Information Systems and Signal Processing Journal (2019) 4: 7-12
Clausius Scientific Press, Canada

DOI: 10.23977/isspj.2019.41002
ISSN 2371-9427

Dynamic Integrity Measurement Scheme of Smart Meter Based on Trusted Computing

Shaomin Zhang[a,*], Tengfei Zheng[b], Baoyi Wang[c]

Department of Control and Computer, North China Electric Power University, Huadian Road, Baoding, China
zhangshaomin@126.com,zhengayt@foxmail.com,wangbaoyi@126.com

Keywords: smart meter, integrity, trusted computing, dynamic measurement, software traces

Abstract: With the rapid development of smart grids, smart meters, as terminals of smart grids, are taking on increasingly important tasks. As the smart meter terminal's working environment is complex and diverse, and smart meters suffer from endless physical attacks and cyber attacks, so the integrity of smart meters is under threat. Aiming at addressing the problems above, a dynamic integrity measurement scheme for smart meters based on trusted computing is proposed in this paper. Based on the idea of software traces and embedding, this scheme extracts the software traces and embeds it in the file in the offline phase. The process monitor is introduced to monitor software behavior in real time, effectively solving the TOC-TOU attack problem. Practice shows that this scheme saves memory resources, has little impact on system performance and high application value.

1. Introduction

In recent years, smart grids have received great attention from governments, industry, and academia in the world due to their high efficiency and reliability. As an important part of the smart grid, smart meters are used to monitor the electricity consumption of different customers, both residential and industrial, and to feed back the billing information to the customers, providing great convenience to energy producers and consumers [1]. The bi-directional interaction between the smart meter and the power control center in real time increases the reliability, flexibility and efficiency of the whole power system.

However, network attacks against the power system are emerging, and the majority of attacks are initiated from the terminal [2]. Because of recording and transmitting sensitive information such as electricity charges, smart meters are high-value target for cyber attacks. Unauthorized physical access to smart meters can have as affect the retrieval of sensitive data and encryption keys, as well as unauthorized hardware modification by adversaries; it can also lead to electricity theft by the final customer. Also, local or remote attackers might try to modify application code running on smart meters in order to get access to sensitive data or alter consumption readings (electricity theft) [3]. Malicious code for smart meter terminals can tamper the control logic of terminals, or even can initiate backtracking attacks on the master system via terminal devices [4]. Collection terminals are facing security issues such as eavesdropping, tampering, deletion, the destruction of hardware and

software [5]. Therefore, the security and integrity of smart meters should be given sufficient attention.

2. Related work

There are no effective monitoring methods for smart meters because of the wide range of them. In order to solve the above problems, the author introduces Trusted Computing (TC) into smart meters. The idea of using trusted computing to protect terminal security in smart grids is not new [6]. Reference [7] defines smart meters with the Trusted Platform Module (TPM) as trusted smart meters. Similarly, Kuntze et al. [8] describe a system for using non-migratable TPM keys for device identification in smart grid environments. Reference [9] proposes to establish a network security immune system of power monitoring and control system based on trusted computing technology, which provides an effective and active defense mechanism for power control system. However, the above documents focus on remote proof in smart grids, and the method of dynamic metering of smart meter terminals is not given. The inappropriate dynamic measurement scheme will reduce the operation efficiency of smart meters, even affect the security of smart meter terminals, thus threatening the security of the entire smart grid. Integrity measurement, as an important operation of trust transfer and remote proof in trusted computing, deserves extensive attention of researchers.

In recent years, researchers have proposed a series of integrity measurement models. Commonly there are IMA [10], PRIMA [11], BIND [12], LKIM [13] and so on. Essentially, IMA and PRIMA are static measurement systems, because they only measure the software when the software load, and then the behavior of the software will not be regulated, which causes TOC-TOU problems. To a certain extent the BIND system realizes the dynamic measurement of the software's key codes, but requires the programmer to insert the measurement points manually. But the programming work is large, and the compatibility of this system is poor. LKIM system uses the isolation characteristic of virtual platform to measure the integrity of virtual machine by monitoring the memory of virtual machine. It defines a series of kernel system variables, which can trigger the measurement of the kernel by the change of these variables, but these state variables cannot represent the real dynamic measurement. Affected by the above challenges, a dynamic measurement scheme for smart meters based on software behavior and the idea of embed was proposed in this paper, which can detect the integrity of the smart meter terminals in real time and solve the TOC-TOU problem effectively.

3. Dynamic integrity measurement scheme design

3.1 Scheme architecture

In recent years, software behavior has become an important method to describe the dynamic credibility of software [14]. Based on software behavior and the idea of embed, a dynamic integrity measurement scheme that can embed in software traces in this paper. The structure of the scheme is shown in Fig. 1. The measurement architecture of this paper can be divided into offline test phase and operation phase. In the offline test phase, the normal running behavior traces of the software installed on the smart meter is collected in a trusted environment, and then the traces are embedded in the application. Then it initializes and generates the benchmark values and stores them in the benchmark library. In the running phase, the scheme measures the smart meter application process embedded traces dynamically according to the behavior traces and the benchmark database of measurement values.

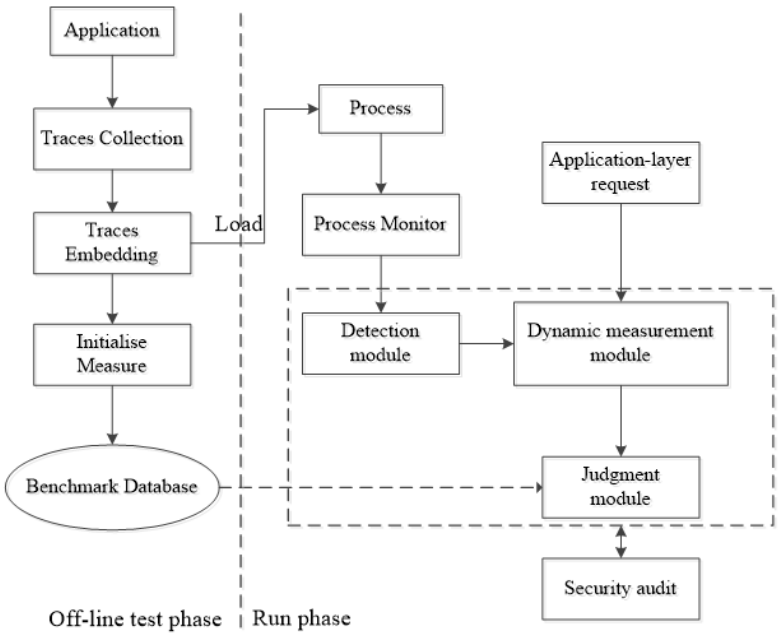

Figure 1: Dynamic measurement architecture.

3.2 Architecture definition

Definition 1 (Software Behavior): In the actual operation of software system, software as the subject, implements, operates or acts on the object relying on its own functions [14]. The subject refers to the process in the system, the object refers to the system resources that the process operates, and the action refers to an operation that the subject acts on the object. The definitions are as follows:

$$\text{Actions} = \{action = (s)\ APPLIES\ (f)\ to\ (obj)\} \qquad (1)$$

Where $s \in Subjets, f \in Functions, obj \in Objects$, $Subjets$ is the subject set, $Functions$ is the function set (process, service), and obj is the object set.

Definition 2 (Behavior Traces): The sequence of the behaviors of the subject in a certain period of time are recorded in the form of string, which becomes the behavior trace of the subject. The definitions are as follows:

$$\text{T} = \{trace(n) = A_1, A_2 \ldots A_i \ldots A_n (1 \ll i \ll n)\} \qquad (2)$$

Where A_i is the behavior of a process at a certain time. In the off-line test phase, the software behavior traces of smart meters is acquired through system call sequence when they work in trusted environment. The traces of each software function in smart meters is T_i.

Definition 3(Detection Function): The detection function f_d is mainly used to detect whether the behaviors of the process belong to T, that is $f_d: A_i \rightarrow \{true, false\}$. If and only if $A_i \in$ T , $f_d(A_i) = true$. Otherwise $f_d(A_i) = false$.

Definition 4(Verification Function): The verification function f_v is mainly used to verify whether the current metric of the process is consistent with the embedded metric.

Assuming the current metric be $newHV$, the embedded value be HV, and when $newHV = HV$,$f_v = true$, otherwise $f_v = false$.

Definition 5(Judgement Function): The judgement function f_j is mainly used to judge whether process P is trustworthy, that is, $f_j: p \rightarrow \{trust, malicious\}$. Assuming the current modification

behavior of process P is A_p, the following rules are used to judge.

$$\begin{cases} f_j(p) = trust & iff \quad (f_v = true) \wedge (f_d(A_i) = true) \\ f_j(p) = malicious & iff \quad f_v = false \\ f_j(p) = malicious & iff \quad f_d(A_i) = false \end{cases} \tag{3}$$

Definition 6(Modification Behavior): There are many processes in the running smart meter, and a process can be broken down into several process behaviors. If a process behavior in a smart meter can undermine system integrity, then we call it a modification behavior.

3.3 Behavior traces embedding

In embedded systems, the commonly used executable file format is ELF (executable and linkable format) format. The ELF file consists of four parts, an ELF header, a Program header table, a Section, and a Section header table [15]. ELF file format as shown in Fig. 2. A detailed introduction of ELF can be found in the literature [15], and will not be repeated here.

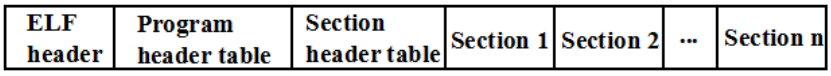

Figure 2: ELF file format.

As shown in Fig. 3, the method of embedding behavior trace into ELF file is similar to that in paper [16]. We add the behavior trace as an additional section to the end of the file to make it a legitimate part of the file. At the same time, the new section is marked as a read-only attribute to protect it from being modified to a certain extent.

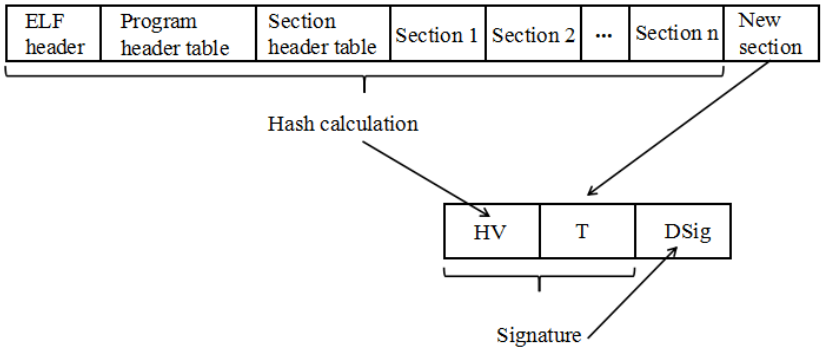

Figure 3: Trace embedding diagram

The new section content includes the following: ①The eigenvalue HV. It is a hash calculation of the part other than the new section. It will be used as the standard value when the judgement function f_j judge whether the process p is trusted. ②Behavior trace T. It will serve as the basis for the detection function f_d to detect whether the behavior of the process belongs to T. ③DSig is a signature of ①② by TPM in order to ensure the integrity of the trace and prevent malicious modification of new sections.

3.4 Dynamic measurement process

In the off-line test phase, the initial measurement of software can ensure the static initial credibility of the software. After the software load into memory, the process monitor starts to monitor the process in the system. The dynamic measurement process of this scheme is shown in

Fig.4.

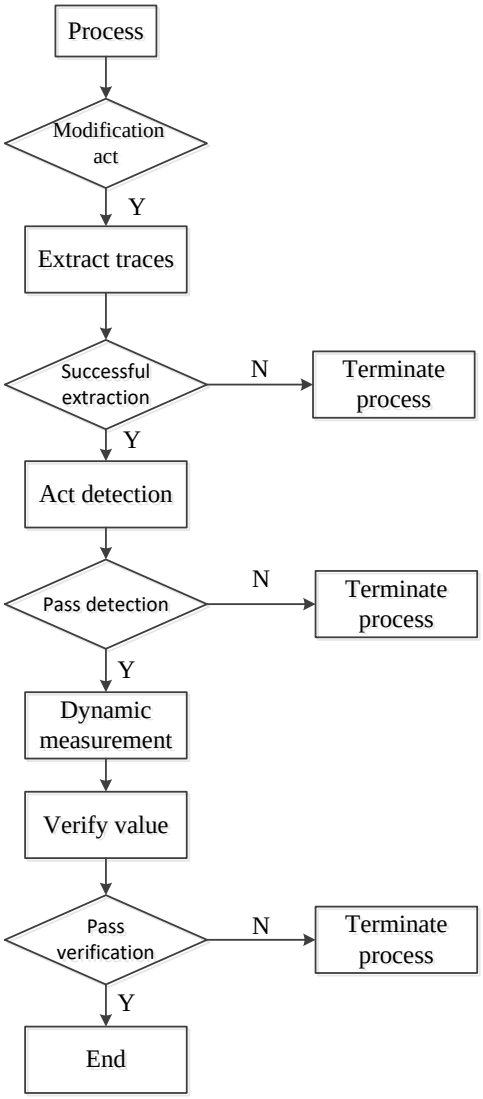

Figure 4: Dynamic Measurement Process

The dynamic measurement process is described as follows.

1) The process monitor monitors the process. If it finds that process p has modification behavior A, it sends process p to the detection module for processing.
2) The detection module first extracts the trace T embedded in p, and terminates the process if the extraction fails. If the extraction is successful, the detection function f_d is used to detect whether behavior A belongs to T, if $f_d(A) = false$, the process is terminated, and if $f_d(A) = true$, the process is sent to the dynamic measurement module for processing.
3) The dynamic measurement module measures the integrity of the process p, that is, calculate the hash value of the section other than the newly added to obtain a new feature value $newHV$. Then, the measurement result $newHV$ and the process p are sent to the judgement module for processing.
4) The judgement module first extracts the eigenvalue HV, and then uses the verification function f_v to verify whether the $newHV$ and the HV are equal. Finally, judging whether the process p is trusted according to the function f_j. If $f_j(p) = trust$, the process p is trusted, let it continue to run, and if $f_j(p) = malicious$, the process is terminated.

5) The judgement module generates a metric log and sends it to the security audit module, which is stored securely by the security audit module.

4. Conclusions

Aiming at the special use scenarios of smart meters in smart grids, a smart meter dynamic integrity measurement scheme using trusted computing technology is proposed in this paper. The scheme introduces a process monitor to monitor the behavior of the process on the smart meters in real time, implement dynamic measurement on the suspicious process, and solve the TOU-TOC problem effectively. Based on the idea of embed, the behavior trace is embedded in the software, which saves the memory resources and improves the system operation efficiency. And the cumbersome operations such as the extraction and embedding of the software behavior trace are completed in the offline test phase. How to refine the detection function and improve the efficiency of the measurement algorithm are problems worthy of further study.

Acknowledgements

The paper is supported by the Fundamental Research Funds for the Central Universities (2018ZD06).

References

[1] Brown R E. Impact of Smart Grid on distribution system design[C]// Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the, Century. IEEE, 2008:1-4.
[2] ZHANG Shaomin, WANG Zhinan ,WANG Baoyi.Terminal integrity detection scheme of electricity information acquisition system based on trust computing [J].Electric Power Automation
[3] Karopoulos G, Xenakis C, Tennina S, et al. Towards trusted metering in the smart grid[C]// IEEE, International Workshop on Computer Aided Modeling and Design of Communication Links and Networks. IEEE, 2017.
[4] Kunlun, WANG Zhihao, AN Ningyu, et al.Construction of the immune system of cyber security for electric power supervise and control system based on trusted computing[J].Advanced Engineering Sciences, 2017, 49 (2) :28-35(in Chinese).
[5] Mesbah W. Securing Smart Electricity Meters Against Customer Attacks[J]. IEEE Transactions on Smart Grid, 2016, PP (99):1-1.
[6] Kuntze N , Rudolph C , Cupelli M , et al. Trust infrastructures for future energy networks[C]// Power & Energy Society General Meeting. IEEE, 2010.
[7] Zhao J, Liu J, Qin Z, et al. Privacy Protection Scheme Based on Remote Anonymous Attestation for Trusted Smart Meters[J]. IEEE Transactions on Smart Grid, 2016, PP(99):1-1.
[8] Kuntze N , Rudolph C , Bente I , et al. Interoperable device identification in Smart-Grid environments[C]// Power & Energy Society General Meeting. IEEE, 2011.
[9] Gao Kunlun , Wang Zhihao , An Ningyu , et al. Construction of the immune system of cyber security for electric power supervise and control system based on trusted computing[J]. Advanced Engineering Sciences, 2017, 49(2): 28-35(in Chinese).
[10] Sailer R , Zhang X , Jaeger T , et al. Design and implementation of a TCG-based integrity measurement architecture[C]// Conference on Usenix Security Symposium. USENIX Association, 2004.
[11] Jaeger T , Sailer R , Shankar U . PRIMA -- Policy Reduced Integrity Measurement Architecture [J]. 2007.
[12] E , Perrig A , Doorn L V . BIND: A Fine-Grained Attestation Service for Secure Distributed [13]Systems[C]// IEEE Symposium on Security & Privacy. IEEE Computer Society, 2005.
[13] Loscocco P A , Wilson P W , Pendergrass J A , et al. Linux kernel integrity measurement using contextual inspection[C]// Acm Workshop on Scalable Trusted Computing. 2007.
[14] Qu Y w Software Behavior[M].Publishing House of the Elccn-onic Industry,2004.
[15]Tool Interface Standards (TIS) Committee. Executable and linkingformat(ELF) specification version 1. 2 [EB /OL]. [2007-01-26] .http: / /x86. ddj. com /ftp /manuals /tools /elf. Pdf
[16] DENG Y, CHEN Z. Policy embedded dynamic integrity active measurement architecture[J]. Application research of Computers, 2013,30(1) : 162 - 264.