Sustainable Secure Management Against APT Attacks for Intelligent Embedded-Enabled Smart Manufacturing

Sustainable Secure Management against APT Attacks for Intelligent Embedded-Enabled Smart Manufacturing

Jun Wu, Mianxiong Dong, Kaoru Ota, *Member, IEEE*, Jianhua Li and Wu Yang

Abstract—Intelligent embedded-enable smart manufacturing is an important infrastructure for future industries. Increasing security threats are disturbing the normal operations of smart manufacturing. As a novel type of threat, an advanced persistent threat (APT) has the novel features of strong concealment, latency and long-term entanglement, which can penetrate the core systems of smart manufacturing, especially for intelligent embedded systems, and cause great destruction from the cyber side to physical side. However, the existing security schemes cannot provide sustainable resource management, which causes the core system in smart manufacturing not to perform sustainable secure detection and defense against APTs. To address this challenge, this paper proposes a sustainable secure management mechanism for smart manufacturing against APTs. The proposed mechanism includes two parts: sustainable threat intelligence analysis and sustainable secure resource management. Sustainable threat intelligence analysis provides sustainable discovery of the indications of potential APTs , which has features of a weak signal, low correlation and slow time variation. The sustainable secure resource management provides deep and continuous protection for intelligent embedded systems in smart manufacturing. The evaluations show the defense capabilities and the feasibility of the proposed mechanism.

Index Terms—Smart manufacturing, sustainable security, resource management, advanced persistent threat (APT).

1 Introduction

Smart manufacturing is currently an important developmental trend for future industries, especially future manufacturing [1][2]. In smart manufacturing, industrial control and automation systems play important roles and improve the intelligence and efficiency of the power control, transportation, water supply, etc. Many advanced computing and networking technologies can be introduced into smart manufacturing to improve the efficiency and performance, such as software-defined networking (SDN), information-centric networking (ICN), and fog computing [3][4][5][6]. Artificial intelligence is also a promising technology to optimize the performance of smart manufacturing [7][8][9]. As a core component of smart manufacturing, the intelligent embedded system in industrial automation systems is the basic infrastructure to implement advanced computing and networking technologies. Cyber security is also a very important issue for smart manufacturing[10][11][12]. The intelligent embedded systems in smart manufacturing are often deployed in potentially adverse or even hostile environments, from which an attacker can generate all types of threats and attacks. These threats can cause great impacts and disrupt the normal running of resources of the embedded systems in smart manufacturing, such as user processes, kernels, industrial I/O, etc. Furthermore, an advanced persistent threat (APT) [13][14], which is a novel kind of attack, can target smart manufacturing and cause great destruction. Different from traditional threats, an APT is a set of stealthy and continuous cyber hacking processes, often orchestrated by a single attacker or a specific entity. An APT usually targets either private organizations, states or both for business or political motives. APT processes require a high degree of covertness over a long period of time. APT defense is an unresolved problem in smart manufacturing.

Figure 1 provides the conceptual architecture and work principal of smart manufacturing. As shown in Fig. 1, the service components of smart manufacturing include service networks, smart logistics, smart products, and the sensing and control infrastructure. All the data and control flows in smart manufacturing are processed by the functional components of manufacturing data and intelligence processing, intelligent embedded systems, and smart manufacturing applications. The data flows are the status date from the service-layer, such as the data from the logistics and supply chain, which can provide the input for smart analysis and decisions. Moreover, the control flows are the commands from the managers, which can insert human-decision into the smart manufacturing. Manufacturing data and intelligence processing can enable intelligent decisions in complex environments based on smart logistics and products, which can enhance the efficiency and capacity from products to markets. Intelligent embedded systems are the carriers of all the smart and normal processing of data and services, which is the core component of smart manufacturing. Based on intelli-

- *Jun Wu and Jianhua Li are with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: junwuhn@sjtu.edu.cn, lijh888@sjtu.edu.cn.*
- *Mianxiong Dong and Kaoru Ota are with the Department of Physics, Department of Information and Electric Engineering, Muroran Institute of Technology, Muroran 050-8585, Japan. E-mail: mx.dong @csse.muroran-it.ac.jp, ota@csse.muroran-it.ac.jp.*
- *Wu Yang is with the Information Security Research Center, Harbin Engineering University, Harbin 150001, China. E-mail: yangwu@hrbeu.edu.cn.*

gent embedded systems, smart manufacturing applications can be implemented and directly provide smart services for users. When facing an APT, the intelligent embedded system is the most vulnerable of the aforementioned components of smart manufacturing. In many existing APTs in smart industry including smart manufacturing, such as Stuxnet [15], the intelligent embedded system is the main target. By interfering with the intelligent embedded system, an attack chain can be generated. Therefore, it is very important to provide sustainable secure protection for intelligent systems again an APT in smart manufacturing.

In the current smart industry, especially smart manufacturing, Linux-based open-source operating systems,

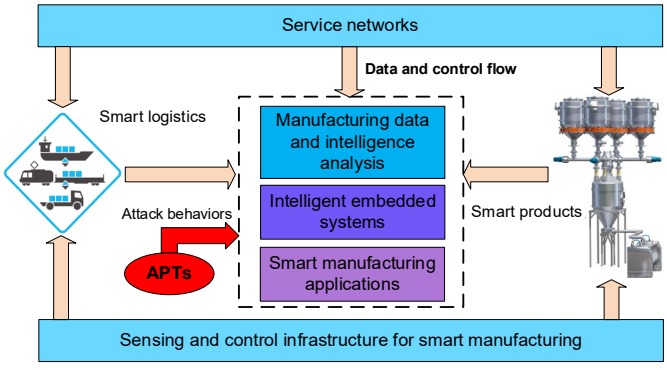


Fig. 1. Smart manufacturing architecture and APT.

especially Real-Time Application Interface (RTAI)-Linux, have been widely applied in manufacturing industry systems [17]. Currently, increasing threats are capable of disturbing the normal function of an industrial automation system because many sophisticated attacks can masquerade as legitimate applications or services to destroy the resources of RTAI-Linux, such as user processes, the kernel, and industrial I/O. There have been some works that focused on the security of embedded systems[18], [19] [20], [21], [22], [23]. However, these works just studied single points of the security in embedded systems, including the security of the code, memory, processor, etc. These existing schemes cannot provide sustainable resource management against APTs, which means that the software and hardware of the smart devices in industry cannot perform sustainable security detection and defense against sophisticated threats. Therefore, it is still an open issue to propose a global security framework for the resource management of intelligent embedded systems.

To address the above challenges, this paper proposes a sustainable secure management mechanism for smart manufacturing. There are two main contributions in this paper. First, this paper introduce threat intelligent analysis into smart manufacturing to detection APTs with strong concealment, latency and long-term entanglement. Second, a secure management approach is proposed to implement the sustainable secure resource management and provide deep and continuous protection. The rest of the paper is organized as follows: In Sect. 2, the preliminaries and basic idea are presented, and related works are analyzed in Sect. 3. The details of the sustainable threat intelligence analysis and secure resource management are given in Sect. 4 and Sect. 5, respectively. Sect. 6 presents evaluations. Finally, Sect. 7 concludes this paper.

2 Preliminaries and Basic Idea

2.1 Advanced Persistent Threat (APT)

An APT is a kind of novel and sophisticated attack that is often started by directly targeting a few power users with malicious software [13]. The APT then propagates itself by exploiting flaws in applications deployed across the systems of key infrastructures including smart manufacturing, the power grid, or the water supply. When facing APTs, all networks are untrusted and the traditional secure defense components are no longer in force because of the strong concealment, latency, and long-term entanglement. Through penetrating the cyber systems of the key infrastructure, an APT can destroy a physical system. Some existing APTs, such as Stuxnet[15] and Duqu[16], have already caused great destruction in smart energy, manufacturing, and water systems. The embedded system is the most vulnerable attack point for the APT. Currently, the development of a systemic secure framework against APTs is still an open issue.

2.2 Intelligent Embedded Systems

Real-time application interface (RTAI) provides a real-time approach based on Linux [24]. RTAI defines a real-time hardware abstraction layer (RTHAL) on Linux that provides a set of program interfaces for the necessary modifications on Linux. Therefore, RTAI just needs to communicate with the program interfaces. RTAI can reduce the modifications to the codes in the Linux kernel.

The basic architecture of the RTAI-Linux-based intelligent embedded system is shown in Fig. 2. In this paper, we take RTAI-Linux as a typical example of an intelligent embedded system for study.

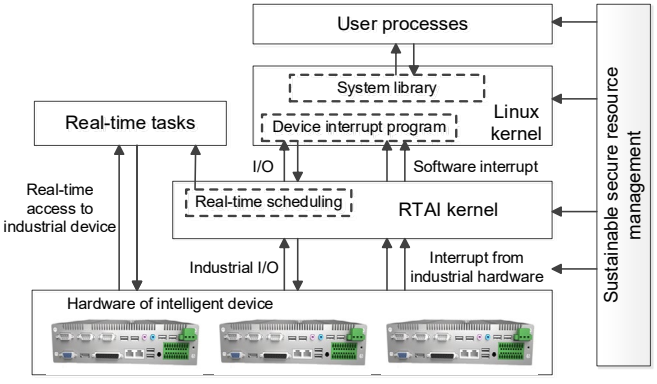


Fig. 2. RTAI-Linux-based intelligent architecture and the security requirements.

2.3 Requirements of Secure Resource Management

In smart manufacturing, intelligent embedded systems can control the resources of an industrial automation system, including the user process module, application module, control progress, communication module, clock module, etc. Based on cooperation among the resources in the industrial embedded systems, a smart control capability can be provided for smart manufacturing. However,

the sustainable secure management of the resources in intelligent embedded systems is lacking in current smart manufacturing, which will cause vulnerabilities into the intelligent embedded systems in smart manufacturing. Because intelligent the embedded system is the most critical infrastructure of smart manufacturing, the normal work flow can be disturbed once security threats occur. First, to defend against APTs, an efficient and sustainable threat intelligence analysis approach is necessary to detect the potential threat clues. Second, the software and hardware of intelligent embedded systems must form a sustainable and cooperative defense architecture. When the threat propagates from one module to another, the secure management mechanism should stop the propagation and control the destruction. These requirements are still open issues in smart manufacturing.

2.4 Basic Idea

To improve the secure defense capability against APTs in smart manufacturing, this paper proposes a sustainable secure management mechanism. The proposed sustainable secure management mechanism includes two parts: sustainable threat intelligence analysis and sustainable secure resource management. APT intelligence analysis provides the sustainable discovery of the potential APT clues, which have the features of a weak signal, low correlation and slow time variation. Moreover, the sustainable secure resource management provides deep and continuous protection for the software and hardware of the intelligent embedded systems in smart manufacturing. The APT intelligence analysis and sustainable secure management form a seamless loop, which is shown in Fig. 3.

3 Related Works

3.1 Embedded Systems in Smart Applications

Real-time Linux-based embedded systems for smart industry including smart manufacturing have attracted much attention.

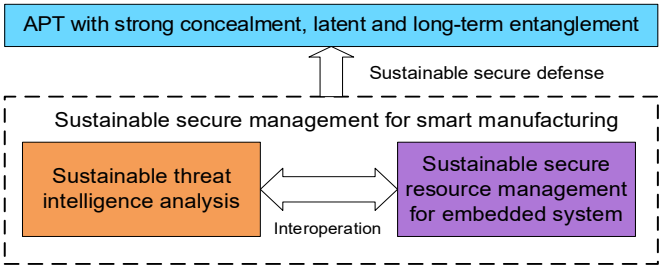


Fig. 3. Basic idea of sustainable secure management.

The application of Ethernet Powerlink for communication in a computerized numerical control (CNC) system using Linux was presented by Krystian Erwinski et al. [24]. In this work, the necessary modifications to the Linux configuration, Ethernet Powerlink (EPL) stack, computer basic I/O system, etc. were studied in depth. As the next revolution in embedded-system development, a model-driven engineering (MDE)-based framework was used by George Doukas et al. as a control and automation system using real-time Linux [25]. A. Barbalace considered the Linux IP stack and RTnet, an open-source hard real-time network protocol for Xenomai and RTAI, which verified that real-time Linux is feasible to replace VxWorks [26]. Keith G. Erickson analyzed the National Spherical Torus Experiment Upgrade (NSTX-U) advances in real-time C++11 on Linux [27]. NSTX-U was proposed by the Princeton Plasma Physics Laboratory to satisfy the needs of fusion devices in industrial automation systems. To realize predictable industrial embedded systems, Mikael Asberg et al. proposed a fast Linux bootup based nonintrusive scheme [28]. In this work, the Linux kernel is the unmodified Linux kernel, and predictability and reliability are achieved by a nonpatched real-time scheduler module. Praween Amontamavut et al. proposed a separate Linux process logging model for embedded systems [29] that can be used to log the behavior of Linux processes in Android. To capture the traffic from a large number of CAN buses in industrial automation systems, Michal Sojka et al. studied in depth the performance evaluation of Linux CAN-related system calls [30]. For cooperative retransmission, Vitalik Nikolyenko et al. focused on the implementation and evaluation of a SoftMAC-based Linux kernel module [31], which reduced the overhead due to block acknowledgment. This work shows the possibility of cooperative retransmission achieving even greater performance improvements at 802.11n rates. In fact, 802.11n is often used to establish the wireless networks in smart cities, so this work is significant for wireless industrial automation systems in smart cities.

3.2 Security Technologies for Embedded Systems

To enhance the security level of the intelligent embedded systems, a number of schemes have been proposed to defend against traditional security threats.

To secure the Linux containers and their workloads, the automatic construction of rules describing the expected activities of containers is used in [18]. To establish a secure and reliable embedded system based on open source software, a security assurance development process was proposed for building a reliable Linux-based operating system [19]. In a previous work, a layered and componentized security architecture for Linux was proposed that divides the security module into individual components, and thus high cohesion and low coupling can be achieved. However, the existing security hardening mechanisms protect specific applications and are not designed to protect entire environments such as those inside the containers. A new lightweight compact encryption system was proposed based on bit permutation instruction group operation (GRP) that satisfies the resource-strained environment in the present industry [20]. Moreover, there have been some existing works focusing on coding security. For example, a security coding scheme was proposed for an embedded system that combines rate splitting, superposition coding, nested binning, and channel prefixing. This coding scheme can produce a secrecy capacity region for the channel in several scenarios. The works in [22] and [23] focused on the virus detection and hardware security problems of embedded systems, respectively. In addition, there have been works

that focus on the security of the control networks of various industrial systems [32], [33], [34]. Although these approaches can enhance the security of embedded systems, most of these works only defend against traditional threats but do not provide sustainable protection. A sustainable secure defense approach for secure resource management is still an unresolved problem in smart manufacturing.

3.3 Threat Intelligence Analysis

Threat intelligence analysis is a novel secure technology that can provide predictions and alarms for threats. Threat intelligence analysis has also attracted much attention.

A threat intelligence scheme was proposed for safeguarding Industry 4.0 systems [35]. The proposed threat intelligence approach is based on beta mixture-hidden Markov models (MHMMs) for detecting anomalous activities against both the cyber and physical sides of Industry 4.0. A smart management module and a threat intelligence module are included in the proposed intelligence scheme. Moreover, an intelligence-driven security-aware defense mechanism was proposed [36]. In this mechanism, a risk admission control policy was first designed to accommodate the risk tolerance and response capacity of the hosts. Then, the defense control policy was designed on two-stage decisions, involving proportional fair resource allocation and host-attack assignment, which can address the multiple attacker resources. Additionally, a distributed auction-based assignment algorithm was designed to capture the uncertainty in the number of resolved attacks. In addition, an intelligence analysis approach was proposed to detect malicious analysts who attempt to manipulate decision makers' perceptions through their intelligence reports [37]. To detect abnormal behaviors, a psychological indicator with user-modeling techniques is used in this scheme. Some important factors, such as Time Series and contents can also be used as threat intelligence elements [38][39]. Although there are some existing threat intelligence analysis schemes, it is still an open issue to propose a specific intelligence analysis model to address an APT with strong concealment, latency and long-term entanglement.

4 Sustainable Threat Intelligence Analysis

4.1 Threat Intelligence Driven APT Discovery

Intelligence analysis with respect to potential threats to the intelligent grid is the basis for an efficient defense. The reason that an APT is difficult to discover is because the APT is very insidious. The framework of the APT intelligence analysis is shown in Fig. 4.

As shown in Fig. 4, the threats related to the APT shares the characteristics of a weak signal, low correlation and low frequency, so that the regular methods of detecting the threats do not work at all. This topic uses the concept of threat intelligence to perform a threat correlation analysis in an industrial communicating network. In particular, it focuses on weak signals with a weak correlation and a low frequency.

4.2 Weak Signal Threat Intelligence Integration and Restoration

Weak signals are predictive, vague and fragmentary symbols that have various forms and origins. In threat intelligence, a weak signal has such characteristics as an abundant form, low intelligence integrity, low information relevance and practicability, and low value of independence. However, a weak signal also contains a great deal of

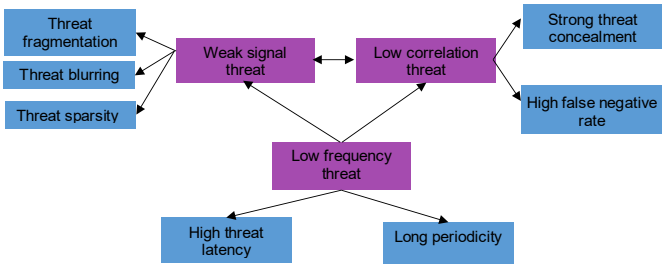


Fig. 4. Framework of sustainable threat intelligence analysis.

useful content for an APT analysis that have generally been ignored in previous defense methods. In this paper, the process of capturing weak signals is based on an analysis of the industrial communication network environment. The weakly concealed threat intelligence is clustered according to the business rules of industrial control communication, and those signals are used as inputs for multiscale symbol sequence entropy calculation, fragment correlation analysis and scenario analysis. Then, based on the industrial communication theme jigsaw puzzle and dynamic time-varying processing, we could achieve a restoration from weak signal information fragments to the full APT threat.

4.3 Low-Correlation Threat Intelligence Analysis

One of the characteristics of the smart grid APT is that the elements of its threat situation are of low relevance to the network elements, and a set of methods is needed to analyze the low correlation of the potential threat elements. This approach requires identifying and removing extraneous or redundant features from a set of feature attributes, and correlation-based feature selection (CFS) can be used to implement this process. For an APT, it cannot be completely determined that a certain feature variable is irrelevant by analyzing the known sample data, so it still needs to be considered for the unselected low-relevance threat information. As shown in Fig. 5, based on the decision tree method, this topic chooses the information gain ratio of each eigenvalue when selecting the threat intelligence features and weights the information gain ratio of the eigenvalues in the set of optimal eigenvalues w.

The correlation coefficient between the two threat intelligence attributes x and y is calculated as shown in formula (1).

$$Col(x_m, y_n) = \frac{\sum_{k=1}^{n} IDF(b_1, m) \cdot TF \times IDF(b_1, n) \cdot TF}{\sqrt{\sum_{k=1}^{n} \left(IDF(b_1, m) \cdot TF\right)^2} \times \sqrt{\sum_{k=1}^{n} \left(IDF(b_1, n) \cdot TF\right)^2}} \quad (1)$$

where TF represents the frequency of the threat intelligence attribute, IDF represents the frequency of the re-

verse attribute of one attribute value in the threat intelligence, and b represents the consecutive attribute values of the threat intelligence. The attribute with the smallest mean correlation is selected to obtain the set of low-correlation features. Then, using opportunity discovery theory, we apply a unified map of the attributes of the threat intelligence features mapped to the weights. Thus, we could represent a low-level threat as a new node of the attack map.

4.4 Analysis of Slow Time-Varying Threat Intelligence

In smart manufacturing, the randomness and dynamism of the slow time-varying parameters associated with APT and the limitations of obtaining information from slowly time-varying objects are among the major causes of the difficulty of discovering APTs. This section will study this issue. Sparse approximation is a kind of analysis method

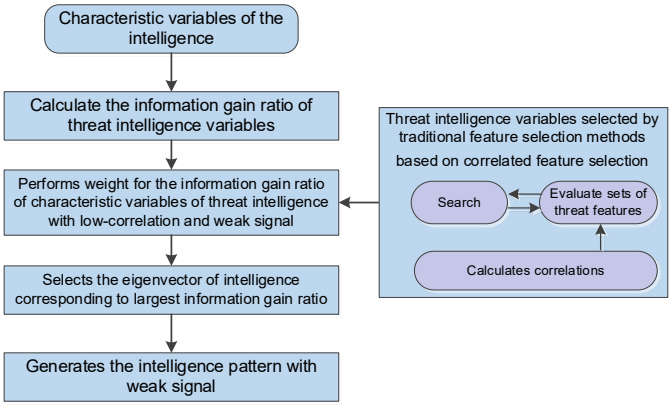


Fig. 5. Analysis of low-correlation threat intelligence correlation

applied to long-term observation data for event detection. In this paper, dynamic principal component analysis (PCA) is used to realize the sparse approximation analysis of slow time-varying threat information. In the time-domain data of the threat intelligence, additional time-varying parameters should also be introduced into the intelligence analysis. It is assumed that the one-dimensional time series data from the threat intelligence in (k) is denoted as {in (1), in (2),..., in (N)} and N is the observation point. The data matrix constructed for further principal component analysis is:

$$IN = \left[in(k-t+1), in(k-t+2), \cdots, in(k)\right] \tag{2}$$

$$O^{in} = \begin{pmatrix} in(1) & in(2) & \cdots & in(t) \\ in(1+t) & in(2+t) & \cdots & in(2t) \\ \vdots & \vdots & \cdots & \vdots \\ in(qt-t+1) & in(qt-t+2) & \cdots & in(qt) \end{pmatrix} \tag{3}$$

In Eq. (2), t is the time interval, which is set according to the actual situation in the application. After obtaining Eq. (2), the principal component analysis of the covariance matrix of IN is used to calculate the eigenvalues to evaluate the dynamics of the random process $in(k)$. On this basis, adding a nonoverlapping moving time window can improve its analytical performance. In this way, the threat intelligence data O^{in} observed in $in(k)$ can be calculated according to Eq. (3). Here, $N=qt$, and q is the number of transition time windows of $in(k)$, each with a length of t. Then, we obtain the result of the sparse approximation analysis of slow time-varying threat information data through a combination of a few pivot elements.

In addition, we use a partially observable Markov decision process to model the slow time-variable threat data, and the MDP solution provides a value or strategy for each state. The condition for such a solution is to keep an informed and complete view of the state. The partially observable Markov decision process (POMDP) just blurs the current state, which makes the existence of the uncertainty of the current status dependent on the MDP choose action; the current state is thus not always effective. Although the POMDP potential dynamic features still have Markov properties, since they are unable to obtain the current state directly, when they make decisions, the whole process of the history is required, so that they are considered non-Markov processes. At a certain point in time, there is an understanding of the beginning and the record of all actions and the values of all observations. Fortunately, when a probability distribution of all states is simply maintained, it is as if the entire history of the model is controlled. In the MDP, we track the current state and update the status after each action. However, this approach can only be an assumption, or an attempt, because it depends on the completely observable property. In the POMDP, due to the partial observability, a probability distribution of the state has to be maintained. When the model is executed and the action is observed, the probability distribution has to be updated again.

The POMDP can be solved by value iteration such as the MDP. However, as mentioned above, the POMDP strategy is the mapping between the reliability state and the action, unlike the mapping of the state and action in the MDP. Therefore, the optimal function of the MDP becomes the following:

$$V^*(b)_{a\in A} = \max\{\rho(a,b) + \gamma \sum_{b'\in B} \tau(b',a,b) V^*(b')\} \tag{4}$$

B is the set of all possible reliability states. In the previous formula, the previously defined conversion function $T(s',a,s)$ and the income function $R(s',a,s)$ are replaced with $\tau(b',a,b)$ and $\rho(a,b)$. This phenomenon is because in the POMDP, the state and decision makers cannot be known completely, so the conversion and reward functions will be defined on the basis of the state of the reliability b, rather similar to the MDP, which is based on a single state transformation function and reward function.

5 Sustainable Resource Management Mechanism

5.1 Overall Design

To realize the secure resource management for RTAI-Linux in industrial automation, a security domain based model is used as the authority to provide access control. Specifically, domain-type enforcement (DTE) access control is applied in the proposed secure resource management mechanism. DTE takes the embedded system as a collection set of subjects and objects. Here, the subjects correspond to the attribute domains, and the objects correspond to the attribute types. An operating authority table is established between the subject domain and the object types, based on which the access to the practical system is judged. The security policy contains a DTE general access control policy and white list. There are many implementations for the access control system, such as the LSM-based general access control framework and the system call reloaded-based technology. However, the source codes of the system call need to be modified to realize the reloading, and the system call is one of the most important application programming interface (API) groups. This API group has poor portability and security, so the LSM general access control framework is adopted to establish the access control system. Moreover, the LSM has plenty of hook functions, which can provide most authority and access control requirements, such as the passing of process authority, file access, and socket access. In the system design, the LSM-based general access control framework is used. Within the following specific design, the implementation of each subrequirement's access control will be described in detail. The details that are especially difficult to be address in the LSM are emphasized . Otherwise, to make the whole architecture of the access control system clearer, the Flask security architecture is referred to support more policies. The system design of the authority and access is shown in Fig. 6, which refer the Flask security architecture and use the Linux Security Module (LSM) general access control framework. Corresponding to the Flask security architecture, the LSM hooks are equivalent to the object manager, which is shown in Fig. 6. Corresponding to the LSM architecture, the security server in Fig. 6 is equivalent to the LSM's policy engine. To support the policy configuration and polling, an interactive program running in the user space is needed, which can be implemented as a user interface. The security administrator can configure the security policy with the program or poll the existing policy. The policy can be stored in two ways; in the user space, the policy is stored as a configuration file that is stored on the disk. To support policy polling by the kernel, the accessible policy database needs to be established in the kernel, which will be loaded in the memory. With a reasonable data structure design, it can be accessed conveniently and efficiently. By referring to the Flask security architecture, an authorization and access control architecture is proposed to support multiple security policies. In the proposed mechanism, there are two main kinds of security policy, a DTE security policy and the white list mechanism.

The entire system in the thick dashed box shown in Fig. 6 is proposed to be implemented, and the three parts circled with a fine dotted line are the major parts, while the other parts that are not circled with a dotted line are the processes or interfaces in the operating system. Among these three main parts, the policy configuration and polling module provide the security administrator with the capabilities of policy management and polling. The security server module parses the security policy and

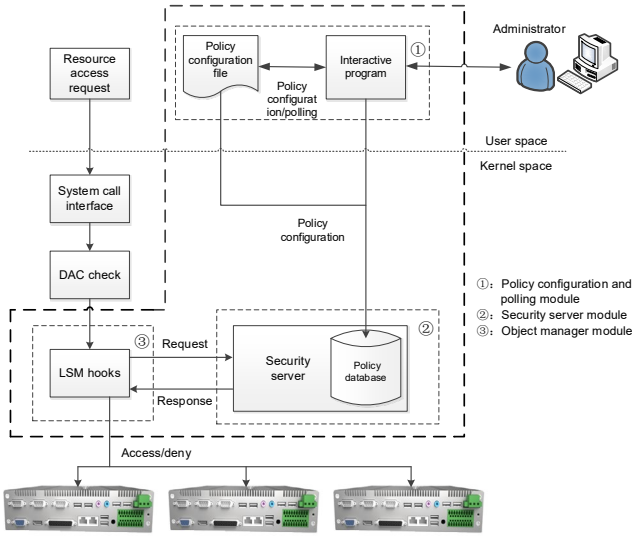


Fig. 6. Overall design of secure resource management.

the policy query requests of the corresponding hook function. Moreover, the object manager intercepts resource access requests and puts forward policy query requests to the security server. The system will be implemented as an independent kernel module with the Linux operating system, which is convenient for the upgrade and portability of the Linux system.

In RTAI-Linux, LSM is an important security module. Usually, LSM is loaded as the dynamic kernel module, and can only be loaded at the time of the system boot. Moreover, LSM just supports hooking the registrations and the loading of functions for one-time use, and the kernel is not allowed to dynamically uninstall the LSM hook. Because the Linux kernel itself has many limitations, and the update of the hook function needs to modify the source code and recompile, it is impossible to replace or use multiple load hook functions dynamically.

The design of the policy configuration and polling module, security server module and object manager module (LSM hooks) are described in the following. In Fig. 6, the thick dashed box denotes the entire system to be implemented. In the architecture, all the three parts circled with the fine dotted line are the three major components, and others parts without the dotted line are the processes or interfaces in the operating system. Among the three main parts, the policy configuration and polling module provide the security administrator with the capabilities of policy management and polling. The security server module parses the security policy and the policy

query requests of the corresponding hook functions. Moreover, the object manager intercepts resource access requests and can put forward the policy query requests to the security server. The system will be implemented as an independent kernel module with RTAI-Linux operating system, which is convenient for the upgrade and portability of the Linux system.

5.2 Policy Configuration and Polling

All operating systems' access control is based on the access control attributes associated with the subject and the object. In the traditional DAC, the access control attributes of the subject are the uid and gid of the processes, and the access control properties of the object are the uid and gid to which the object belongs. The traditional access control attributes cannot satisfy the requirements in the proposed mechanism, so a security context is introduced to present more abundant access control attributes. The security context is a variable-length string, and the security context is defined as the type.

In fact, the security context and security identifier (SID) just borrow the concept of Flask to present the security-related properties of the subject and the object. These properties are used as the flags of access control and can be defined according to the practical scenario, which Flask did not specify. In the proposed mechanism, the DTE access control model is mainly used. Thus, the type of the subject or object is taken as the security property, which is the Flask security context. The SID is the value of the security context and has the same meaning. Thus, we define the security context as type..

The policy configuration file:

In the proposed secure resource management mechanism, the subjects and objects are identified by the path and name. The industrial automation applications and services cannot be controlled when copied and run under another directory. To perform this mechanism, the administrator needs to add the new security policy to the control. Similarly, neither the software nor hardware links of the subjects and objects can be controlled without the addition of the security policy. Multiple copies and links evaluations are not supported in either the application layer or the kernel layer, which needs the administrator to manually configure the security policy.

Application policy configuration program:

The user policy configuration program provides the interface for security administrators to carry out policy polling and management. The policy configuration program runs in the user space, and its main processing flow is shown in Fig. 7.

During the policy configuration procedure, the following operations should be conducted. 1) The program receives user input information, which may include the type definition of new subject, object, mapping between subject or object and the type, and the access permissions definition for the subject to the object. This part can be implemented as a user interface, which is convenient for the security administrator; 2) After obtaining the input parameters, the user policy configuration program will map the security context, type, object class, and permissions from the variable length string to the fixed length value according the predefined rules, considering the far higher efficiency of the numerical comparison than that of a string and less space during the kernel policy polling; 3) After mapping, the user policy configuration program will write the policy into the file based on the given format; 4) Finally, the policy update module of the configuration program informs the kernel to update the policy database. The security server in the kernel incrementally updates the security rules to keep the synchronization of the policies in user space and kernel space. This operation will be implemented with the device file creation mechanism of RTAI-Linux.

5.3 Security Server

The security server is mainly composed of the kernel policy database and four subfunction modules, including the initialization submodule, the policy update monitor submodule, the policy loading submodule and the policy polling submodule. The composition and the relationships among the submodules are important issues for the proposed security mechanism, as is shown in Fig. 8. The policy configuration polling module and the object management module are also given in Fig. 8. The initialization

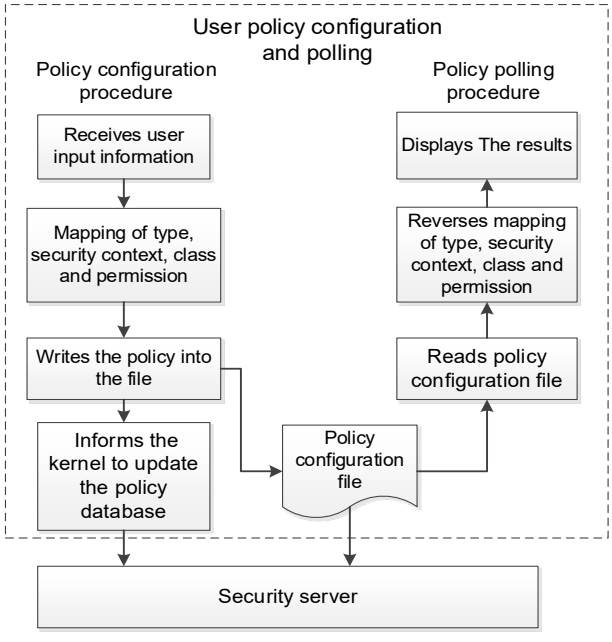


Fig. 7. Process flow of the application configuration file.

module is responsible for the initialization of the security server. To update the kernel policy space in a timely manner and provide dynamic policy configuration, the policy update monitor module monitors the notification of the policy update from the policy configuration program in the user space. Upon receiving a notification of a policy update, the policy loading module reads the policy file and loads the updated policies to the kernel policy database. Moreover, the access computing result module retrieves the policy database with the security attributes of the subject, security attributes of the object, object class and requested access methods. Additionally, the module returns the permission or rejection according to the policy database. The security server is a kernel module, and kernel trust is a prerequisite in the proposed mechanism.

Therefore, the security server is also trusted and needs no integrity protection to avoid tampering.

Two kinds of access control policies are supported in the proposed secure resource management mechanism, which are the DTE and the white list mechanism. When access requests are processed, the policy of the white list is retrieved first for the resources covered in the white list. Then, the DTE policy is retrieved. The resource access is permitted only if both security policies are passed. Otherwise, the access will be rejected as being in violation of any policy. The process flow of the security server is shown in Fig. 8.

Initialization:

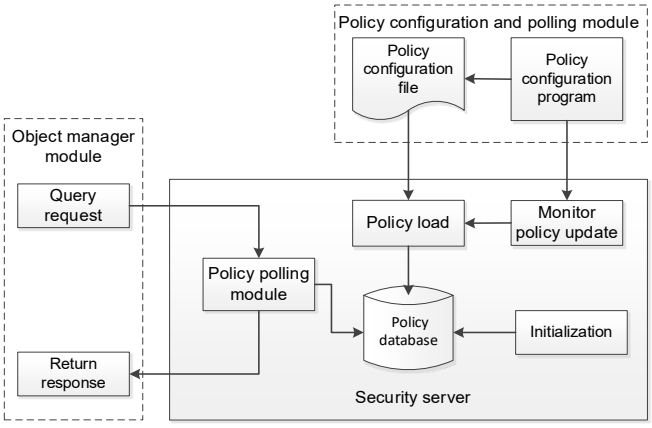


Fig. 8. The process flow of the security server.

The main function of the initialization module is to initialize the entire security server. The initialization module can realize the following operations: initializing the policy database, emptying the default initialization policies, loading the policy configuration file as the default access control policy if there is an established policy, and registering the device file needed by the incremental policy update.

Policy Incremental Update:

Policy transfer needs to perform the updates of the policy. The dynamic policy update needs a mechanism to inform the kernel to incrementally update the policy database of the security server in kernel layer when the user updates the policy through the policy configuration program. The policy update monitor module in the kernel is responsible for receiving the message of the policy update, which involves the interaction between the user space and kernel space. A virtual device is created to provide communications between the kernel module and the application of the upper layer. The virtual device driver does not care about the specific hardware device but rather aims to fulfil the data interaction between the kernel module and the application through various operating functions of the driver. A character device driver is programmed to realize various operating functions in the file operations structure, such as *Open*, *Release*, *Read*, and *Write*. Not all operating functions need to be developed in most of the device drivers. In the proposed mechanism, only the *Write* function needs to be implemented to pass the incremental policy update information generated by the application to the kernel module. According to the above information, the kernel will add, delete, or modify the corresponding policy.

Policy Loading Module:

The policy loading module converts the policy configuration file into a policy database in the kernel, which is not the security policy database of SELinux and needs to be developed. SELinux is not directly used in the proposed mechanism, but some design ideas and methods drew on the design and implementation of SELinux. Because the kernel can read the file directly, the use of the policy configuration file for policy storage greatly simplifies the process of transferring the policy to the kernel. The policy loading module mainly solves how to convert the configuration files into the defined policy database structure, that is, the process of policy parsing. The policy configuration files are read one by one, and the contents are loaded into the corresponding items of the policy database structure. Self-protection measures are taken for the policy files. In other words, as the key resources, these files are also to be protected by the white list mechanism.

Policy Polling Module:

The policy polling module receives the formatted request from the object manager. There are two main kinds of requests. One kind of request is to query the class of the subject or the object. The corresponding table includes the subject-domain and object-class mapping table. The input is the name of the subject or the object, and the output is the class of the subject or the object. Another kind of request is to query whether a subject is permitted to access a particular object. The corresponding table includes the white list and the access control table. Moreover, the input is the security attributes of the subject (subject name as white list, subject type as access control table), the security attributes of the object (object name as white list, object class as access control list), object class, and access mode (the permissions of the access). The output is then allowed or denied access.

To improve the query efficiency, the Access Vector Cache (AVC) submodule is added in the policy polling module. This submodule can cache some decision results, which can effectively improve the query speed for a request at a particular time.

The Policy Database in the Kernel:

In a general linear table or tree, the relative position of a record in the structure is random and has no determined relationship with its keyword. Therefore, the record lookup in structure needs a series of comparisons with the keyword. This kind of lookup is based on the comparison, and the lookup efficiency depends on the lookup time. The ideal case of this work is to find the record directly. Therefore, a fixed relationship between the stored position of the record and its keyword must be established, which makes each keyword correspond to a unique memory position in the structure.

As shown in Fig. 9, a hash table provides a different storage and lookup method from the general linear table, chain and search trees. The hash key value is directly mapped to a position in the table, where its correspond-

ing data elements are stored. When a lookup operation is performed, the address where the data elements are stored can be directly found according to the searched key value, and thus the data elements can be obtained. Different keywords may have the same hash address, that is, *Key*1≠*Key*2, and *f*(*Key*1) = *f*(*Key*2), and this is called a conflict.

5.4 Object Manager Module

The development of Object Manager/LSM hooks is to program corresponding hook functions for the hook points to be evaluated. From the perspective of function, LSM hooks can be classified into file hooks, socket hooks, program hooks, and some other hooks. The program hook is mainly used to control the multiple nested access of the module. The file hook mainly controls file-related access, and the socket hook mainly controls network access. These three types of hooks are mostly used in the proposed mechanism. Because there are two kinds of policies, and the white list mechanism has a higher priority,

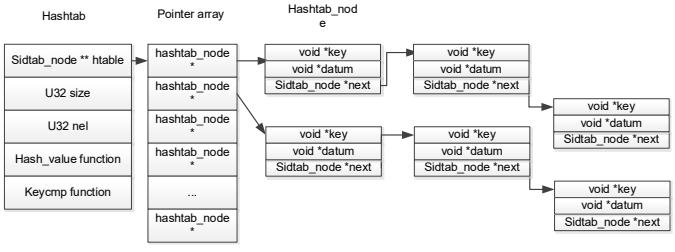


Fig. 9. The structures of the hash table.

the object manager first issues a white list query request to the security server, in which the subject name is the subject security attribute, the object name is the object security attribute, and the object class and access mode will be determined in the specific access. If the requested object is in the white list, that is, the access request is controlled by the white list, a permission decision of the access should be made according to the corresponding control policy.

5.5 Task Scheduling Algorithm

Assume that a set of independent tasks is denoted as $S=\{s_1, s_2, \ldots, s_n\}$. To denote the secure management task $s_i \in S$, multielement tuples are used, which are ($T(s_i)$, $L(s_i)$, $Y(s_i)$, $M(s_i)$, $B(s_i)$, $G(s_i)$). Here, $T(s_i)$ denotes the maximum running time, $L(s_i)$ denotes the deadline of *Task* s_i, $Y(s_i)$ denotes the response time of *Task* s_i, $M(s_i)$ denotes the remaining time of s_i, $B(s_i)$ denotes the elapsed time of s_i, and $G(s_i)$ denotes the time of s_i occupied by a task with high priority. In addition, $CurSt(s_i)$ denotes the current state of the task, including the running state, waiting state, blocked state, discarded state, finished state, etc.

Definition 1: For real-time task s_i, if the security management policy $C(p_i)=\{p_{i,j}^1, p_{i,j}^2, \cdots, p_{i,j}^n\}$ is used, the security cost *SecC* and security quality *SecQ* can be computed as follows.

$$SecC_i = \sum_{k=1}^{n}\left(N^k(s_i)\times H(p_{i,j}^k)\right) \tag{5}$$

$$SecQ_{i,j} = \sum_{k=1}^{n}\left(QoS(p_{i,j}^k)\times e_k\right) \tag{6}$$

where $N^k(p_i)$ denotes the number of *Task* p_i calls to the security service. $H(p_{i,j}^k)$ denotes the time cost to realize service $p_{i,j}^k$, and $QoS(p_{i,j}^k)$ denotes the security quality. e_k denotes the weight of the security management service in the whole security system. If the value of e_k increases, the impact of the current security service also increases.

Definition 2: A schedulable task set is the real-time task set in the scheduled queue, which is denoted as S_{sched}.

Definition 3: For the security management task , if the task is added into task set S_{sched}, the maximum response time can be calculated based on Eq. (4).

$$Y(s_i) = B(s_i) + M(s_i) + G(s_i) \tag{7}$$

$$M(s_i) = T(s_i) + SecC_i - B(s_i) \tag{8}$$

TABLE 1
SCHEDULING ALGORITHM

Performs updates for the scheduling data in $S_{sched.}$;
Switch case $s_i = S_{Key} \cup S_{General}$;
if s_i is schedulable in S_{sched} then
$CurSt(s_i)$=*Waited State;*
else if $S_{dis-schedulable}(s_i) \neq \varnothing$
rejects each $s_i = S_{dis-schedulable}(s_i)$ to the discarded queue;
$CurSt(s_i)$=*Waited State;*
else
report fault of key task;
end if
insert s_i to S_{sched} based on the order of its priority;
end case
case $s_i \in S_{General}$
if s_i is schedulable in S_{sched} then
$CurSt(s_i)$=*Waited State;*
insert s_i to S_{sched} based on the order of its priority;
else
rejects s to the discarded queue;
end if
end case
Return.

For real-time task s_i=S that is added into the task set S_{sched}, if the following Eq. (5) and Eq. (6) are satisfied, this task is schedulable in task set S_{sched}.

$$\forall s_i \in Lower(s_i, S_{sched}) \cup \{s_i\},\ Y(s_i) \le L(s_i) \quad (9)$$

where $Lower(s_i, S_{sched})$ denotes the task set whose priority is lower than the task si in S_{sched}.

The core idea of the scheduling algorithm is to provide a security policy for new tasks joining the scheduled queue. The latest real-time task in the arrived queue will be obtained, and the corresponding processes will be given based on the task type. First, the algorithm is adjusted to ensure that the key tasks have satisfactory schedulability. If the task is schedulable, it will be input into the scheduled queue. However, if a key task is not schedulable, the normal tasks will give way for it. The algorithm is shown in Table 1.

6 Evaluations

6.1 Security Capability Evaluation

To implement the security evaluation for the proposed sustainable secure management mechanism, we perform two simulations. The first simulation is to use the traditional security data set to perform the evaluation, and the second simulation is to simulate an APT and evaluate the defense capability of the proposed mechanism. To implement comparisons, the mechanism in [35] is introduced.

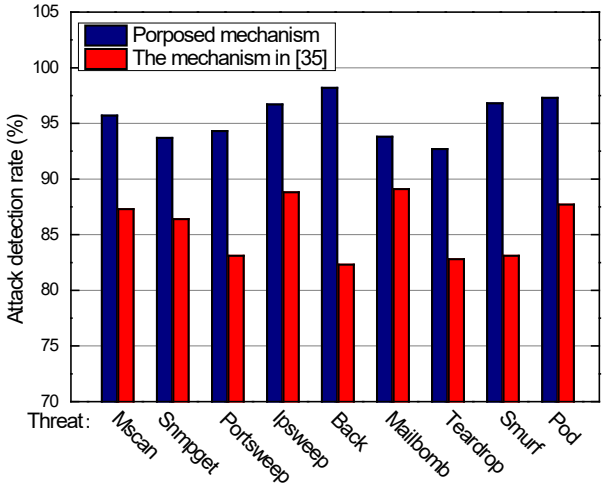


Fig. 10. Security evaluation based on traditional threat data sets.

First, the data set DARPA Intrusion Detection Evaluation Data for is used for testing and training. Nine weeks of network-based attacks of general background data are used for the training. To keep the universality of the data, the middle data of the general background data are applied. The evaluations of the defense capabilities of the proposed secure management mechanism are shown in

Fig. 10. The vertical axis in Fig. 10 denotes the threat detection rate. Moreover, nine kinds of threats are taken as the example from the DARPA Intrusion Detection Evaluation Data, which are denoted as a number of

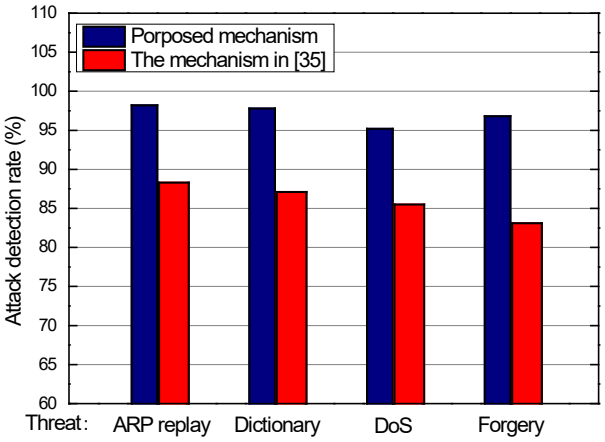


Fig. 11. Security evaluation-based traditional threat experiment.

groups of columns. As illustrated in Fig. 10, the threat detection rate of the proposed mechanism is much higher than that of the mechanism in [35]

In addition, an attack experiment was performed on a prototype system with two intelligent embedded systems. One machine acted as a server node, and the second machine was deployed to generate legitimate and hostile access attempts. As illustrated in Fig. 11, the threat detection rate of the proposed mechanism is higher than that of the mechanism in [35], and the increment is 11 percent points on average.

Next, we simulate an APT attack on an intelligent embedded system. The attack principle is based on Stuxnet. We evaluate the error of the APT feature discovery. To avoid the negative effects of the large span of the original data, the security factor data are standardized for extrema. For the original data D=(d_1, d_2, …, d_k), the extremum standardization process is computed as follows.

$$d_i' = \frac{d_i - \min(d)}{\max(D) - \min(D)} \quad (10)$$

6.2 Time Overhead Evaluation

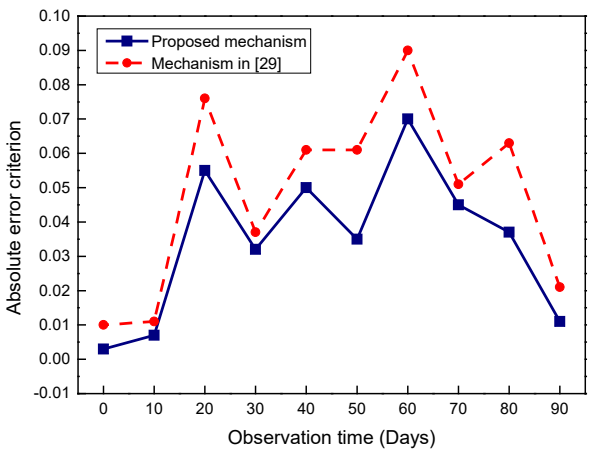


Fig. 12. Absolute error criterion of APT defense.

To evaluate the time overhead of the proposed sus-

tainable secure management mechanism, a test is performed on the system based on an environments using Ubuntu 11.04 and TRAI 3.8.1 with an Intel Core 2 Duo T6570 CPU with 4 GB RAM. The time overhead is shown in Fig. 13. The time overhead is acceptable.

7 Conclusion

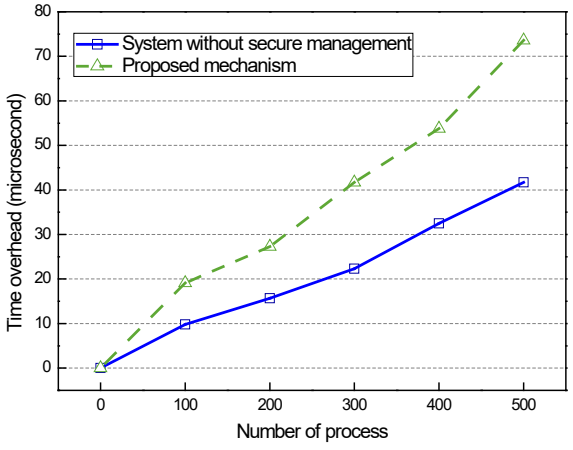


Fig. 13. Time overhead.

When an APT is present in smart manufacturing, the traditional secure defense components are no longer in force, because of the concealment, latency and longterm entanglement. This paper proposes a novel and sustainable secure management mechanism that can perform sustainable threat intelligence analysis and continuous secure resource management. The proposed mechanism is significant to improve the security of smart manufacturing. Edge artificial intelligence is a novel technology, which provides a feasible way to realize efficient defense at the edge of smart manufacturing. In the future, we will aim to propose edge artificial intelligence based defense approach to enhance the security capability of smart manufacturing.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 61431008, 61831007 and partially supported by the JSPS KAKENHI Grant Number JP19K20250, JP16K00117, KDDI Foundation. Mianxiong Dong is the corresponding author.

References

[1] F. Tao, J. Cheng, and Q. Qi, "IIHub: An Industrial Internet-of-Things Hub Toward Smart Manufacturing Based on Cyber-Physical System," IEEE Transactions on Industrial Informatics, vol. 14, no. 5, pp. 2271-2280, 2018.

[2] P. Lalanda, D. Morand, S. Chollet, "Autonomic Mediation Middleware for Smart Manufacturing," IEEE Internet Computing, vol. 21, no. 1, pp. 32-39, 2017.

[3] J. Wu, M. Dong, K. Ota, J. Li, Z. Guan, "Big Data Analysis based Security Cluster Management for Optimized Control Plane in Software Defined Networks," IEEE Transactions on Network and Service Management, vol. 15, no. 1, pp. 27-38, 2018.

[4] X. Cheng, Y. Wu, G. Min, A.Y. Zomaya, "Network Function Virtualization in Dynamic Networks: A Stochastic Perspective," IEEE Journal on Selected Areas in Communications, vol. 36, no. 10, pp. 2218-2232, 2018.

[5] L. Li, K. Ota, and M. Dong, "DeepNFV: A Light-weight Framework for Intelligent Edge Network Functions Virtualization," IEEE Network, DOI: 10.1109/MNET.2018.1700394, to be published.

[6] J. Wu, M. Dong, K. Ota, J. Li, Z. Guan, "FCSS: Fog-Computing-based Content-Aware Filtering for Security Services in Information-Centric Social Networks," IEEE Transactions on Emerging Topics in Computing, DOI: 10.1109/TETC.2017.2747158, pp. 1-12, 2017.

[7] L. Li, K. Ota, and M. Dong, "Deep Learning for Smart Industry: Efficient Manufacture Inspection System With Fog Computing," IEEE Transactions on Industrial Informatics, vol. 14, no. 10, pp. 4665- 4673, 2018.

[8] Y. Wu, F. Hu, G. Min, and A. Zomaya (eds.), Big Data and Computational Intelligence in Networking, Taylor & Francis/CRC, ISBN: 9781498784863, 2017.

[9] Y. Zuo, Y. Wu, G. Min, L. Cui, "Learning-based Network Path Planning for Traffic Engineering," Future Generation Computer Systems, vol. 92, pp. 59-67, 2019.

[10] Y. Ma, Y. Wu, J. Ge, J. Li, "An Architecture for Accountable Anonymous Access in the Internet-of-Things Network," IEEE Access, vol. 6, pp. 14451-14461, 2018.

[11] B. Wang, Z. Chang, Z. Zhou and T. Ristaniemi, "Reliable and Privacy-Preserving Task Recomposition for Crowd sensing in Vehicular Fog Computing," in Proc. IEEE 87th Vehicular Technology Conference (VTC 2018 Spring), 2018.

[12] X. Du, H. H. Chen, L. Zhu, J. Li and Z. Chang, "Security and Privacy in Wireless IoT," vol. 25, no. 6, pp. 10-11, 2018.

[13] E. Baize, "Developing Secure Products in the Age of Advanced Persistent Threats," IEEE Security & Privacy, vol. 10, no. 3, pp. 88-92, 2012.

[14] Q. Zhu, S. Rass, "On Multi-Phase and Multi-Stage Game-Theoretic Modeling of Advanced Persistent Threats," IEEE Access, vol. 6, pp. 13958-13971, 2018.

[15] A. Nourian, S. Madnick, "A Systems Theoretic Approach to the Security Threats in Cyber Physical Systems Applied to Stuxnet," IEEE Transactions on Dependable and Secure Computing, vol. 15, no. 1, pp. 2-13, 2018.

[16] G. Bonfante, J. Y. Marion, F. Sabatier and A. Thierry, "Analysis and Diversion of Duqu's Driver," in Proc. 8th International Conference on Malicious and Unwanted Software: "The Americas" (MALWARE 2013), Fajardo, USA, 2013.

[17] M. Cereia, I. C. Bertolotti, and S. Scanzio, "Performance of a Real-Time EtherCAT Master under Linux," IEEE Transactions on Industrial Informatics, vol. 7, pp. 679-687, 2011.

[18] [18] Y. Lan, T. Han, "SADP: Security Assurance Development Process for Building Reliable Linux-based Operating System," in Proc. 2015 IEEE International Conference on Computer and Communications (ICCC), Beijing, China, 2015, 50-55.

[19] S. Zhang, A. Zhang, J. Wu, L. Guo, J. Li, and B. Pei, "A Layered and Componentized Security Architecture for Linux based Mobile Network Elements," in Proc. 2015 Ninth International Conference on Frontier of Computer Science and Technology (FCST), Dalian, China, 2015, pp. 330-334.

[20] G. Bansod, N. Raval, and N. Pisharoty, "Implementation of a New Lightweight Encryption Design for Embedded Security," IEEE Transactions on Information Forensics and Security, vol. 10, no. 10, pp. 142-151, 2015,

[21] H. D. Ly, T. Liu, Y. Blankenship, "Security Embedding Codes," IEEE

Transactions on Information Forensics and Security, vol. 7, Feb. 2012, pp. 148-159.

[22] C. J. Cheng, C. C. Wang, W. C. Ku, T. F. Chen, and J. S. Wang, "A Scalable High-Performance Virus Detection Processor against a Large Pattern Set for Embedded Network Security," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 20, no. 5, pp. 841-854, 2012.

[23] A. S. Iyengar, S. Ghosh, K. Ramclam, "Domain Wall Magnets for Embedded Memory and Hardware Security," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 5, Mar. 2015, pp. 40-50.

[24] K. Erwinski, M. Parocki, L. M. Grzesiak, and K. Karwowski, and A. Wawrzak, "Application of Ethernet Powerlink for Communication in a Linux RTAI Open CNC System," IEEE Transactions on Industrial Electronics, vol. 60, pp. 628-636, Feb. 2013.

[25] G. Doukas, K. Thramboulidis, "Real-time-Linux-based Framework for Model-Driven Engineering in Control and Automation," IEEE Transactions on Industrial Electronics, vol. 58, no. 3, pp. 914-924, Mar. 2011.

[26] A. Barbalace, A. Luchetta, G. Manduchi, M. Moro, A. Soppelsa, and C. Taliercio, "Performance Comparison of VxWorks, Linux, RTAI, and Xenomai in a Hard Real-Time Application," IEEE Transactions on Nuclear Science, vol. 55, no, 1, pp. 435-439, Feb. 2008.

[27] K. G. Erickson, "NSTX-U advances in real-time C++11 on Linux," IEEE Transactions on Nuclear Science, vol. 62, no. 4, pp. 1758-1765, Aug. 2015.

[28] M. Asberg, T. Nolte, M. Joki, J. Hogbrink, and S. Siwani, "Fast Linux Bootup using Non-intrusive Methods for Predictable Industrial Embedded Systems," in Proc. IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA), Cagliari, Italy, 2013, pp. 1-8.

[29] P. Amontamavut, Y. Nakagawa, and E. Hayakawa, "Separated Linux process logging mechanism for embedded systems," in Proc. 2012 IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, Seoul, South Korea, 2012, pp. 1-4.

[30] M. Sojka, P. Pisa, and Z. Hanzalek, "Performance evaluation of Linux CAN-related system calls," in Proc. 2014 10th IEEE Workshop on Factory Communication Systems (WFCS), Toulouse, France, 2014, pp. 1-8.

[31] V. Nikolyenko, L. Libman, "Coop80211: Implementation and evaluation of a SoftMAC-based Linux Kernel Module for Cooperative Retransmission," in Proc. 2011 IEEE Wireless Communications and Networking Conference (WCNC), Quintana-Roo, Mexico, 2011, pp. 239-244.

[32] T. Vollmer, M. Manic, "Cyber-Physical System Security with Deceptive Virtual Hosts for Industrial Control Networks," IEEE Transactions on Industrial Informatics, vol. 10, no. 2, pp. 1337-1347, 2014.

[33] A. Valenzano, "Industrial Cybersecurity: Improving Security Through Access Control Policy Models," IEEE Industrial Electronics Magazine, vol. 8, Jun. 2014, pp. 1932-4529.

[34] M. Cheminod, L. Durante, A. Valenzano, "Review of Security Issues in Industrial Networks," IEEE Transactions on Industrial Informatics, vol. 9, pp. 1551-3203, Feb. 2013.

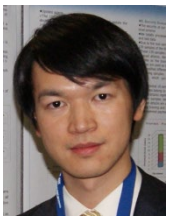
[35] N. Mousafa, E. Adi, B. Turnbull and J. Hu, "A New Threat Intelligence Scheme for Safeguarding Industry 4.0 Systems," IEEE Access, vol. 6, pp. 32910-32924, 2018.

[36] Y. Li, W. Dai, J. Bai, X. Gan, J. Wang, and X. Wang, "An IntelligenceDriven Security-Aware Defense Mechanism for Advanced Persistent Threats," IEEE Transactions on Information Forensics and Security, vol. 14, no. 3, pp. 646-661, 2019.

[37] E. Santos, H. Nguyen, F. Yu, K. J. Kim, D. Li, J. T. Wilkinson, A. Olson, J. Russell, and B. Clark, "Intelligence Analyses and the Insider Threat," IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 42, no. 2, pp. 331-347, 2012.

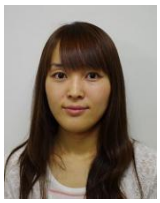
[38] Y. Ma, Y. Wu, J. Li, and J. Ge, "APCN: A Scalable Architecture for Balancing Accountability and Privacy in Large-scale Content-based Networks," Information Sciences, DOI: 10.1016/j.ins.2019.01.054, 2019.

[39] C. Huang, G. Min, Y. Wu, Y. Ying, K. Pei, and Z. Xiang, "Time Series Anomaly Detection for Trustworthy Services in Cloud Computing Systems," IEEE Transactions on Big Data, DOI: 10.1109/TBDATA.2017.2711039, 2017.



Jun Wu received the Ph.D. degree in information and telecommunication studies from Waseda University, Japan, in 2011. He was a Post-Doctoral Researcher with the Research Institute for Secure Systems, National Institute of Advanced Industrial Science and Technology (AIST), Japan, from 2011 to 2012. He was a Researcher with the Global Information and Telecommunication Institute, Waseda University, Japan, from 2011 to 2013. He is currently an associate professor of School of Cyber Security, Shanghai Jiao Tong University, China. He is also the vice director of National Engineering Laboratory for Information Content Analysis Technology, Shanghai Jiao Tong University, China. He is the chair of IEEE P21451-1-5 Standard Working Group. He has hosted and participated in a lot of research projects including National Natural Science Foundation of China (NFSC), National 863 Plan and 973 Plan of China, Japan Society of the Promotion of Science Projects (JSPS), etc. His research interests include the advanced computing, communications and security techniques of software-defined networks (SDN), information-centric networks (ICN) Energy Internets, Internet of Things (IoT), etc. He has been a Guest Editor of the IEEE Sensors Journal. He is an Associate Editor of the IEEE Access. He is a member of IEEE.

Mianxiong Dong received B.S., M.S. and Ph.D. in Computer Science and Engineering from The University of Aizu, Japan. He is currently an Associate Professor in the Department of Information and Electronic Engineering at the Muroran Institute of Technology, Japan. He was a JSPS Research Fellow with School of Computer Science and Engineering, The University of Aizu, Japan and was a visiting scholar with BBCR group at University of Waterloo, Canada supported by JSPS Excellent Young Researcher Overseas Visit Program from April 2010 to August 2011. Dr. Dong was selected as a Foreigner Research Fellow (a total of 3 recipients all over Japan) by NEC C&C Foundation in 2011. His research interests include Wireless Networks, Cloud Computing, and Cyber-physical Systems. He has received best paper awards from IEEE HPCC 2008, IEEE ICESS 2008, ICA3PP 2014, GPC 2015, IEEE DASC 2015, IEEE VTC 2016-Fall, FCST 2017, 2017 IET Communications Premium Award and IEEE ComSoc CSIM Best Conference Paper Award 2018. Dr. Dong serves as an Editor for IEEE Transactions on Green Communications and Networking (TGCN), IEEE Communications Surveys and Tutorials, IEEE Network, IEEE Wireless Communications Letters, IEEE Cloud Computing, IEEE Access, as well as a leading guest editor for ACM Transactions on Multimedia Computing, Communications and Applications (TOMM), IEEE Transactions on Emerging Topics in Computing (TETC), IEEE Transactions on Computational Social Systems (TCSS). He has been serving as the Vice Chair of IEEE Communications Society Asia/Pacific Region Information Services Committee and Meetings and Conference Committee, Leading Symposium Chair of IEEE ICC 2019, Student Travel Grants Chair of IEEE GLOBECOM 2019, and Symposium Chair of IEEE GLOBECOM 2016, 2017. He is the recipient of IEEE TCSC Early Career Award 2016, IEEE SCSTC Outstanding Young Researcher Award 2017, The 12th IEEE ComSoc Asia-Pacific Young Researcher Award 2017, Funai Research Award 2018 and NISTEP Researcher 2018 (one of only 11 people in Japan) in recognition of significant contributions in science and technology. He is currently the Member of Board of Governors and Chair of Student Fellowship Committee of IEEE Vehicular Technology Society, and Treasurer of IEEE ComSoc Japan Joint Sections Chapter.



Kaoru Ota was born in Aizu-Wakamatsu, Japan. She received M.S. degree in Computer Science from Oklahoma State University, USA in 2008, B.S. and Ph.D. degrees in Computer Science and Engineering from The University of Aizu, Japan in 2006, 2012, respectively. She is currently an Assistant Professor with Department of Information and Electronic Engineering, Muroran Institute of Technology, Japan. From March 2010 to March 2011, she was a visiting scholar at University of Waterloo, Canada. Also she was a Japan Society of the Promotion of Science (JSPS) research fellow with Kato-Nishiyama Lab at Graduate School of Information Sciences at Tohoku University, Japan from April 2012 to April 2013. Her research interests include Wireless Networks, Cloud Computing, and Cyber-physical Systems. Dr. Ota has received best paper awards from ICA3PP 2014, GPC 2015, IEEE DASC 2015, IEEE VTC 2016-Fall, FCST 2017, 2017 IET Communications Premium Award and IEEE ComSoc CSIM Best Conference Paper Award 2018. She is an editor of IEEE Transactions on Vehicular Technology (TVT), IEEE Communications Letters, Peer-to-Peer Networking and Applications (Springer), Ad Hoc & Sensor Wireless Networks, International Journal of Embedded Systems (Inderscience) and Smart Technologies for Emergency Response & Disaster Management (IGI Global), as well as a guest editor of ACM Transactions on Multimedia Computing, Communications and Applications (leading), IEEE Internet of Things Journal, IEEE Communications Magazine, IEEE Network, IEEE Wireless Communications, IEEE Access, IEICE Transactions on Information and Systems, and Ad Hoc & Sensor Wireless Networks (Old City Publishing). She is the recipient of IEEE TCSC Early Career Award 2017, and The 13th IEEE ComSoc Asia-Pacific Young Researcher Award 2018.

Jianhua Li is a professor/Ph.D. supervisor and the dean of School of Cyber Security, Shanghai Jiao Tong University, Shanghai, China. He is also the director of National Engineering Laboratory for Information Content Analysis Technology, the director of Engineering Research Center for Network Information Security Management and Service of Chinese Ministry of Education, and the director of Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, China. He is the vice president of Association of Cyber Security Association of China. He got his BS, MS and Ph.D. degrees from Shanghai Jiao Tong University, in 1986, 1991 and 1998, respectively. He was the chief expert in the information security committee experts of National High Technology Research and Development Program of China (863 Program) of China. He was the leader of more than 30 state/province projects of China, and published more than 300 papers. He published 6 books and has about 20 patents. He made 3 standards and has 5 software copyrights. He got the Second Prize of National Technology Progress Award of China in 2005. His research interests include information security, signal process, computer network communication, etc.

Wu Yang received the Ph.D. degree in computer system architecture specialty from the Computer Science and Technology School, Harbin Institute of Technology. He is currently a Professor and a Doctoral Supervisor with Harbin Engineering University. His main research interests include wireless sensor network, peer-to-peer network, and information security. He is a member of ACM and a Senior Member of CCF.