Security and Privacy of Smart Meters: A Survey

Farid Molazem
74505116
University of British Columbia
faridm@ece.ubc.ca

ABSTRACT

In this survey, we discuss the security and privacy issues of smart meters. We study security mechanisms developed for smart metering systems and classify them to two categories of intrusion-detection-based, and remote-attestation-based techniques. We analyzed each class and discussed their strengths and weaknesses. Also, we study the threats for the privacy of users of smart meters and discuss the techniques developed to protect the privacy of customers. We classify these techniques into two groups of architecture-based and protocol-based techniques and analyze them.

1. INTRODUCTION

Smart grids are poised to replace traditional power grids in north America and Europe. Unlike traditional grids, smart grids use Advanced Metering Infrastructure (AMI, also known as smart meters) with two-way communication capabilities. Smart meters have a key role in providing monitoring and control capabilities for smart grids. During the past few years, smart metering systems have been widely deployed in the world. It is estimated that by the end of 2015, more than 250 million smart meters will be installed around the globe [4]. The reason for that is rooted in the benefits resulted from developing smart metering infrastructures. Smart meters run a software and create a two-way communication between the consumer and the utility server. This enables the meters to provide various services that were not feasible before. For instance, the meters can instantly detect and report outages, provide more precise billings, reduce in-person visits, and provide detailed consumption reports to help reduce energy consumption. Rapid deployment of smart grids has resulted in developing advanced metering infrastructure without adequate security and reliability planning [22, 31, 40]. Current estimates indicate that in the US alone, $6 billion is lost by providers due to fraud [28]. The financial benefits that would be accrued from tampering with smart metering devices makes security of smart meters an important issue. Currently many security issues exist in the domain of smart meters and many vulnerabilities and attacks are discovered against these systems [35, 27, 14, 39].

On the other hand, the current architecture of smart grids have serious privacy issues [2]. The meters record fine-grained measurements and transmit them to a database at a utility server. It has been recognized that detailed consumption data are private information and can lead to leakage of information such as the devices being used at homes [18, 24]. These information can be used to build a profile of customer behavior. Information such as if the users are at home, when they come back from work or when they eat can be extracted from the consumption profiles.

Due to benefits that result from tampering with the smart meters for malicious users, the security and privacy of smart meters is an important issue. The fact that smart meters will be installed where the adversary can have a full physical access to (for instance homes) makes the protection mechanisms more challenging. What we want to study in this survey are the security and privacy protection mechanisms that can be used for the smart metering systems. To do that, we study the attacks that exist and could be applied to the meters. We investigate the techniques that have been studied to protect the meters and classify them. We also analyze the strength and weakness of these techniques and identify the research gaps that currently exist in the literature. mIn summary, we do the following studies in this paper:

- We study the techniques proposed for addressing security of smart meters.
- We classify the security techniques for smart meters and analyze their strengths and weaknesses.
- We study the privacy issues of smart metering systems.
- We classify the privacy protection techniques for smart meters and analyze them.
- We identify the research gaps that currently exist in the literature in the field of security and privacy of smart meters.

The structure of the paper is as follows. In Sec. 2 we provide an overview of the architecture of smart meters. In Sec. 3 we study the threat model for smart metering systems. In Sec. 4 we go over security techniques proposed for smart meters and classify them into two categories of intrusion-detection-based techniques and remote-attestation-based techniques. We discuss current works on these two categories and point out their strengths and weaknesses. In Sec. 5, we study the architectures that address the privacy flaws of current smart metering systems and discuss their pros and cons. In the end, we summarize our studies, provide our conclusions and discuss the most important research gaps that we found in Sec. 6.

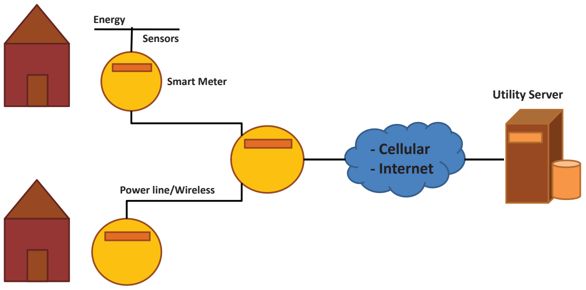

Figure 1: Smart grid

Figure 2: A modern smart meter

2. SMART METER STRUCTURE

In this section, we provide some more details about the structure of smart meters and the components that make it work.

Smart meter is a networked embedded system that has sensors to receive data (electrical current for instance). Here we briefly explain the basic components of a smart meter. Smart meter as an essential component of a smart grid (Fig. 1). Each smart meter has a base and a cover that protects the internal parts of the device. The cover is sealed through a flag-style tamper seal to protect the meter from being tampered with. Inside the meter, there is a Microcontroller Unit (MCU). MCU, transfers data measured by the low-level meter engine to a flash memory. Also, MCU saves logs of important events during the activity of the smart meter.

Smart meter receives data regarding power usage, water consumption, etc. through an Analog Front End (AFE). This component receives analog data (for example electric currents), converts it to digital data and passes it to microcontroller. Smart meters are equipped with memory component that can be in the form of a flash memory. Microcontroller can read/write data from/into this flash memory for backing up data or sending it to the utility server. Also, the smart meters have an LCD which displays data for users of the meter.

For the meters to be able to communicate with each other and the server, they are equipped with an NIC card. The type of communication interface differs from region to region. In some regions (US), companies might prefer Zigbee as an standard communications interface and in some other regions companies might use power line communication interface. It is not feasible to connect every individual meter directly to the utility server. Normally, the meters form a LAN and connect to the server through their gateways. Common topologies for meter LAN's are RF Mesh networks or power line communication networks. In RF mesh networks, repeaters send data to the collector which is communicating with the server. In power line communication, the meters are connected to each other and to the collector through power lines. The utility server is connected to the collectors through Internet, cellular network, etc., and gathers all data from them.

For the meters to have the capability of providing time-of-use billing services, the smart meters also have to be equipped with a real-time clock (RTC). This clock should be reasonably accurate and also be synchronized with the server clock on a regular basis to prevent any major drift. The real-time clock is normally integrated with the analog front end.

Here we mentioned the basic necessary components of smart meters. But smart meters can have many additional components. For instance, smart meters could be equipped with a disconnect switch. Therefore, if the utility wants to disconnects a user's power, it sends a request to the meter, and the meter will open the switch and disconnects the power. A modern smart meter is shown in Fig. 2.

3. THREAT MODEL

In this paper, we consider the adversary to be a malicious user who can obtain system level access to the meter or has access to the meter network. System level access can be obtained through recovering the root password or exploiting any vulnerability in the system. For example [29] has shown the applicability of password recovery attack on a smart meter. Also, similar to any other computing systems, smart meters are potentially vulnerable to exploits such as buffer overflow attacks. In [10] it has been shown that an attacker can gain system level access to a modern car and control it through applying buffer overflow attack. The scale in which smart meters are deployed makes performing regular updates and patching more difficult which increases the vulnerability of advanced metering infrastructure against these attacks. Obtaining access to the meter network is easy as the meters are installed at homes and places that are physically accessible to the attacker. The attack vectors that is considered for the smart metering systems includes all software attacks against the meter including denial of service attacks, man in the middle attacks, buffer overflow attacks and etc.

In [26] six type of attackers against AMI are identified. These attackers are categorized based on their motivations for attacking smart metering systems:

1. Curious eavesdroppers: Who just want to know about the activities of their neighbors.

2. Motivated eavesdroppers: Who want to gather information for malicious purposes.

3. Unethical customers: Who want to steal electricity and not pay for the services.
4. Intrusive data management agencies: Who want to gather private information and create user profiles for marketing and economic purposes.
5. Active attackers: Who want to perform large-scale attacks. Terrorist activities fall into this category.
6. Publicity seekers: Who are more interested in getting famous rather than harming the users and gaining financial rewards.

Based on the architecture of the smart meters and characteristics of attackers, in [5], the following high level groups of attacks against smart meters are identified:

- Network: Communication interception and traffic analysis. Traffic modification, injection, and replay.
- System: Authorization or authentication violation. Spoofing of utility system. Compromise node, spoofing of metering device.
- DoS: Resource exhaustion, Signal Jamming, Dropping packets.

To perform the above attacks, an adversary can exploit potential vulnerabilities of the smart meters. These system-level vulnerabilities can exist on any computing device and are discovered everyday. In [16, 9] there are more details on low-level system vulnerabilities and attacks.

4. SECURITY

We categorize existing software techniques to provide security for smart meters into two main groups of intrusion-detection-based techniques and remote-attestation-based techniques. In this section, we study both these techniques and analyze their limitations in addressing requirements of smart metering systems. Physical protection techniques (such as sealed covers, etc.) are not in the scope of this study.

4.1 Intrusion-Detection-Based Techniques

4.1.1 Definitions and background

Intrusion is defined as a set of actions that try to bypass the security mechanisms of a computer system [30]. Normally, intrusion is occurred by an adversary accessing the system through network, exploiting an OS vulnerability or a third party application running on the system. Based on this, intrusion detection is the process of monitoring the system and network for any access, activity, and modification on data. An intrusion detection system (IDS) is a software or hardware that automatically performs this monitoring process.

Intrusion detection systems are either host-based or network-based. Host-based IDS is installed on the system and monitors communication between applications, applications and OS, network access, file access, and etc. One downside of hos-based IDS is that updating the underlying OS can affect functionality of the IDS. Network-based IDS is attached to network and protects all the machines present on that network segment. All the incoming and outgoing traffic for the devices on the network segment pass through the IDS and is checked for attacks.

Whether an IDS is network-based or host-based, it performs attack detection using signature-based (also known as misuse detection), anomaly-based, or specification-based techniques [30, 23]. Here we briefly explain each of these techniques:

- Signature-verification approach: In this approach, the intrusion is detected based on a predefined sequence of events, for example sizes of specific packets. The advantage of signature verification systems is that normally, for sniffing network traffic, only one machine could be assigned to monitor the whole traffic for several machines and this reduces the computational overhead for the systems. But on the other hand, it is difficult to develop signature-verification based systems that can detect new attacks and these systems should be constantly updated to detect newer attacks.
- Anomaly detection approach: This technique is more suitable for detecting new attacks but it can have a high false alarm rate. Also, some users (like administrator) and activities might be difficult to model.
- Specification based approach: Specification-based technique is also an approach that can be used to detect new attacks. In this technique, the security behavior of a systems is carefully specified and violation from this specification is a sign of intrusion. This approach could have a low false alarm rate. But it is difficult to implement this technique since security specifications should be written for all monitored programs and the programs are frequently updated.

The main difference between specification-based techniques compared to anomaly-based and signature-based techniques is that, in the latter two approaches, the low-level activities of the processes are modeled (such system calls, activities on levels of 2, 3, or 4 of OSI layers). But for specification-based techniques, the behavior of the processes is modeled on a higher level by building a state machine of the process.

4.1.2 IDS Requirements for Smart Meters

In this section we analyze the necessary requirements for intrusion detection system built for smart meters with respect to specifications, architecture, and the types of attacks against metering systems.

The components of a traditional IDS are

1. Sensors or agents to monitor activity
2. management server to manage and centralize gathered data
3. database server to store data
4. console so that the administrators can check the status of the system.

When applying intrusion detection systems with the above architecture to the domain of smart meters, the specifications and architecture of the target system should be considered. Smart meters are deployed in the scale of millions, they are low-powered and have limited resources. Based on the architecture and characteristics of smart metering systems, in [5] the main challenges of IDS in the context of advanced metering infrastructure are recognized to be the following:

- Being highly accurate: The number of nodes in smart meter networks is very large (in the order of millions) and false positives can aggregate quickly and impose a huge overhead on the alert management system, making the system impractical.
- Ability to detect unknown attacks: Since smart metering systems are new, no comprehensive database of attacks exists for them. Therefore, it is important for the IDS developed for AMI to detect unknown attacks accurately.
- Having low overhead: Smart meters are deployed on large scales. Also the meters themselves are small, low-powered computing devices. Therefore, any intrusion detection solution for AMI should have low overhead both on the network and on the meter.
- Architecture: The large scale and distributed architecture of smart meter network makes developing an intrusion detection system, specially with a centralized architecture, challenging.
- Heterogeneity: The network infrastructure is not homogeneous and there are constraints such as bandwidth limitations that should be considered when developing network intrusion detection systems. Studying characteristics of network communication mechanisms is important. For example since most of the communication is wireless, it provides opportunities for traffic interception attacks.
- Resilience of management server against attacks: Management server can be a single point of failure for a thousands or millions of nodes.

There are several solutions suggested in the literature for addressing the above challenges. In [5], Berthier et al argue that specification-based approaches are more suitable for the case of smart meters. They mention three main reasons for this. First, specification-based techniques are more accurate compared to signature-based techniques. Second, lack of database of attacks for the case of smart meters makes building a black lists challenging. Third, since the functionalities of smart meters are limited and for a specific domain, specifying and modeling the system activities is easier and less-costly compared to general purpose computers.

In [5, 6] distributed architecture for IDS developed for smart meters is suggested to address some of the challenges. The number of nodes in the network of smart meters could be in the order of millions and the traditional architecture in which there is a central database and management system is not suitable for the case of smart meters. Therefore they suggest using a decentralized approach in which most of the processing is done locally and there are a set of alert aggregators to which the nodes are connected.

Another important issue for the network IDS that motivates a distributed architecture is that it has to be resilient to attacks against the management server. This problem can be handled by using several redundant nodes to remove single point of failure [3].

As mentioned earlier, high rate of attacks against the meters (due to large scale deployment) is a challenge. Attacks against IDS sensors could be limited if the sensors are isolated through virtualization, or their hardware is separated from the meter itself, or use a different network than the one used by the meters to communicate among themselves. In a distributed environment, false positives can produce large overhead on the utility server as there could be millions of nodes. Therefor there should be a mechanism to address this problem. Using a reputation system to evaluate how much the report of node can be trusted could be one solution to this problem as suggested in [8].

The volume of generated alerts is an important problem for IDS applied to smart metering systems. It is essential to have an alert management system. This alert management system should reduce the volume of information by aggregating the alerts. This means that the alerts that share similar attributes should be grouped together. Also, the alert management center should be able to extract correlation information about the alerts. This means that if, for example, two alerts potentially have similar root causes, it should be identified. This can be done through techniques discussed in [11], and through having correlation rules which consider time, location, sequences and other properties of alerts.

4.1.3 Specification-based IDS

In [5] it is argued that specification-based intrusion detection system is a better fit for the domain of smart meters compared to other techniques. In [6] a specification-based network intrusion detection system for smart metering systems is proposed. This intrusion detection system is not general, covering all the aspects of smart metering system. It is a monitoring system that provides security for the communication of smart meters. Security requirements for this IDS are extracted based on: 1) Threat model and analysis of communication protocol and expected behavior of the system. 2) Historical training data from previous use cases. The work in [6] focuses on the communication security and discusses common standard protocols used for smart metering infrastructure, namely ANSI C12.22 and ANSI C12.19. The first one is the protocol to transfer predefined data tables (usage table for instance) over a reliable network like TCP/IP, and C12.19 is the protocol to define such data tables.

In [6], specifications of smart metering systems are defined. These specifications represent how the system is supposed to work. If the behavior of the system is violating these specifications, then a security problem has occurred. To model these specifications, the behavior of the system in represented through state machine in different levels of network, device, and application. For example, at the device level, three states for the device is considered. These states are offline, in-use, and to-configure. Each of these states entail some specifications and constraints for the device and help the monitor to make sure of the activity of the device is valid or not. For instance, if a system is in to-configure state, it should not initiate any communications with other meters. To create the model of the meter, a set of constraints are defined for the system. The constraints cover three classes of: network, device, and application. The constraints for each of these classes can have 5 different types of: data, access, and timing, resource usage, and operational. These constraints are identified through analysis of system behavior or historic data. For instance, through analysis of network trace files, it is concluded that for the use case of meter reading, reading request for multiple intervals occurs 25 times per 1000 meters per day and the response time is less than 15 sec. for requests.

In [6], a formal verification framework is built for the specification-based network intrusion detection system. The idea of formal verification of the intrusion detection system is to build a model of the security features of the system (based on the defined constraints) and then showing that no network trace can violate these constraints undetected. For example, the detector checks if action a is performed before action b in the traces if such a sequence is not possible based on the security constraints. A set of rules and some procedures are defined to verify these rules in the network traces. If these rules are verified, then we can conclude that the security policy is not breached.

4.1.4 Discussion

In this section, we discussed the requirements for building intrusion detection systems for smart meters and existing works on such systems. Here, we discuss the gaps that exist in the current works for IDS on the smart metering domain:

- The current work on intrusion detection system for smart meters is mostly focused on network intrusion detection systems. In [5], some guidelines to build intrusion detection systems for advanced metering infrastructure are proposed but there no architecture for host-based intrusion detection systems is discussed. Network Intrusion detection systems by themselves, cannot fully secure smart meters, as they may have false negatives that allow attackers to bypass the security mechanism by exploiting software vulnerabilities. Therefore, protecting the meter at the software level is a necessary complement to network intrusion detection systems, which is a domain that is still unexplored.

- Developing host-based intrusion detection systems (HIDS) for smart meters is a challenging task for two main reasons: 1) they impose a high performance overhead on the machine they are monitoring, 2) They are normally easier to compromise compared to network IDS. In [25], leveraging virtualization to solve the second problem is discussed. But leveraging virtualization still does not solve the problem with the overhead of host-based intrusion detection systems.

- There is no formal model to extract and define security policies for the case of smart meters. For instance, in [6], verifier procedures are defined which check if the security policies are violated or not. But there is no guarantee that the security policies are well-defined. Therefore, defining security policies or validating a security policy for smart metering systems is still an unexplored but important problem.

4.2 Remote-Attestation-Based Techniques

4.2.1 Definition and Background

Remote attestation is the process through which, an application is authenticated for a remote party. It its simplest form, when an application is asked to authenticate itself, it asks the operating system to endorse it. The OS creates a hash of the application, signs it, and the entire certificate chain is sent to the remote party. In the attestation process, the client and server must share a secret key, otherwise, the session can be hijacked. Today's techniques for remote attestation are mostly relied on challenge-response protocol. In this protocol, a verifier sends a challenge in the form of a nonce to the target device, and then the device uses a predetermined verification procedure to compute a response to this challenge. The device sends this response to the verifier and then the verifier checks it see of it is correct or not.

The motivation behind performing remote attestation is that many attacks are performed through inserting malicious code into the target system remotely. For example in 2008, unauthorized code was inserted into the servers in the branches of a super market chain in US [21]. The code gathered credit card data for transactions and periodically submitted the information to a third party server. As a result of this attack, over 4,200,000 credit and debit cards were compromised. Therefore, the administrators of a system are interested in being able to verify the integrity of a system and make sure that the codes running on a system are legitimate codes and not malicious codes.

Trusted Computing is the building block of performing remote attestation. The main goal of trusted computing is to make sure that the software is running as expected and unmodified. Trusted Computing Group (TCG) has introduced Trusted Platform Module (TPM) and the concept of remote attestation [32]. Trusted Platform Modules are chips on the platform device. They have Platform Configuration Registers (PCR) in which data regarding the state of hardware or software can be stored. TMP contains cryptographic keys to sign its messages. A *quote* is a digitally signed message of the PCR content. Therefore, the receiver of the TPM message can verify that the message content is not tampered with. Through this, TPM can provide facts about the state of hardware and software, digitally sign them, and send them to any receiver. Using a chain of reasoning that starts from TPM at the boot time, the users can be sure of the state of a machine from hardware layers to the software layers. TPM has the hash of the BIOS, and at the boot time, control is passed to the TPM and it recomputes the hash of BIOS and compares it to the value it already has. If verified, control is passed to BIOS and it does the same process with the next component. Through this process, a chain of verification is performed through all components up to the software layer. All the messages are encrypted and decrypted using public key/private key pairs.

To be applicable for smart metering systems, remote attestation techniques should be 1) light-weight: so that they do not impose high overhead on the meter, 2) scalable: so that they will be practical when the number of nodes increases to millions 3) effective in keeping the system secure as there will be no user constantly monitoring the system (unlike conventional computer systems). We categorize remote attestation techniques into two groups of software attestation and behavior attestation. In the first category, the remote verifier can make sure that the software running on the meter is the legitimate software. But it cannot guarantee that any vulnerability in the software has not been exploited by an attacker. In behavior attestation, the remote verifier can monitor the behavior of the meter and take actions if suspicious activity is observed.

4.2.2 Software Attestation

In this category, a remote verifier can conclude whether or not the software running on the system (smart meter) is legitimate or not. The developer of this system can tailor it for the case of smart meters to build a light-weight system that does not impose high overhead on the meter which is an important consideration. But on the negative side, this technique does not guarantee that any potential vulnerability on the meter software will not be exploited by an attacker. Software attestation techniques are mostly based on challenge/response protocol [36, 37, 13] where a verifier sends a nonce to the device and receives a response to that nonce. Then the verifier can check the correctness of the response.

In [37] a one-way memory attestation technique called OMAP is proposed. In OMPAP, the smart meter sends checksums of randomly selected regions of memory to the utility server. The server knows how the meters calculates the checksum and therefore can verify if the memory is modified or not. One-way memory attestation will reduce the chances of performing man-in-the-middle attacks against the smart meters. Based on this, OMAP consists of three steps: checksum generation, checksum transmission, and checksum verification. Checksum generation is done using a time-based seed to prevent guessing attacks. The smart meter generates a seed using a hash function and using time and serial number of the meter as parameters. Then, using a pseudo-random number generator, the meter selects a memory range to calculate the checksum. After that, hash of the checksum is transmitted along with the selected times to the verifier. The verifier has the memory content of the smart meter and uses the same procedure to recalculate the checksum, and finally compares the results with the received value. It is important to note that the start time of checksum computation is included in the message sent to the verifier to prevent impersonation attacks. Increased calculation time can be a sign of modified meter. This means that if the meter does not send the checksum within a given time, the verifier assumes that it has been attacked. Although in practice, network latency can cause problems for adopting this concept.

A two-way attestation protocol called Pioneer is proposed in [36]. Pioneer works based on a verification function. The verification function basically performs a checksum over the code and makes sure that the code is not changed. The high level steps are as follows. The trusted computing machine, called dispatcher, sends a challenge message to the untrusted platform. The platform computes a checksum over the verification software and sends it to the dispatcher. Then it computes the hash of the executable code and again, sends it to the dispatcher. Finally, it executes the code and sends back the results to the dispatcher. To make sure that the checksum code is executed correctly on the untrusted platform, the authors explain the types of attacks that could happen and how their system blocks them. For instance, the adversary might be able to precompute the checksum. In order to block this attack, the checksum is depended on the random challenge initially sent from the dispatcher. Also, the adversary might run another code after the checksum is computed and change the memory values. In order to prevent this, the system designers make sure that the interrupts are disabled by including the flag registers into the checksum. Also, the authors show that if the adversary tries to run another checksum code from the beginning, the process is going to make more time. They achieve this by including cpu state in the checksum calculation and also doing pseudo-random memory traversal. Applying this technique to the smart meters could be very challenging as one server has to constantly verify the integrity of thousands of smart meters.

Imposing low overhead is an important consideration for any attestation mechanism applied to smart metering systems as smart meter devices have limited computing resources. In [13], SMART: a light-weight and minimal architecture for establishing dynamic root of trust in low-end embedded systems, is introduced. SMART has two components called prover and verifier. Prover is the component that needs to be authenticated and verifier is the component that verifies the authenticity of prover. SMART has three security objectives which are: 1) verifier component obtains authentication of prover 2) verifier is assured that any memory segment on the prover contains the expected content 3) verifier is assured the code on prover is executed. At the beginning, verifier sends some parameters to prover: attestation region, a nonce to prevent replay attacks, and a memory location x for the prover to jump the control to after attestation. Prover calculates checksum of the specified region and passes the control to location x and runs the code located there. After that, it returns the checksum (implemented as HMAC:hash-based message authentication code) to the verifier which recalculates the checksum and compares it with the received value. The main goal of SMART is to make sure that the code is running.

Although remote software attestation techniques are effective in terms of providing low-overhead attestation services for smart meters, they do no provide any security guarantees for the system. In other words, behavior of the meters are unmonitored and therefore, an attacker who has exploited a vulnerability of the meter software can remain undetected.

4.2.3 Behavior Attestation

Behavior attestation techniques monitor the behavior of the software running on the remote device. This is important for the smart meters since in the absence of other monitoring techniques and a user constantly working with the system, this can enhance the security of the meter. Any vulnerability on the original software running on the smart meter can be exploited by an attacker. In this case, software attestation might result in verifying the integrity of the system while the system has actually been compromised. Therefore, behavior attestation of the smart meters is an important approach. In this technique, the semantics and features of the system to be monitored are modeled. The meter has to record the previously defined events on the system and periodically (along with other verification information) submit them to the verifier. In the following we briefly explain some of the systems in this group.

In [17] a semantic-based remote attestation technique is proposed. In this technique, the assumption is that the application to be monitored is platform independent and is running on a virtual machine (for instance Java Virtual Machine). In this technique, they propose to capture and attest behavior of the application rather than simply verifying the executable. To do this, the high level semantics of the code is analyzed and the claim is that since the code is running on a virtual machine and is written in a platform independent and high-level manner, the task is simpler. Also, the virtual machine can monitor the dynamic behavior of the application. This behavior is checked against some predefined security policy to make sure that the application is running correctly and safely. This dynamic monitoring is done through the lifetime of the application and on specific points in time. The virtual machine itself is verified through normal hash-based techniques.

In [38], BTRAM (Behavior based Remote Attestation Model) is proposed. In BTRAM, the behavior of the application is defined and classified into two categories of system behavior (for OS) and application behavior (for user-launched applications). To model the behavior of the application, several attributes and values are defined. These attributes include auto-transmitting (infecting files), auto-activating (registering in startup items), self-protecting (hiding directories) and etc. The trustworthiness of an application is evaluated based on how the values of its attributes are compared to the expected values. The main components of BTRAM are User (U), Verifier Proxy (VP), and Access Object (AO). The User initiates the access request which is the verification request to the Verifier. The User has to dynamically report the behavior of the system to the Verifier Proxy. Verifier Proxy itself has several components. It analyses the behavior data according to some policy and decides if it is trusted or not. Access Object does policy maintenance and updates the policy for Verifier Proxy whenever necessary.

Although behavior attestation techniques are able to detect suspicious activity of smart meters, the downside of these techniques is that they puts extra overhead on the system to be monitored since they record detailed activities and events of the system compared to software attestation techniques. This limits the applicability of these systems to smart meters which are limited in terms of computing resources.

In Table 1 we have brought the summary of the strengths and weaknesses of existing security classes for smart meters which we discussed here. In this table, letter 'Y' indicates that the security technique is capable of providing the specific requirement and letter 'N' indicates otherwise.

4.3 Discussion

To conclude this section, we list the downsides and gaps in the applicability of existing remote attestation techniques for smart meters:

- The main issue with the proposed remote attestation techniques in literature is the scalability problem. Smart meters are deployed in large scale (several million nodes) and therefore, any security technique developed for them must be highly scalable. In works such as [36, 38, 37, 17], there is a verifier that has to communicate with the device and run a procedure (sometimes the same procedure running on the meter) to verify the response from the meter. Considering the number of the meters to be monitored, this is not a practical approach in real scenarios. Having a highly distributed verification architecture might be a possible solution to be explored. But to the best of our knowledge, no distributed verification architecture for addressing this issue in the domain of smart meters has been proposed yet.

- Remote software attestation techniques do not guarantee security of the smart meters. Smart meters are new devices and mature security architecture has not been developed for them yet. As a result, many vulnerabilities and attacks are discovered against them [35, 27, 14, 39]. Therefore, making sure that the software running on the meter is not modified does not guarantee that the meter does not have vulnerabilities that can be exploited by the attackers.

- Behavior-based attestation techniques are not accurate. Smart metering systems can only afford very low false negative rate due to the scale of the system and the overhead of monitoring. This weakens the applicability of these techniques.

- Upgrades and patches of the meter software can be problematic for remote attestation techniques as the verification process must be modified to adapt to the new software. Since smart meters are new systems, we are not aware of the frequency of software updates in the long term for these systems yet.

5. PRIVACY

The current architecture of smart grids have serious privacy issues [2]. The meters record fine-grained measurements and transmit them to a database at a utility server. It has been recognized that detailed consumption data are private information and can lead to leakage of information such as the devices being used at homes [18, 24]. These information can be used to build a profile of customer behavior. Information such as if the users are at home, when they come back from work or when they eat can be extracted from the consumption profiles. Bohli et al [7] were the first to propose a solution were the electricity service providers were not aware of the up-to-date consumption data of the individual customers but a group of them. In their solution, there is a trusted third party proxy that is involved in meter reading from individual customers and aggregates data. Also they

	Scalability	Low overhead	Host attack detection	Network attack detection	Software Integrity
Network IDS	Y	Y	N	Y	N
Host IDS	Y	N	Y	N	N
Software Attestation	N	Y	N	N	Y
Behavior Attestation	N	N	Y	N	Y

Table 1: Summary of strengths and weaknesses of existing security classes for smart meters

propose to add random values to data to preserve the privacy of individual customers. Garcia et al [15] propose to use homomorphic encryption to prevent electricity service providers from accessing to consumption data of individual households.

In the domain of smart metering systems, the electricity service provider prefers to receive as much data as possible. Service provider needs current consumption data for planning purposes as well as providing accurate and authentic billing. For the provider, the correctness of the calculated bills are the most important issue. From the customer's point of view, anonymity is an important issue. They do not want anyone, even the service provider to be able to link the consumption data back to individual customers or create any profile based on customer data that reveals information.

To evaluate privacy threats against smart meters, it is important to identify data types that exist in a smart metering workflow. Marek et al [19] recognize the following data types as the major data types in a smart metering system:

- Contact details: for identifying the customers and sending invoices.
- Billing details: for directly collecting payments from customers' banks.
- Measurements: periodically collected from meters.
- Payment records: history of payments from customers.
- High-resolution measurements: Real-time consumption data recorded by the meters.
- Smart appliance information: are seen by the smart meter when interacting with the appliances. The usage times and patterns are seen by the meter.

Privacy loss associated with leakage of these data items can have different impacts. Data items can be static or dynamic. Static data items are the ones that do not change that much and therefore do not leak a lot of information. Dynamic data items are the ones that change over time. For example high-resolution measurements can reveal the usage pattern of the customers. From another perspective, data items can reveal explicit information or implicit information. For instance, the fact that the dryer was started at 2pm is an explicit information. But the increase in the usage at a specific time interval can be implicit information and be used to infer information such as the fact that the customers were at home during that time interval.

For protecting the privacy of customers, in [19], the degree to which data should be revealed to different parties in a smart metering system is discussed and it is recommended that data be revealed to a party only when necessary. Data items can be necessary for some parties, be only necessary in aggregate form, or only necessary in anonymized form. For example contact details of the users are necessary information for service providers. Consumption data are only necessary in aggregated form for billing purposes. Also, the grid operator for instance does not need to know which consumer produces which bottleneck, the only need to know the sum of the load on different segments of the grid, so this information can be anonymized.

We categorize techniques providing privacy for customers in a smart grid into two major classes. The first class of techniques focus on the architecture of smart grid [33, 12] and we call them architect-based techniques. In this class, the current architecture of the grid is modified and a trusted third party (not the service provider) is introduced which hides the information from the service provider. In the other class of techniques, the architecture of the smart grid is not changed, but either data is modified locally (anonymized) or new set of protocols are defined to address the privacy of customers [15, 34, 20]. In the next sections we study these two approaches.

5.0.1 Architecture-Based Techniques

In architecture-based techniques, the architecture of the smart grid is modified to address the privacy of the users. It is argued that the current architecture of the grid does not take the privacy of customer's into account and reveals information by sending unnecessary data to the utility server [33, 12].

In the modified architecture, the smart meters of the households are connected to the site current transformer (CT) in a specific topology (a star topology for instance [33] and use Powerline Communication (PLC), WiMAX, DSL, or etc. as a shared broadcast medium according to the existing infrastructure. The current transformers are connected to a switchyard and the switchyards are connected to the internet backbone. The switchyards act as proxies between the households and the service provider. In [33] it is proposed that a collector component in the switchyard can submit the consumption data to the service provider with its own IP address (and not the households IP address) so that the server cannot identify individual customers based on their IP addresses.

For the above architecture to work, a Trusted Third Party (TTP) is necessary. This trusted party is in charge of validating the identity of the meters. In [33] it is suggested that each smart meter be equipped with an Endorsement Key (EK) certificate and the trusted third party has access to this data. When a new smart meter is installed, the trusted third party has to verify it first. To do so, the smart meter sends its endorsement key certificate and personal data of the household to the trusted third party. This data is encrypted with the third parties public key. In response, the TTP sends the meter a unique identifier. This message is encrypted with the endorsement key of the meter so only the meter can read it. The meter does the communication using this identifier. In [12], it is suggested that the meters have two IDs, one public and one private ID. The private ID is only submitted to the trusted third party and no one else. The reason behind having two IDs is that, in [12], two types of data in a smart metering system are identified: high-frequency data, and low-frequency data. High-frequency data are the ones that are submitted to the server in short periods of time (every few minutes for instance). Low-frequency data on the other hand are scarcely transmitted to the server (every few weeks or months). It is argued that high-frequency data can be used to extract private information regarding user behavior. The problem with this arrangement is that if it would be difficult for the service provider to authenticate the meters. Therefore, the trusted third party should be aware of the HFIDs. The trusted third party can be the manufacturer of the meter. High-frequency data are sent to the trusted third party. The third party verifies the HFID and aggregates data if necessary, and submits them to the utility server.

Either the meter data are submitted to the server with a different source address or with a different ID, the source of information is hidden from the service provider. But these techniques require modification of the smart grid architecture which, considering the size of the network, are very expensive. Also, these techniques are delegating the trust to a third party which is not a promising approach. This means that users still have to be able to trust an entity with their private data.

5.0.2 Protocol-Based Techniques

Another class of work focuses on modifying protocols and calculations performed in the smart grid to protect the privacy of customers without modifying the architecture of the smart grid. Considering the scale of the grid and the cost of architectural modification, these techniques are worth attention. In this approach, either the communication protocol with the server is modified [15, 34], or data is modified locally [20] to prevent transmission of unnecessary information.

Rial et al [34] propose a protocol between the smart meter, user and the service provider to preserve the privacy of the users while allowing the service provider perform the necessary tasks such as billing activities. In their architecture, there are three components: the smart meter which produces consumption data, a service provider that establishes a pricing policy and specific periods of time requests the user to pay the bill, and user that receives consumption data from the meter and pays the bills to the service provider. Bills are calculated based on a public pricing policy which takes consumption data along with some other information (such as time of use) and outputs the total price. The basic flow of the operations is as follows. The service provider sends the user a pricing policy. During a billing period, the smart meter outputs the consumption data along with other necessary information (such as consumption times) to the user, the user calculates the bill based on the pricing policy and sends the proof of correct calculation to the service provider. This procedure prevent any leakage of information regarding detailed consumption data from the user side to the service provider. They use homomorphic commitment schemes to construct proofs for the service provider that they have used data from the meter and the policy from the provider to calculate the bill. They send the proof to the service provider so that it can verify the calculation.

In [20], load signature moderation is proposed to hide information regarding the consumption patterns of the users. Load signature is defined to be a series of time-stamped average power loads derived from energy values at short time intervals. From the privacy point of view, load signatures can be used to extract information about users activities (for example if they are home or not or if extra devices have been turned on or not). Load signature moderation is defined as reshaping technique for load signature with which, activities of appliances can be hidden (for instance by smoothing out the load signatures). The key assumption here is that the users have access to energy storage and energy generator devices. This means that they have devices that can release stored power so that the users do not have to consume power generated through the service provider. An also, the device can recharge itself through the service provider when the consumption is low. In [20], this component is called Load Signature Moderator (LSM) and is in charge of detecting privacy threats and smoothing out data whenever necessary.

The downside of protocol-based techniques is that they either need extra equipments on the user side [20] or add extra overhead to the system by introducing cryptographic computations on the smart meters [15, 34]. This is very important since the expected service time of the meters is expected to be around 20 years [1] and running state of the art cryptographic algorithms would be a challenge for smart meters over time.

6. CONCLUSION AND DISCUSSION

In this survey, we discussed the security and privacy issues of smart meters. We classified the existing mechanisms developed to address security of smart meters into two categories of intrusion-detection-based, and remote-attestation-based techniques. We analyzed each of these classes and discussed their strengths and weaknesses. Also, we studied the threats regarding the privacy of smart meters and classified them into architecture-based and protocol-based techniques. We studied each group discussed their pros and cons.

Smart grid, and as a result smart meter, is a new technology which is quickly being deployed around the world. As we discussed in previous sections, there are many unexplored problems in this domain. Here we provide the summary of the most important problems that exist in this field which the existing works have not explored yet:

- Existing work does not provide any analysis on the software running inside the smart meter. There is no model for analysis of the attacks that can potentially target the meter software. Existing work on building models and systematic techniques to provide security for smart meters either target the network communications of smart meters, or testing of the meters based on existing generic attacks. For example, Berthier et al [6] model the normal behavior of the communication of the meters and propose an specification-based intrusion detection system (IDS) based on their model. But this does not cover the vulnerabilities of the software running on the meter. In [29] a systematic way to perform penetration testing for the meters, given the attack models is proposed. However, there is no well-defined mechanism to build the attack models.

- Other than analyzing the attacks against the smart meter software, there is no work done on providing a monitoring/protection system for the software running inside the meter. Current intrusion detection systems (IDS) for addressing security of smart meters are mostly limited to security of the communication link. Generic security mechanisms such as host-based intrusion detection systems incur high performance overheads, making them unsuitable for meters [23, 22]. Therefore security mechanisms for smart meters must be carefully tailored to specifically target the attacks associated with advanced metering infrastructure to comply with the constraints and requirements of these systems. This domain has not been addressed in the literature yet.

- The service time of smart meters is long (about 20 years [1]). Many of the existing security techniques applied to the domain smart meters rely on running cryptographic algorithms on the meters. Old smart meters might not have the processing power and adequate memory to perform intense cryptographic operations. Analysis of the security of smart meters considering technology advances over time is an important research problem to investigate whether the existing security techniques will still hold in the time span of 20 years or not.

7. REFERENCES

[1] Private communication with BCHydro's engineers.

[2] Anderson, R., and Fuloria, S. On the security economics of electricity metering. *The Ninth Workshop on the Economics of Information Security*.

[3] Balasubramaniyan, J. S., Garcia-Fernandez, J. O., Isacoff, D., Spafford, E., and Zamboni, D. An architecture for intrusion detection using autonomous agents. In *Proceedings of the 14th Annual Computer Security Applications Conference* (Washington, DC, USA, 1998), ACSAC '98, IEEE Computer Society.

[4] Bc hydro home page. `http://www.bchydro.com/energy_in_bc/projects/smart_metering_infrastructure_program.html`.

[5] Berthier, R., Sanders, W., and Khurana, H. Intrusion detection for advanced metering infrastructures: Requirements and architecture directions. In *Smart Grid Communications (SmartGridComm), 2010* (2010), pp. 350 – 355.

[6] Berthier, R., and Sanders, W. H. Specification-based intrusion detection for advanced metering infrastructures. *Pacific Rim International Symposium on Dependable Computing, IEEE 0* (2011), 184–193.

[7] Bohli, J.-m., Sorge, C., and Ugus, O. A privacy model for smart metering. *Computer* (2010), 1–5.

[8] Buchegger, S., and Le Boudec, J.-Y. Performance analysis of the confidant protocol. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing* (New York, NY, USA, 2002), MobiHoc '02, ACM, pp. 226–236.

[9] Carpenter, M., Goodspeed, T., Singletary, B., Skoudis, E., and Wright, J. Advanced metering infrastructure attack methodology. `http://inguardians.com/pubs/AMI_Attack_Methodology.pdf`.

[10] Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S., Koscher, K., Czeskis, A., Roesner, F., and Kohno, T. Comprehensive experimental analyses of automotive attack surfaces. In *Proceedings of the 20th USENIX conference on Security* (Berkeley, CA, USA, 2011), SEC'11, USENIX Association, pp. 6–6.

[11] Debar, H., and Wespi, A. Aggregation and correlation of intrusion-detection alerts. In *Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection* (London, UK, UK, 2001), RAID '00, Springer-Verlag, pp. 85–103.

[12] Efthymiou, C., and Kalogridis, G. Smart grid privacy via anonymization of smart metering data. *Distribution* (2010), 238–243.

[13] Eldefrawy, K., Francillon, A., Perito, D., and Tsudik, G. Smart : Secure and minimal architecture for (establishing a dynamic) root of trust. In *19th Annual Network and Distributed System Security Symposium, February 5-8, San Diego, USA* (San Diego, UNITED STATES, 02 2012).

[14] Fehrenbacher, K. Smart meter worm could spread like a virus., 2010. `http://earth2tech.com/2009/07/31/smart-meter-worm-could-spread-like-a-virus/`.

[15] Garcia, F. D., and Jacobs, B. Privacy-friendly energy-metering via homomorphic encryption. In *Proceedings of the 6th international conference on Security and trust management* (Berlin, Heidelberg, 2011), STM'10, Springer-Verlag, pp. 226–238.

[16] Goodspeed, T., Highfill, D. R., and Singletary, B. A. Low-level design vulnerabilities in wireless control system hardware. In *Proceedings of the Scada Security Scientific Symposium (S4)* (2009).

[17] Haldar, V., Chandra, D., and Franz, M. Semantic remote attestation: a virtual machine directed approach to trusted computing. In *Proceedings of the 3rd conference on Virtual Machine Research And Technology Symposium - Volume 3* (Berkeley, CA, USA, 2004), VM'04, USENIX Association, pp. 3–3.

[18] Hart, G. W. Nonintrusive appliance load monitoring. *Proceedings of the IEEE 80*, 12 (Aug. 2002), 1870–1891.

[19] Jawurek, M., and Ferling, F. C. Privacy threat analysis of smart metering.

[20] Kalogridis, G., Efthymiou, C., Denic, S. Z.,

Lewis, T. A., and Cepeda, R. Privacy for smart meters: Towards undetectable appliance load signatures. *2010 First IEEE International Conference on Smart Grid Communications* (2010), 232–237.

[21] Kerber, R. Advanced tactic targeted grocer: 'malware' stole hannaford data. Boston Globe.

[22] Khurana, H., Hadley, M., Lu, N., and Frincke, D. A. Smart-grid security issues. *IEEE Security & Privacy* (2010), 81–85.

[23] Koller, R., Rangaswami, R., Marrero, J., Hernandez, I., Smith, G., Barsilai, M., Necula, S., Sadjadi, S. M., Li, T., and Merrill, K. Anatomy of a real-time intrusion prevention system. In *Proceedings of the 2008 International Conference on Autonomic Computing* (Washington, DC, USA, 2008), ICAC '08, IEEE Computer Society, pp. 151–160.

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

[25] Laureano, M., Maziero, C., and Jamhour, E. Intrusion detection in virtual machine environments. In *Proceedings of the 30th EUROMICRO Conference* (Washington, DC, USA, 2004), EUROMICRO '04, IEEE Computer Society, pp. 520–525.

[26] LeMay, M., Gross, G., Gunter, C. A., and Garg, S. Unified architecture for large-scale attested metering. In *Proceedings of the 40th Annual Hawaii International Conference on System Sciences* (Washington, DC, USA, 2007), HICSS '07, IEEE Computer Society, pp. 115–.

[27] Lewson, N. Smart meter crypto flaw worse than thought, 2010. `http://rdist.root.org/2010/01/11/smart-meter-crypto-flaw-worse-than-thought`.

[28] McDaniel, P., and McLaughlin, S. Security and privacy challenges in the smart grid. *IEEE Security & Privacy* (2009), 75–77.

[29] McLaughlin, S., Podkuiko, D., Miadzvezhanka, S., Delozier, A., and McDaniel, P. Multi-vendor penetration testing in the advanced metering infrastructure. In *Proceedings of the 26th Annual Computer Security Applications Conference* (New York, NY, USA, 2010), ACSAC '10, ACM, pp. 107–116.

[30] Patel, A., Qassim, Q., and Wills, C. A survey of intrusion detection and prevention systems. *Information Management Computer Security 18*, 4 (2010), 277–290.

[31] Paulo Veríssimo, Nuno Ferreira Neves, M. C. Crutial: The blueprint of a reference critical information infrastructure architecture. In *Proceedings of the 1st International Workshop on Critical Information Infrastructures @ ISC06* (Aug. 2006).

[32] Pearson, S. *Trusted Computing Platforms: TCPA Technology in Context*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2002.

[33] Petrlic, R., and Paderborn, U. A privacy-preserving concept for smart grids. *Sicherheit in vernetzten Systemen 18 DFN Workshop* (2010).

[34] Rial, A., and Danezis, G. Privacy-preserving smart metering. In *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society* (New York, NY, USA, 2011), WPES '11, ACM, pp. 49–60.

[35] S. Brinkhaus, D. Carluccio, U. G. B. J. D. L. C. W. Smart hacking for privacy. In *28th Chaos Communication Congress* (Berlin, Germany, DEC. 2011).

[36] Seshadri, A., Luk, M., Shi, E., Perrig, A., van Doorn, L., and Khosla, P. Pioneer: verifying code integrity and enforcing untampered code execution on legacy systems. In *Proceedings of the twentieth ACM symposium on Operating systems principles* (New York, NY, USA, 2005), SOSP '05, ACM, pp. 1–16.

[37] Song, K., Seo, D., Park, H., Lee, H., and Perrig, A. Omap: One-way memory attestation protocol for smart meters. In *Proceedings of the 2011 IEEE Ninth International Symposium on Parallel and Distributed Processing with Applications Workshops* (Washington, DC, USA, 2011), ISPAW '11, IEEE Computer Society, pp. 111–118.

[38] Wang N, W. Z. Q. C. H. F. *A remote attestation model in distributed environment*, vol. 1. 2010, pp. V1425–V1429.

[39] Zetter, K. Security pros question deployment of smart meters. *Threat Level: Privacy, Crime and Security Online* (March 2010).

[40] Zonouz, S., B. R., and Haghani, P. A fuzzy markov model for scalable reliability analysis of advanced metering infrastructure. In *IEEE PES Innovative Smart Grid Technologies Conference (ISGT)* (2012).