OMAP: One-way Memory Attestation Protocol for Smart Meters

Kyoungsub Song, Dongwon Seo, Haemin Park and Heejo Lee
Div. of Computer and Communication Engineering
Korea University
Seoul, Korea
{cadetks, aerosmiz, gaiger, heejo}@korea.ac.kr

Adrian Perrig
CyLab
Carnegie Mellon University
Pittsburgh, PA, USA
adrian@ece.cmu.edu

***Abstract*—A smart meter is one of the key elements of smart girds. An attacker can compromise smart meters by injecting malicious codes, and take financial benefits by modifying memory contents of the smart meters. An attestation scheme can prevent such a memory forgery attack as verifying memory contents. In smart grids, however, attestation processes are remotely performed through networks by a faraway utility. Therefore, attestation processes are exposed to network attacks such as man-in-the-middle (MITM) attacks. Even though existing attestation mechanisms detect local attacks such as the memory forgery, they are vulnerable to network attacks since they adopt a two-way attestation so-called a challenge-response protocol. In this paper, we propose a novel attestation mechanism, termed One-way Memory Attestation Protocol(OMAP), not only to detect local attacks, but also to defend against network attacks. Instead of using the two-way attestation, OMAP adopts an one-way attestation protocol; OMAP conducts a pre-defined internal algorithm, generates a checksum, and sends it to a verifier in one direction. Thus, OMAP does not require any information (e.g., challenges) from a verifier that can be exploitable by an adversary. In our experiments, as a smart meter scans only 0.004% of its memory, OMAP enables a verifier to detect memory modification with 95% probability if an attacker changes 20% of the memory.**

***Keywords*-Smart grid; smart meter; software-based remote attestation**

I. Introduction

Smart grids are systems that controls electricity in two-way communication between power utilities and customers. For efficient power generation and consumption, smart grids transmit sensitive information, such as metering and personal information. Therefore, smart girds can become attractive targets for attackers to gain benefits and to raise social chaos [1], [2]. A smart meter is one of the components of the smart grid, and plays a key role in data transmission between a user and a utility. Since the smart meter is located in the place where smart grid administrators cannot control properly (e.g., home), attackers can exploit smart meters easily. For example, by injecting malicious codes into the memory of a smart meter, attackers can spoof private information and launch network attacks, such as man-in-the-middle (MITM) attacks.

To detect and prevent such attacks, a utility has to verify the memory contents of a smart meter whether it is modified. Recent remote attestation schemes [3], [4], [5], [6] rely on a challenge-response protocol. A user receives a random challenge from a utility and returns a response to the challenge. They are resistant to local attacks. Local attacks mean that attackers modify a device's local resources, such as a memory and a CPU, so that they deceive a verifier, or quickly compute challenge-response pairs. However, the challenge-response protocol is exposed to network attacks that can occur between a smart meter and a utility during attestation procedures. Network attacks is that attackers modify attestation-related packets, so that they collect challenge-response information, or interfere attestation processes. Once such network attacks are successfully done to a victim, attackers can impersonate other legitimate users by spoofing attestation packets [7], [8]. A remote attestation protocol for smart grids requires to consider both local and network attacks.

In this paper, we propose a novel attestation scheme, termed One-way Memory Attestation Protocol (OMAP). OMAP not only detects local attacks by constructing a checksum using a random memory traversal, but also prevents from network attacks because its response (e.g., checksum) for the attestation is forwarded in one direction from a smart meter to a utility. That is, a smart meter using OMAP generates a checksum by randomly selecting specific ranges of a memory, and forwards the checksum to a utility. Because the utility decides how the smart meter generates the checksum, the utility can verify if the memory of the smart meter is modified.

In experiments, we show that OMAP can detect memory modification attacks assuming malicious code injection. By checking only 0.004% of a memory, OMAP detects the memory modification with 95% probability when attackers changes 20% of the memory. Moreover, the detection ratio is not relevant to the size of a smart meter's memory.

The remainder of this paper is organized as follows. In Section II, we briefly explain the concepts of smart grids, and a challenge-response protocol, which forms the basis of our approach. Section III introduces existing approaches. Section IV presents problem definition and assumptions. Section V describes how our proposed mechanism works. Then, experimental results are presented in Section VII.

978-0-7695-4429-8/11 $26.00 © 2011 IEEE
DOI 10.1109/ISPAW.2011.37

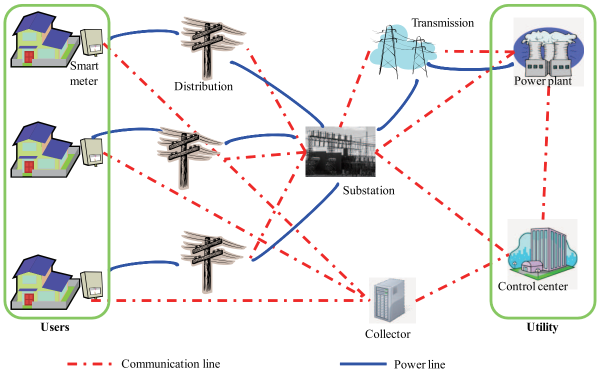

Figure 1. A smart grid environment

Finally, Section VIII presents our conclusion.

II. Background

A. Smart Grid System

In smart grid environment, a utility measures the power usage of customers, and has a function to be on strain during a time for peak demands. The utility can connect or disconnect electricity remotely. Users can confirm the usage of electricity and power rates. Moreover, they reduce electric usage by themselves during a peak energy use time for saving money. Two-way communication is a prerequisite for achieving this. As shown in Fig. 1, data for electricity usage from users is sent to a utility, the power rates information and the control commands from the utility are sent to homes and factories. The features of components in smart grids are detailed in the Table II.

Table I
THE SMART GRID COMPENENTS AND FEATURES

Components	**Features of each component**
Utility	·Generating electricity ·Providing electricity to users ·Controlling power systems ·Attesting smart grid components including smart meter
Substation	·Converting high-voltage electricity from power plants into lower-voltage electricity for homes or factories
Transmission	·Sending electricity to substation
Distribution	·Delivering electricity to a number of homes or factories
Smart meter	·Receiving the information for power rates and the control commands from the control center ·Sending the electricity usage data
Collector	·Delivering the data between a smart meter and utility

B. Advanced Metering Infrastructure and Smart Meter

An advanced metering infrastructure (AMI) is one of the major infrastructure in smart grids. The AMI provides information of energy usage (or demand) to utilities and consumers. As shown in Fig. 2, a smart meter exchanges data in real time through communication networks. Smart meters and communication networks provide AMI services. A smart meter in AMI performs four fundamental functions as follows [7]:

- Monitoring and recording of demand
- Logging of power relevant events(e.g., outages)
- Delivering usage and log information to the upstream verifier
- Delivering and receiving of control messages(e.g., controlling smart appliances, remote disconnect, etc.)

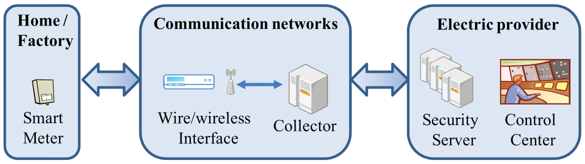

Figure 2. The communication network in AMI

C. Smart Meter Security

A smart meter is designed to help to deliver electricity more efficiently. However, an attacker gets malicious motivation due to smart meter's vulnerabilities. Cyber attacks in wired and wireless networks are also risk in smart meters. In particular, power relevant information and billing data are attractive target for attackers. Disaster such as blackout happens on account of cascading failure and malicious programs. Disaster as a low-probability event is considered serious problems.

As mentioned above, a smart meter is a very attractive target for a malicious attacker seeking to profit illegally, and the attack against the smart meter can be achieved easily. The attackers who compromise smart meters can immediately manipulate their power rates or forge generated smart meter readings. Such economic gain for attackers becomes a great motive (reducing electric rates) to an attacker [1].

By exploiting vulnerabilities of a smart meter, an attacker may attempt to modify the content of memory for the smart meter, or send forged control messages to other systems linked with smart meters. An attacker can also try a massive attack on smart grid communication networks by using worms that spread between smart meters. Bots, Distributed Denial of Service (DDoS) attacks and viruses on the Internet threaten smart meters in the future.

D. Remote Attestation

Existing researches for a remote attestation between a verifier and a device rely on a challenge-response protocol. Fig. 3 shows the overview of the challenge-response protocol. A verifier sends a random challenge as a nonce for the attestation to a target device, and then the device computes a response to this challenge using a pre-programmed verification procedure. When the device returns a response, the

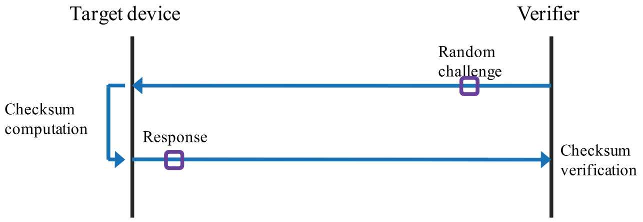

Figure 3. A challenge-response protocol

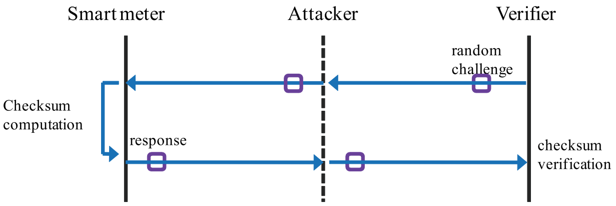

Figure 4. The man-in-the-middle (MITM) attack

verifier can examine the answer if its memory contents are correct, since the verifier can locally compute the answer to its challenge [3].

III. Related Work

We review existing studies indicating the tendency of remote attestation schemes. There are two types of attestation mechanism: hardware-based and software-based attestation mechanisms. We introduce hardware-based attestation mechanism and two existing software-based attestation mechanisms: a challenge-response protocol and Cumulative Attestation Kernel (CAK).

Table II
Types of attestation mechanism

Types of attestation mechanism	**Features of each mechanism**
Trusted Platform Module	·Tamper-evident hardware-based attestation ·Do not update software continuously
Challenge-response protocol	·Software-based attestation to verify modification of a memory
Cumulative Attestation Kernel	·Implemented at a low level in systems ·Do not consider secure communication with a verifier

- Trusted Platform Module: Hardware-based attestation mechanism uses TPM designed by Trusted Computing Group (TCG) for trusted computing [9]. It is the tamper-evident chip that allows the method for verifying platform information by attestation. TPM ensures that initial information stored in a memory by a manufacturer is not modified. However, hardware-based, such as TPM, is not able to be used in a smart meter. The smart meter required lifetime more than 10 years needs updates, but TPM cannot update its software continuously. Thus, the only one way to modify is to change the device. Software-based attestation mechanisms verify the integrity of a device without depending on the hardware; moreover, they do not require any additional hardware extensions. Cost, power, memory and computational limitations of a smart meter restrict the deployment of TPM for a smart meter [10].
- Challenge-response protocol: Several recent mechanisms utilize the challenge-response protocol based mechanism [3], [4], [5], [6]. Especially, Aakash *et al.* [6] deploy the protocol at remote terminal unit (RTU), one of the smart grid components. It detects the modification of firmware and memory of a target device with high probability using a pseudo-random memory traversal. However, as shown in Fig. 4, the two-way communication of the protocol embraces vulnerabilities to the network attacks such as a MITM attack.
- Cumulative Attestation Kernel (CAK): CAK [11] is a remote-attestation mechanism implemented at a low level in the embedded system. Its prototype is developed on a microcontroller typically used in smart meters. CAK provides cryptographically secure audit data for an unbroken sequence of firmware upgrade deployed on the embedded devices. CAK only focuses on the firmware integrity verification of the system; however, it does not consider secure communication with a verifier.

IV. Problem Definition

A challenge-response protocol is that a device computes a response to a random challenge and returns to a verifier in order to check the integrity of the device. By checking the memory contents of the device, a verifier detects the modification of a firmware or a target device with high probability. Therefore, many attestation mechanisms adopt this protocol to verify their target devices. These mechanisms are appropriate for the detection of memory modification locally. However, they do not consider the attacks on the communication network such as MITM attacks. In this section, we discuss feasible attack scenarios for a challenge-response protocol and describe our assumptions and problem statements.

A. Problems of a Challenge-Response Protocol

As shown in Fig. 4, an attacker can intervene between a smart meter and a verifier and intercept important data. This is an attack scenario known as MITM attacks.

The MITM attack can lead to two possible threats that can occur during attestation procedures: a rainbow attack, and an interference attack. The rainbow attack is that an attacker sends arbitrary challenges to smart meters as constructing a challenge-response pair table . An interference attack has the objective of disturbing benign users by inducing the smart meter to send miscalculated responses.

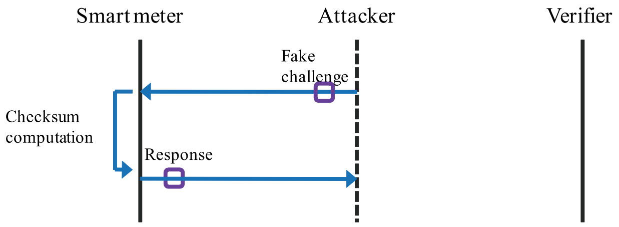

Figure 5. An attacker can launch a rainbow attack to construct a challenge-response pair table.

1) Rainbow Attacks: In a challenge-response protocol, a verifier sends a random challenge to a smart meter for checking integrity and receives a response to the challenge. In this procedure, by eavesdropping these challenges and responses, an attacker can infer attestation information, such as the length of a challenge. Therefore, the attacker can send fake challenges similar to verifier's challenges and receive responses to the fake challenges. The attacker can easily achieve the responses, since the smart meter does not check the genuinity of a challenge. Fig. 5 illustrates such a rainbow attack [12]. Through repeating these procedures, the attacker collects challenge-response pairs and eventually constructs a challenge-response pair table for conducting impersonation. While eavesdropping challenge-response pairs may take a long time to construct a table, the rainbow attack greatly reduces the construction time by forcing responses to a smart meter.

After constructing a challenge-response pair table, the attacker can impersonate another smart meter by sending a correct response to a verifier as shown in Fig. 6. The response is correct because it comes from a legitimate smart meter.

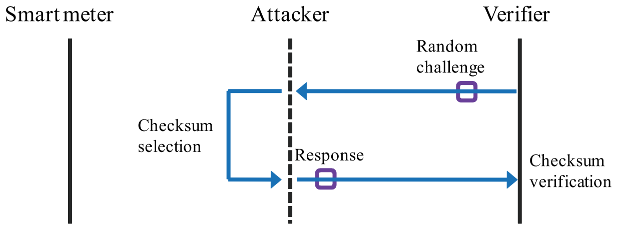

Figure 6. An attacker exploits the challenge-response pair table to impersonate other smart meters.

2) Interference Attacks: An attacker may attempt to simply interfere with the attestation processes of a smart meter through the MITM attack. Fig. 7 shows an example of an attestation interference attack against a smart meter. It immediately begins to compute a response as soon as a smart meter receives the challenge. If an attacker sends an another random challenge to the smart meter before the computation of the smart meter is complete or the smart meter sends a response, the smart meter computes the response to the challenge from the attacker again. Thus, the attestation for the smart meter is failed, because the response from the smart meter is not equal to the value that the verifier computes internally; moreover, an arrival time of the response is over a required time.

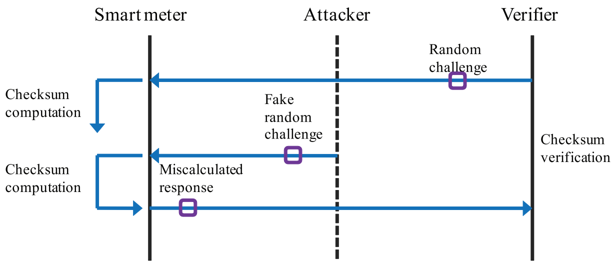

Figure 7. Interference attack causing an attestation failure

B. Assumptions

1) Verifier: We assume that a verifier knows the exact hardware specification and configuration of a smart meter such as a CPU model and a memory size, and maintains the precise memory copy of a smart meter. We also assume that a verifier cannot be compromised by the attacker.

2) Smart meter: We utilize the serial number for attestation. we assume that an attacker cannot modify the memory region containing the serial number even if an attacker compromises the smart meter. We also assume that a verifier sets a unique serial number for a smart meter. Our attestation mechanism utilizes the serial number to generate a checksum for smart meter attestation. This assumption is similar with that of [4]. Using a serial number prevents from the impersonation attack.

3) Attacker: We assume that memory contents of a smart meter can be read and written by an attacker. Therefore, the attacker can inject a malicious code in an empty memory region of a smart meter. Also, the attacker can eavesdrop all data transmitted over the AMI networks, and modify some of data and forward any data to a designated target (e.g., a smart meter or a verifier). However, we assume that an attacker cannot replace the hardware specification of a smart meter. For example, changing a BIOS of the smart meter, adding a memory, changing memory access timing, and increasing clock speed of processor do not occur by an attacker. We do not address physical attacks such as cutting the wire or delivering an electric shock.

C. Problem Statement

MITM attacks are feasible scenario in smart grids due to two-way communication vulnerability. attackers can impersonate a benign user. Furthermore, they can disturb the normal communication by making users send miscalculated values to the verifier. Therefore, we have to prevent from MITM attacks including rainbow attacks and interference attacks in smart grids.

V. OMAP: A One-way Memory Attestation Protocol

In this section, we discuss a remote attestation protocol, termed One-way Memory Attestation Protocol (OMAP). In order to attack a smart meter, an attacker has to inject malicious codes into the memory or falsify parts of firmware codes. Since injecting malicious codes and falsifying firmware codes modify the memory, OMAP can detect the modification of memory. We describe these procudures in following sections in detail.

A. OMAP Description

OMAP consists of three steps: 1) generating a checksum in a smart meter, 2) transmitting the checksum to a verifier, and 3) verifying the checksum by the verifier.

1) Checksum generation: The checksum is randomly generated to prevent from pre-computing or guessing the checksum. OMAP uses the time for generating a seed. That is, the checksum change from moment to moment.
2) Checksum transmission: The checksum is sent to a verifier in one way. A smart meter does not receive any challenges. This one-way communication prevents from MITM attacks.
3) Checksum verification: A verifier computes the checksum and compares the computation results with the checksum received from a smart meter. Since the verifier knows the contents and the structure of a smart meter [13], it can determine if the memory of the smart meter is modified.

Fig. 8 shows an overview of OMAP. We describe these procedure more detail.

B. Checksum Generation

The checksum generation procedure involves three steps: seed generation, memory address selection, message construction.

1) Seed generation: The smart meter generates a seed (W_k) using the hash function with a parameter as time (t) and the serial number (S/N). k increases from 1 to N which denotes the count of seed generation.
$$W_k = H(t_k, S/N)$$
2) Memory address selection: OMAP uses a pseudo-random number generator (PRNG) such as RC4 to collect memory addresses (A_k) randomly. RC4 stream cipher using PRNG takes 32 bits as an input and generates A_k, 32 bits memory addresses [14].
$$A_k = RC4(W_k)$$
3) Message construction: The smart meter reads memory contents (Q). $F_{ad}(A_k)$ is the function to read memory contents with amounts of what we need as the offset.
$$Q_k = F_{ad}(A_k)$$

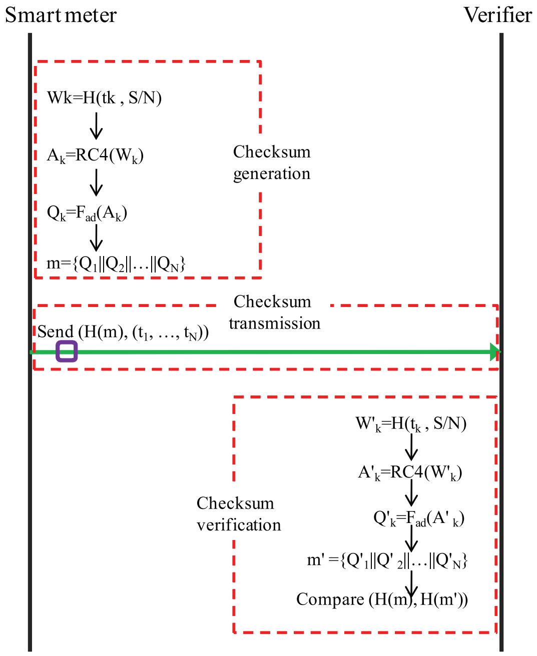

Figure 8. The overview of OMAP

The smart meter constructs a message m by concatenating Q_k in a generated order.
$$m = \{Q_1 \| Q_2 \| ... \| Q_N\}$$

C. Checksum Transmission

The smart meter obtains a checksum after checksum generation. By hashing m, $H(m)$, the smart meter sends it and the sequence of time,t_k to the verifier.Even though t_k is exposed to an attacker on the network, the attacker cannot use it for MITM attacks because the checksum is only used once.
$$Send(H(m), t_k)$$

D. Checksum Verification

The verifier conducts following steps in this procedure: 1) receiving the checksum from the smart meter, 2) generating the checksum by using same mechanism of the smart meter, and 3) comparing the checksum with that of the smart meter if memory contents of the smart meter are compromised.
$$Compare(H(m), H(m'))$$

After the verifier receives the checksum,$H(m)$ and time values, t_k, the verifier arrive at a conclusion through the comparison with $H(m)$ and $H(m')$. This step is possible because the verifier has the copy of the memory of the smart meter. Checksum generation procedure for the verifier is equivalent to the smart meter's procedure. OMAP ensures that the memory of the smart meter is not modified only if $H(m)$ is equals to $H(m')$.

VI. Security Analysis

We divide attacks into two parts for analysis :local attacks and network attacks. Local attacks, such as a checksum forgery attack and a parallel checksum computation attack, can be detected by existing attestation mechanisms, while the network attacks, such as a rainbow attack and an interference attack, cannot be detected by them. We describe how OMAP defends against local and network attacks in this section.

A. Local Attacks

1) Checksum Forgery Attacks: An attacker may attempt to compute a checksum on the memory beforehand. Before the smart meter sends the checksum, the attacker forwards this pre-computed checksum. The computation of checksum start time is included in the checksum to prevent from attacks. As an alternative way for attacks, an attacker can utilize data substitution which changes the location of a memory. When the defense mechanism investigates changed memory addresses, the attacker can divert a position in memory where it stores the original values. This attack can be detected by a pseudo-random pattern that is enabled by existing defense mechanisms. The attacker cannot predict in advance which addresses will be accessed by the defense mechanism. This approach is similar to the mechanisms in [3], [5].

2) Parallel checksum computation attacks: An attacker may attempt to speed up checksum computation in order to perform another illegal operation during extra time [4]. The way to speed up the checksum execution is to leverage several devices to compute the checksum in parallel. Then, an attacker combines the results to obtain the final checksum. We prevent this attack by addressing the checksum function non-parallelizable in order to force sequential execution. In OMAP, each procedure uses the result of previous procedures as an input, so that parallel checksum computing is impossible.

B. Network Attacks

In this section, we discuss network attacks between a smart meter and a verifier. An example of the network attacks is to construct a rainbow table as a preparatory step for impersonation. Another way is to interfere with smart meter attestation in order to deceive a verifier.

1) Rainbow attacks: In a challenge-response protocol, rainbow attacks can be occurred. An attacker can collect responses to the challenges by sending fake challenges. These combined responses are used to construct a challenge-response pair table. After achieving this attack, an attacker can impersonate legitimate smart meters. To address this problem, a smart meter sends a checksum to a verifier without a challenge request. An attacker eavesdrops all data between a smart meter and a verifier. In OMAP, the smart meter does not receive any challenges for attestation from the verifier. Instead, the smart meter sends a checksum in one way. The attacker cannot predict information which come from the smart meter because there is no clue, such as a challenge, a response. That is, collecting responses by sending fake challenges to the smart meter is impossible since OMAP does not return response to any challenges. Therefore, rainbow attacks cannot be conducted between smart meters deploying OMAP and the verifier.

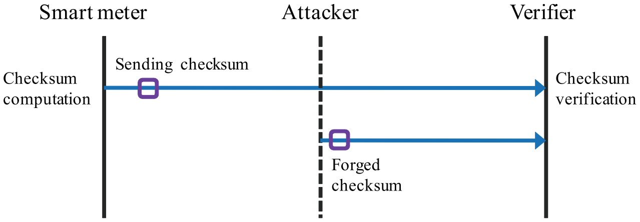

Figure 9. Interference attack against OMAP

2) Interference attacks: An attacker attempts to interference attacks in a challenge-response protocol by inducing a smart meter to transmit incorrect responses to a verifier. The verifier does not recognize that interference attacks are caused by the attacker since the checksum is made by a legitimate smart meter as shown in Fig. 7. Fig. 9 shows that the attacker forwards a forged checksum to the verifier, after the smart meter sends the checksum in order to launch the attack against OMAP. However, because the smart meter sends a correct checksum again, the verifier can recognize that the previous forged checksum is the fake checksum made by the attacker.

C. Attack against OMAP

If there is an attacker who perceives the smart meter deploying OMAP, the attacker can attempt the impersonation attack that pretends to be other smart meters by changing the serial number. The important data for security in OMAP is a seed including the serial number of a smart meter and time to start the checksum computation. If an attacker perceives all of that, the attacker can generate a correct checksum. Therefore, exposure of the seed is considered as serious vulnerability.

VII. Experiment and Results

In this section, We introduce experimental environment and show detection results against memory modification experimentally. Furthermore, we show that attackers can not achieve MITM attacks in our mechanisms theoretically.

A. Experiment Setup

1) Hardware Components: For experiments, we implement two components (a verifier and a smart meter) in a single computer; Intel Core 2 Duo CPU E6750 2.66 Ghz. Specification of components is not necessarily similar with

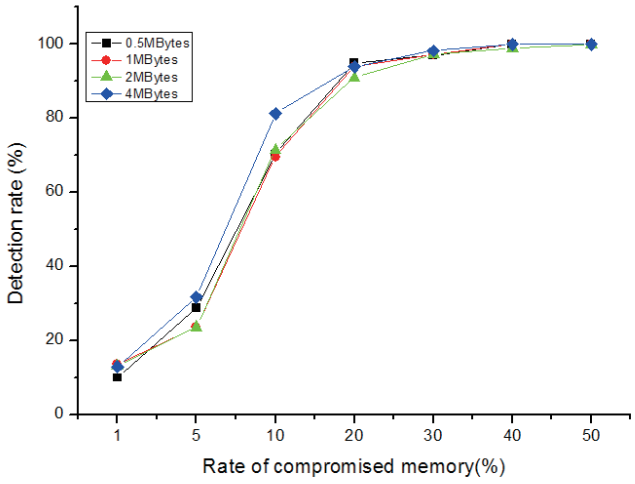

Figure 10. Compromised memory detection rate of OMAP

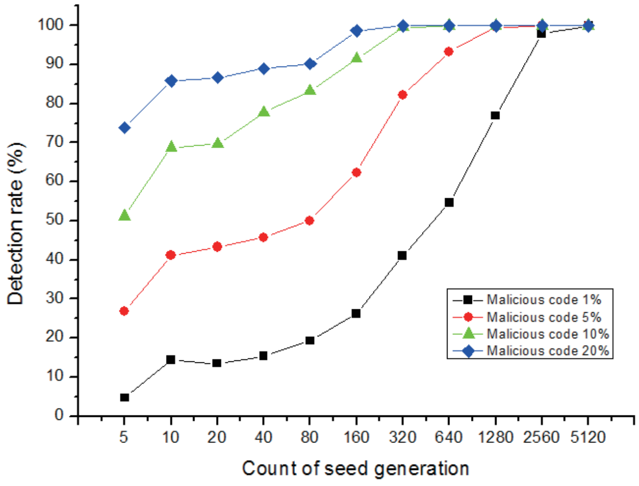

Figure 11. Impact of the count of the seed generation on the detection ratio

smart grids since our experiments only focus on memory attestation.

2) Implementations: The purposes of our experiments are two things: One is to show that OMAP can detect modification of the smart meter memory. The other is to show that OMAP does not have vulnerability against MITM attacks described in Section VI. In normal communication, two components have their benign memory set. A verifier examines the checksum from the smart meter if an attacker tampers with the smart meter's memory. We assume that the memory of the smart meter denotes 0.5 Mbytes and 1 Mbytes following the specification of GE [15]. For attestation, we utilize the RC4 as the pseudo-random number generator (PRNG) and the SHA-1 as the hash function.

In the attack scenario, an attacker between the smart meter and a verifier falsify the memory of smart meter as assuming that they perform the rainbow attack and the attestation interference attack. We vary the size of compromised memory from 1% to 50% as the result of attacks.

B. Experiment Results

1) The detection of the memory modification: In Section IV, we mentioned that if an attacker injects malicious codes or modify a firmware, then contents of memory are changed. To evaluate performance for detection of the memory modification, we measure the detection rate according to the memory size and the amount of compromised memory. We vary the rate of compromised memory from 1% to 50%, and fix the offset, the count of seed generation to 16 bits and 20 times respectively. Fig. 10 describes the rate of compromised memory versus the detection rate.

The size of contents for the attestation in 1 MBytes memory is only 40 Bytes. It is only 0.004% of the memory. In spite of small bytes, it can detect modification when 20% memory is compromised with 95% probability. Although the size of the smart meter's memory grow, the detection rate is not changed because the rate of compromised memory increases with the smart meter's memory. It means that OMAP can ensure the same performance regardless of the size of the smart meter memories.

2) The prevention of MITM attacks: we aforementioned theoretically that MITM attacks are achieved in the system applying the challenge-response protocol, not in OMAP. In experiments, by eavesdropping challenge messages, an attacker can make a rainbow table. Furthermore, we confirm that the attestation interference attack can be achieved when an attacker forwards other challenge messages. However, MITM attacks cannot be achieved since the smart meter in the system applying OMAP does not receive challenge messages.

C. Impact of the offset and the count of seed generation

Through the experiments, we found that the count of seed generation is more important than an offset for increase in the detection rate. The offset means the length of the accessed data bits. The count of seed generation denotes memory iterations. Fig. 11 and Fig. 12 show the impact of an offset and the count of a seed generation.

In Fig. 11, we show that the detection rate increases with the count of seed generation. In Fig. 12, we fix the count as 20 times in the experiment, and the offset is fixed as 16 bits. The percentage of compromised memories set 10% and the sizes of memory set 0.5, 1, 2 and 4 MBytes respectively. The detection rate is not changed regardless of the offset.

D. Consideration of the checksum computation time

Let t_2 be time that a verifier receives a checksum from the smart meter. t_2-t_1 is time to compute a checksum including

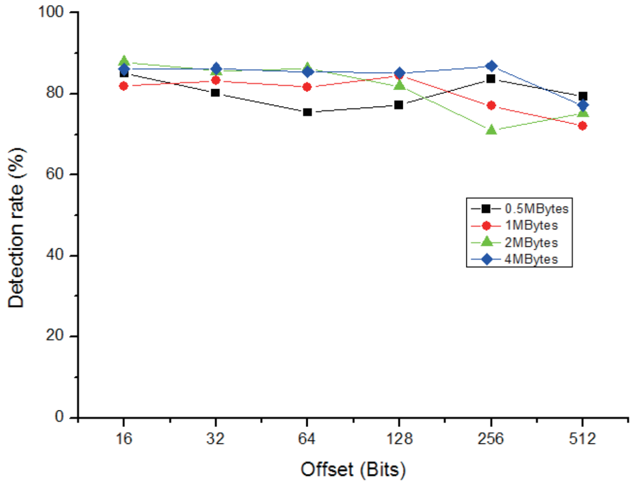

Figure 12. Impact of the offset on the detection rate

transmission time.

$$\text{Computation time} = t_2 - t_1$$

Increase in computation time can be used to verify if the smart meter is modified. We adopt this time to attest the smart meter memory. That is, if the smart meter does not send a checksum within given time, we regard that the smart meter is malfunctioning or is attacked. However, it difficult to adopt this value in practice since computation time depends on the network latency.

VIII. Conclusion

We propose OMAP, the robust remote attestation protocol against network attacks for a smart meter. The challenge-response protocol is not suitable for the smart meter because of network attacks. In this paper, OMAP prevents network attacks such as MITM attacks by sending the checksum in one-way. In addition, OMAP attests effectively a smart meter using the random memory traversal. Detection rate for modification of the memory is approximately 95% probability with only 0.004% verification of the memory.

Acknowledgment

This research was supported by the National IT Industry Promotion Agency (NIPA) under the program of Software Engineering Technologies Development, and Business for Academic-industrial Cooperative Establishments funded Korea Small and Medium Business Administration in 2011 (R0907812). Additionally, this research was supported by a grant NGIT2009100109 from the Northrop Grumman Information Technology Inc Cybersecurity Consortium.

References

[1] P. McDaniel and S. McLaughlin, "Security and privacy challenges in the smart grid," *IEEE Security & Privacy*, vol. 7, no. 3, pp. 75–77, 2009.

[2] The Smart Grid Interoperability Panel Cyber Security Working Group, "Guidelines for smart grid cyber security," Tech. Rep. NISTIR 7628, NIST, Aug. 2010.

[3] A.Seshadri, A.Perrig, L.van Doorn, and P.Khosla, "SWATT: SoftWare-based ATTestation for Embedded Devices," in *Proceedings of the 25th IEEE Symposium on Security and Privacy*, 2004, pp. 272–282.

[4] A.Seshadri, M. Luk, A.Perrig, L.van Doorn, and P.Khosla, "Pioneer: verifying code integrity and enforcing untampered code execution on legacy systems," in *Proceedings of the 20th ACM Symposium on Operating Systems Principles (SOSP)*, 2005, pp. 1–15.

[5] A.Seshadri, A.Perrig, L.van Doorn, and P.Khosla, "SCUBA: Secure Code Update By Attestation in Sensor Networks," in *ACM Workshop on Wireless Security (WiSe 2006)*, 2006, pp. 85–94.

[6] A.Shah, A.Perrig, and B. Sinopoli, "Mechanisms to provide integrity in SCADA and PCS devices," in *International Workshop on Cyber-Physical Systems Challenges and Applications (CPA-CA)*, 2008.

[7] S.McLaughlin, D.Podkuiko, and P.McDaniel, "Energy theft in the advanced metering infrastructure," in *Critical Information Infrastructures Security*, vol. 6027 of *Lecture Notes in Computer Science*, pp. 176–187. 2010.

[8] F.M.Cleveland, "Cyber security issues for Advanced Metering Infrastructure (AMI)," in *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, pp. 1–5. 2008.

[9] D.Schellekens, B.Wyseur, and B.Preneel, "Remote attestation on legacy operating systems with trusted platform modules," *Electronic Notes in Theoretical Computer Science*, vol. 197, no. 1, pp. 59–72, 2008.

[10] H.Khurana, M.Hadley, Ning Lu, and D.A.Frincke, "Smart-grid security issues," *IEEE Security & Privacy*, vol. 8, no. 1, pp. 81–85, 2010.

[11] M. LeMay and C. A. Gunter, "Cumulative attestation kernels for embedded systems," in *ESORICS*, 2009, pp. 655–670.

[12] W.Stalling and L.Brown, *Computer Security*, Pearson Education International, 2008.

[13] A.R.Metke and R.L.Ekl, "Smart grid security technology," in *Innovative Smart Grid Technologies (ISGT)*, pp. 1–7. 2010.

[14] SmartSynch, "Smartsynch: We create smart grids," 2010, http://www.smartsynch.com.

[15] W.F.Boyer and A.M.Scott, "Study of security attributes of smart grid systems-current cyber security issues," Tech. Rep. INL/EXT-09-15500, Idaho National Laboratory, Apr. 2009.