph is based on the performance of cryptographic operations that are done by CPU. While in our simulated environment the collector had a dedicated VM with 3.2GHz CPU (single core), actual collectors usually run on much slower hardware. Therefore we try to estimate the limitation of the protocols based on linear growth of latency functions on different processor architectures. The experimental latency function has a linear growth, and can be approximated as $y(x) = 0.3 \times 10^{-4}x + 9.978$, where constant 0.3×10^{-4} determines the growth of the function. That constant depends on a collector's hardware and has

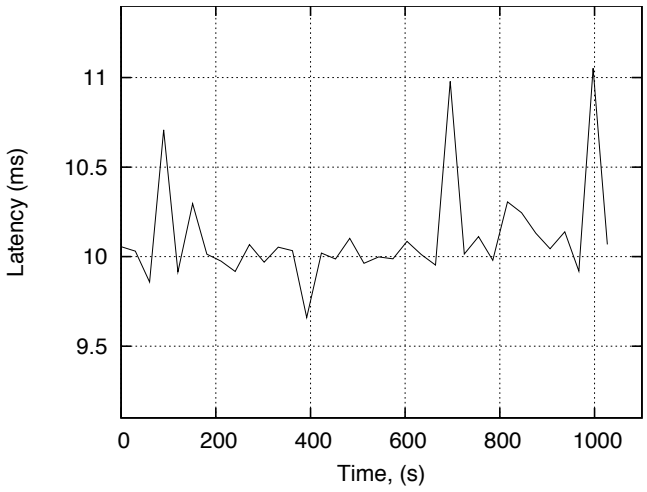

Fig. 5.2. Latencies at smart meters over time for a 100 smart meters

quasi-linear dependency on CPU performance. We use the standardized set of CPU benchmarks from *SPEC* [30] and *cpubenchmark* [24] to estimate the performance of the protocols on different architectures.

We show an estimated performance of the protocols on Atom N270, ARMv5, and Intel 386ex processors on Figure 5.4. As can be observed from the graph, even on Intel 386ex (25MHz) architecture the protocols can scale for up to 1150 smart meters with 80% time utilization on the collector for 15 minutes deposit intervals. Atom N270 and ARMv5 can scale up to 6,800 and 11,000 smart meters with 60% time utilization and up to 7,300 and 11,700 smart meters with 80% time utilization respectively.

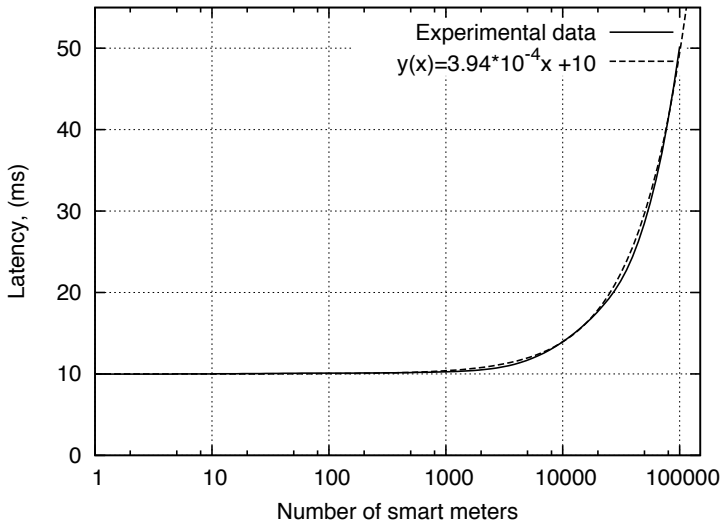

Fig. 5.3. Averaged cost of data deposit transaction for varied number of meters.

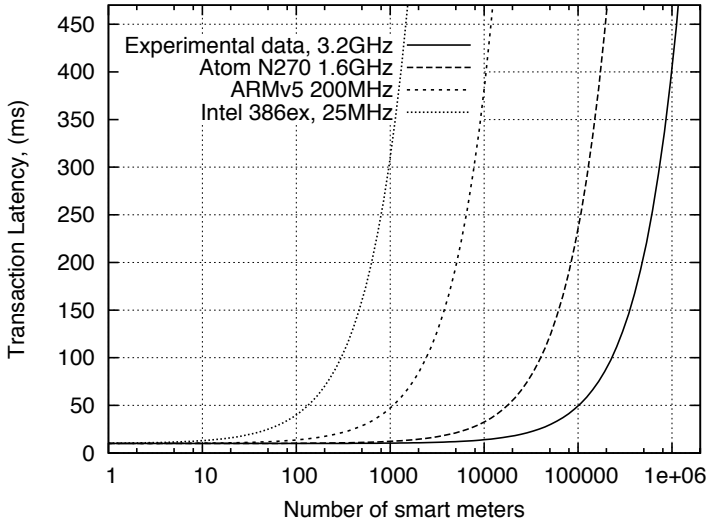

Fig. 5.4. Estimated performance of the protocols on different architectures.

Chapter 6

Conclusion

Advanced Metering Infrastructure must have dependable and reliable bi-directional communication between smart meters and utility providers. We proposed a suite of protocols that provide confidentiality, integrity, non-repudiation, and traitor tracing guarantees. Proposed protocols help to mitigate attacks on smart grid by identifying misbehaving parties. We implemented and evaluated security, scalability and performance of the proposed suite of protocols. The protocols can handle up to thirty thousand smart meters per data collector with an average delay of 22ms per data deposit transaction, which is more than 30 times the number of smart meters of currently deployed systems. The estimated scalability of the protocols on ARMv5 and Atom N270 platforms is 7,300 and 11,700 smart meters respectively.

Appendix

Extended Protocols Diagram

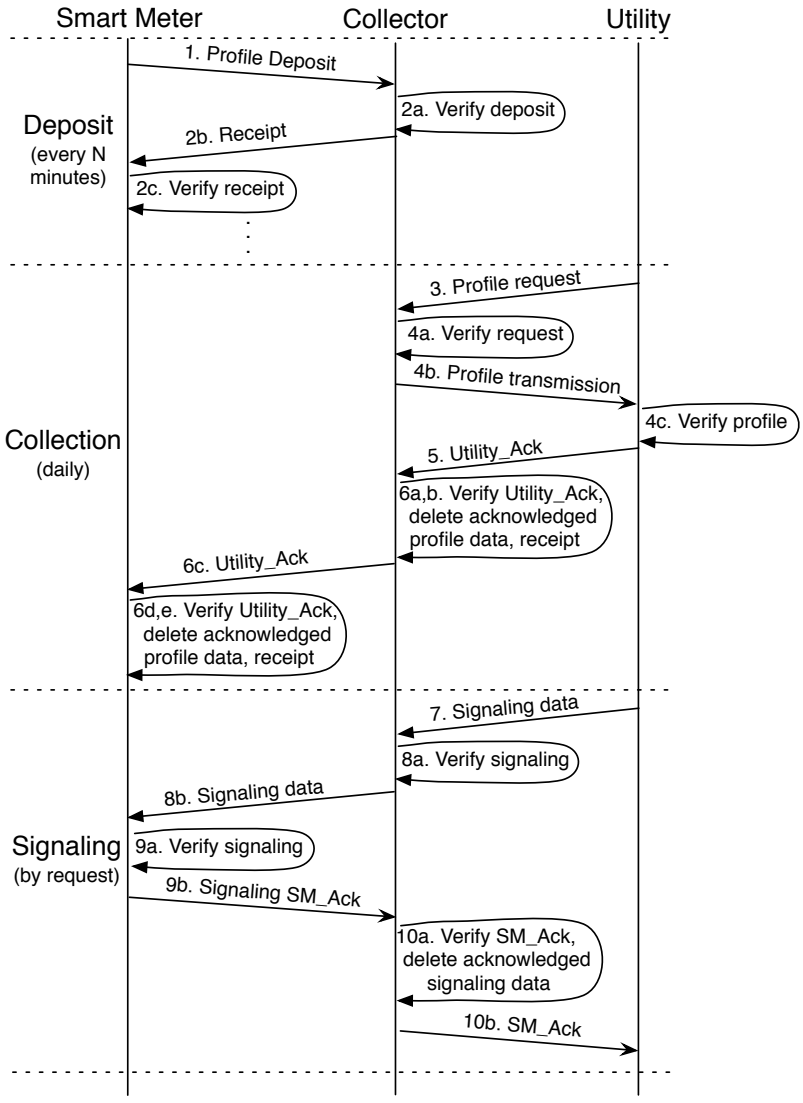

Fig. A.1: Extended data deposit, data collection, and signaling protocols.

References

[1] Ross Anderson. Who controls the off switch. `http://www.cl.cam.ac.uk/~rja14/Papers/meters-offswitch.pdf`, October 2010.

[2] C.J. Bandim, Jr. Alves, J.E.R., Jr. Pinto, A.V., F.C. Souza, M.R.B. Loureiro, C.A. Magalhaes, and F. Galvez-Durand. Identification of Energy Theft and Tampered Meters Using a Central Observer Meter: a Mathematical Approach. In *Transmission and Distribution Conference and Exposition (IEEE PES)*, 2003.

[3] A. Brothman, R. D. Reiser, N. L. Kahn, F. S. Ritenhouse, and R. A. Wells. Automatic Remote Reading of Residential Meters. *IEEE Transactions on Communication Technology*, 13(2):219 – 232, 1965.

[4] P. Datta Ray, R. Harnoor, and M. Hentea. Smart power grid security: A unified risk management approach. In *Security Technology (ICCST), 2010 IEEE International Carnahan Conference on*, pages 276 –285, 2010.

[5] Mike Davis. SmartGrid Device Security. Adventures in a new medium. IOActive, BlackHat, 2009.

[6] Miriam Goldberg. Measure twice, cut once. *IEEE Power and Energy Magazine*, pages 46 – 54, May/June 2010.

[7] Travis Goodspeed. Smart meter crypto flaw worse than thought. `http://rdist.root.org/2010/01/11/smart-meter-crypto-flaw-worse-than-thought`, 2009.

[8] G.W. Hart. Residential energy monitoring and computerized surveillance via utility power flows. *Technology and Society Magazine, IEEE*, 8(2):12 –16, jun. 1989.

[9] J. Huges. The integrated energy and communication systems architecture; volume iv, technical analysis, appendix a., 2004.

[10] Dana Hull. Smart Grid: PGE acknowledges thousands of inaccurate utility bills. `http://cfp.acm.org/wordpress/?searchterm=smart+grid\&p=221`, May 2010.

[11] Doosun Kang and Kevin Lansey. Filtering bad measurement data for water distribution system demand estimation. *Journal of Water Resources Planning and Management*, 136(4):512–517, 2010.

[12] Raymond Kelley and Ron D. Pate. Mesh networks and outage management. White Paper, September 2008.

[13] Chris King. The Economics of Real-Time and Time-of-Use Pricing For Residential Consumers. Technical report, American Energy Institute, 2001.

[14] Chris S. King. The Economics of Real-Time and Time-of-Use Pricing For Residential Consumers. Technical report, American Energy Institute, 2001.

[15] C. Laughman, Kwangduk Lee, R. Cox, S. Shaw, S. Leeb, L. Norford, and P. Armstrong. Power signature analysis. *Power and Energy Magazine, IEEE*, 1(2):56 – 63, mar. 2003.

[16] Michael LeMay, George Gross, Carl A. Gunter, and Sanjam Garg. Unified architecture for large-scale attested metering. In *Proceedings of the 40th Annual Hawaii International Conference on System Sciences*, HICSS '07, pages 115–, Washington, DC, USA, 2007. IEEE Computer Society.

[17] Mikhail A. Lisovich, Deirdre K. Mulligan, and Stephen B. Wicker. Inferring personal information from demand-response systems. *IEEE Security and Privacy*, 8:11–20, 2010.

[18] S. Mak and D. Radford. Design considerations for implementation of large scale automatic meter reading systems. *Power Delivery, IEEE Transactions on*, 10(1):97 –103, jan. 1995.

[19] P. McDaniel and S. McLaughlin. Security and privacy challenges in the smart grid. *Security Privacy, IEEE*, 7(3):75 –77, may-jun. 2009.

[20] Stephen McLaughlin, Dmitry Podkuiko, and Patrick McDaniel. Energy theft in the advanced metering infrastructure. In *Proceedings of the 4th international conference on Critical information infrastructures security*, CRITIS'09, pages 176–187, Berlin, Heidelberg, 2010. Springer-Verlag.

[21] Stephen McLaughlin, Dmitry Podkuiko, Sergei Miadzvezhanka, Adam Delozier, and Patrick McDaniel. Multi-vendor penetration testing in the advanced metering infrastructure. In *Proceedings of the 26th Annual Computer Security Applications Conference*, ACSAC '10, pages 107–116, New York, NY, USA, 2010. ACM.

[22] A.R. Metke and R.L. Ekl. Smart grid security technology. In *Innovative Smart Grid Technologies (ISGT), 2010*, pages 1 –7, jan. 2010.

[23] National Energy Technology Laboratory. NETL Modern Grid Initiative - Powering Our 21st-Century Economy. Technical report, United States Department of Energy Office of Electricity Delivery and Energy Reliability, Dec 2008. p. 17.

[24] Pass Mark Software, CPU Mark. `http://www.cpubenchmark.net/`, February 2011.

[25] Arthur H. Rosenfeld, Douglas A. Bulleit, and Robert A. Peddie. Smart Meters and Spot Pricing: Experiments and Potential. *IEEE Technology and Society Magazine*, March 1986.

[26] Bruce Schneier. Prepaid Electricity Meter Fraud. http://www.schneier.com/blog/archives/2010/09/new_prepaid_ele.html, Sep 2010.

[27] M.F. Schwendtner. Technological developments in electricity metering and associated fields. *Metering and Tariffs for Energy Supply, Eighth International Conference on (Conf. Publ. No. 426)*, pages 240 –242, jul. 1996.

[28] R. Shein. Security measures for advanced metering infrastructure components. *Power and Energy Engineering Conference (APPEEC), 2010 Asia-Pacific*, mar. 2010.

[29] OpenSSL Cryptography and SSL/TLS Toolkit Crypto library. `http://www.openssl.org/`.

[30] Standard Performance Evaluation Corporation SPEC CPU2006, CPU2000. http://www.spec.org/.

[31] Paraskevakos G. Theodoros and Thomas W. Bushman. Apparatus and method for remote sensor monitoring, metering and control. Patent: 4241237, Dec 1980.

[32] Kim Zetter. Security pros question deployment of smart meters. Threat Level: Privacy, Crime and Security Online, Mar 2010.