Hybrid Intrusion Detection Mechanisms for Integrated Electronic Systems

Qi Qiao
School of software engineering
East China Normal University
Shanghai, China
52174501018@stu.ecnu.edu.cn

Daojing He
School of software engineering
East China Normal University
Shanghai, China
djhe@sei.ecnu.edu.cn

Yun Gao
School of software engineering
East China Normal University
Shanghai, China
51194501121@stu.ecnu.edu.cn

Sencun Zhu
Department of Computer Science & Engineering
Pennsylvania State University
University Park, United States
sxz16@psu.edu

Jiahao Gao
School of software engineering
East China Normal University
Shanghai, China
382906025@qq.com

Sammy Chan
Department of Electronic Engineering
City University of Hong Kong
Hong Kong, China
eeschan@cityu.edu.hk

***Abstract*—While integrated electronic systems (IESs) are widely used in military and civilian applications, their security issues are barely studied. By analyzing the architecture of the system and the characteristics of bus communication, this paper proposes an intrusion detection method based on the message sequence and behavioral rules of subsystems. According to the bus protocol, messages are divided into periodic and aperiodic messages. For the previous, we adopt sequence analysis and propose an algorithm that extract the sequence intelligently to determine if there are anomalies. For aperiodic messages, we detect the anomalies by modeling the system behaviors as decision trees. Through implementing experiments on our simulation system, we demonstrate that the proposed detection is more accurate than the existing schemes while incurring both lower false negative rate and lower false positive rate.**

***Index Terms*—Intrusion detection, Integrated Electronic System, 1553B bus, security**

I. Introduction

The integrated electronic system (IES) is a resource-constrained system that integrates the controller and multiple components. It is a standardized and extensible architecture that schedules resources across the whole system in an efficient, collaborative manner to improve the resource utilization. It has been widely used to coordinate the complicated subsystems and payloads in satellites as well as the International Space Station, civil aircraft, armored vehicles and so on.

As shown in Fig. 1, the IES is a two-tier distributed architecture. The primary bus is the main communication bus, which usually adopts the MIL-STD-1553B bus [1], [2], connecting the central management unit and the subsystems. Although other high-speed buses have been introduced [3], [4], the replacement of internal bus would change the whole architecture of the system. Therefore, the 1553B bus is still the most widespread used standard for the IES [5]. This paper uses the 1553B bus as a typical example to illustrate the work mechanism of the IES and studies intrusion detection for the IES. The secondary bus is deployed inside each subsystem and is responsible for communication between the modules. In most cases, the IES also establishes a wireless communication channel with the ground station (GS) through the wireless access point, which works as a subsystem.

In the past few decades, researches on the IES have been focusing on its reliability, which guarantees the system can work properly in extreme physical environments (e.g., desert, atmosphere, outer space, etc.), while ignoring the information security. With frequent occurrence of security incidents and the advance in attack methods [6] such as back-doors [7] and Advanced Persistent Threats (APTs) [8], [9], the study on system security has attracted more and more attention. Moreover, due to the particularity of the working environment and tasks, once these systems are compromised, more adverse effects can be caused than other general security incidents.

We classify attacks against the IES into three categories by the attack target. The first is attacks on the GS, such as APTs. Attackers can access the IES by controlling the GS, and then they can obtain information or send instructions [10], [11]. There already exist many security solutions that protect the GS [7], [8], so our work does not specifically consider

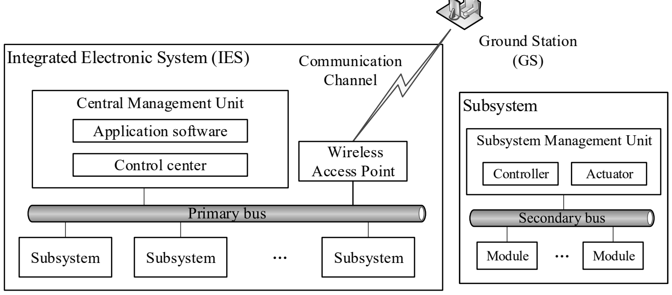


Fig. 1. The general architecture of IESs.

978-1-7281-1062-2/20/$31.00 2020 © IEEE

GS security. The second is attacks against the communication channels. Because the IES generally performs special-purpose tasks, a dedicated link is often used to establish communication with the GS, which has a high security level. Therefore, such attacks also are not discussed in this paper. The third category of attacks comes from the inside modules of IESs. IES hardware are usually provided by different vendors, so back-doors designed for malicious purposes or for management needs may become a logic bomb that will threaten the information security of the system. There are few reports of works related to defending against such attacks.

Thus, this paper focuses on protecting the IES from potential internal threats at the primary bus level. Note that although it is also important to study the security of the subsystem modules and the secondary bus, the main contribution of this paper is to protect the security of communication between the central management unit and subsystems.

There are challenging issues in developing intrusion detection methods for resource-constrained systems like the IES with 1553B bus, as discussed below:

- Most of IESs' limited hardware resources are devoted to ensure its normal operation, and very little are available for security protection. This brings the issue of how to reduce the computation and storage cost of a detection algorithm without sacrificing detection efficiency.
- Unlike general resource-constrained systems, bus-based IESs have their own specific protocols and communication modes. The characteristics of the bus protocol, such as master-slave communication in 1553B bus, should be considered while designing intrusion detection scheme, to make it suitable for the bus system.

We proposes an intrusion detection method based on the message sequences and behavioral rules of subsystems for the IES with 1553B bus. According to the characteristics of the bus protocol, we handle periodic and aperiodic messages separately. Then experiments are implemented on our own simulation platform. The results prove that the proposed method can perform effectively on the IES. Compared with the method of Stan *et al.* [5], we also demonstrate that our method improves the detection accuracy without trading off false positive rate and false negative rate in the same case of limited resources. The main works of this paper are as follows:

- For periodic messages, we propose an algorithm that can intelligently extract the message sequence by analyzing the bus traffic without knowing the details of system and protocol, which serves as the basis for detecting whether there are anomalies. And this kind of feature extraction makes full use of the bus features to save storage.
- For aperiodic messages, the detection of anomalies is performed through the behavior rules of each subsystem. By parsing the communication data packets, we can extract system behavioral rules to build decision trees.
- We implement a simulation platform based on a real 1553B bus for experiments. The training dataset and test dataset for evaluating the proposed method are collected from this simulation platform.

II. Background and Reference Model

A. A brief introduction to the 1553B bus

The latest version of the 1553B bus, released in 1978, was designed with little regard for information security, thus vulnerable to cyber attacks such as denial-of-service (DoS) attacks that have been studied since 1980 [5], [12]. Although the designers later stated in the 1553B guide book that more attention should be paid to protecting the security of the bus communication [13], there is no more specific guidelines provided because of military confidentiality requirements. The 1553B bus is still exposed to security attacks.

The 1553B bus is a centralized time-division serial bus that adopts a dual redundant system with two transmission channels to ensure good fault tolerance. There exist three types of terminals in the 1553B bus, namely bus controller (BC), remote terminal (RT), and bus monitor (BM). The communication between the BC and a RT is master-slave communication, which means each session can only be initiated by the BC, and then RTs respond. The BM is not involved in communication, but records the traffic. In an IES, the central management unit usually acts as the BC, and each subsystem connects to the bus through the interface of each RT.

The communication protocol of 1553B bus defines three types of communication words: *command word*, *data word*, and *status word*. A complete communication data in the bus system is called a *message*, usually contains all three communication words and each message starts with a command word. Messages are divided into periodic and aperiodic messages. Periodic messages are sent at regular intervals (or time periods), and each periodic message is sent at least once in a primary time frame (the length of the primary time frame depends on the longest message time period). Aperiodic messages are event-driven, because they are triggered by some critical system states or sent by the GS, so they have no obvious temporal features. In addition, the number of periodic messages is much more than aperiodic messages.

B. System Model

1) Reference IES: We consider an IES using the 1553B bus as our reference model, as shown in Fig. 2. The system has a BC and a series of subsystems with different functions, and they communicate through a 1553B bus. BC has the authority

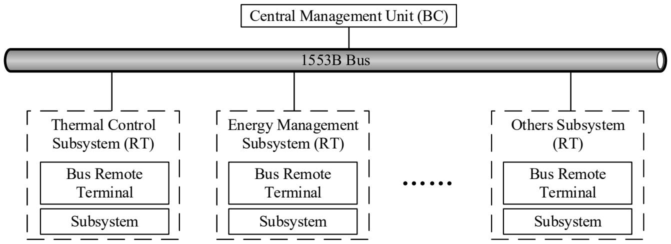


Fig. 2. Reference model of the IES.

of BM, that is, it can get all bus messages. We introduce two general RTs in detail, which are the thermal control subsystem and the energy management subsystem. Almost all IESs deploy these two subsystems, such as armored vehicles and spacecraft. Note that our reference model is designed as a complete IES, but only take the two representative subsystems for ease of illustration.

The basic workflow of the BC and RTs strictly follows the provisions of the 1553B bus protocol. In each time period, the BC initiates a polling to inquire all RTs in turn, and then determines whether to send control commands based on the returned values. Messages containing control commands are referred to as *remote command packet*, and messages containing state values are referred to as *telemetry data packet*.

In our reference model, the polling period is set as 100 ms, that is the current states of system are inquired every 100 ms by the BC. Taking the thermal control subsystem as an example, if the temperature value returned by the RT is below the threshold set by the administrator, the BC needs to send an instruction (i.e., a remote command packet) to turn on the heater. Otherwise, an instruction to turn off the heater will be sent. The process of inquiring the energy management subsystem is similar to the thermal control subsystem.

2) Trust model: In our reference model, we assume that only the BC is fully trusted. Because in the 1553B bus system, the BC has the highest control authority. Once the BC is compromised, the entire system will fall. Other RTs may have back-doors or have been compromised. This means that only the BC will follow the 1553B bus protocol strictly, so that it will periodically initiate polling and send control commands based on the statuses of the system. But the RTs may not work properly or even cause some damage to the system.

3) Threat model: This paper focuses on threats from within the system. For the threat model, the attack source may be a compromised RT or a terminal that illegally accesses the bus. That means there exist two ways for the adversary to launch attacks. It can manipulate the RT through the back-doors to perform illegal operations, or act as a terminal to access the bus without permission through some physical means.

Threats to the IES include integrity attacks, availability attacks, and confidentiality attacks. But for communications in IESs are not encrypted, confidentiality attacks are easy to implement and difficult to defend with intrusion detection methods, which are not described in detail here.

Integrity attacks can tamper with communication data. Typical integrity attacks include forgery attacks and tampering attacks. The method proposed in this paper is effective in detecting such attacks. Here is an example of forgery attacks to illustrate. A compromised RT may send a message to the thermal control subsystem every 300ms to turn off the heater, so that the heater is closed all the time. This will cause the IES to work at low temperatures, which may damage the system hardware and hence affect the normal operation of the system.

Availability attacks disable system functionality. DoS attacks can damage availability, which can be classified into bus DoS and terminal DoS for 1553B bus. The specific method is to send a large amount of data to occupy the bus or make RTs busy (such as sending data frequently), so that normal communication data cannot be transmitted or processed.

4) Design goal: Considering the particularity of the bus protocol in the IES, the proposed intrusion detection scheme must make full use of internal communication mechanisms such as command/response and polling to reduce computation and storage costs, so as not to overload the system with limited resources. Therefore, recent deep learning algorithms for anomaly detection, like DeepLog [14] and many other solutions [15], are too heavy to be deployed on the IES. Another important design goal is to maximize detection accuracy. The failures of safety critical IES are unbearable, so the detection algorithm needs to improve the accuracy rate as much as possible. At the same time, the false negative rate and false positive rate should also remain low. Last, due to the needs of special application scenarios such as armored vehicles, the specific design of the IES may keep confidential for security designers. Thus the proposed intrusion detection scheme should be analyzed and judged based on bus traffic without understanding the system design.

III. Proposed Method

We have known that the communication messages consist of a large number of periodic messages and a few aperiodic messages, and the aperiodic messages are more difficult to process because they have no obvious temporal characteristics. Thus we separate the two types of messages. For periodic messages, the detection scheme is based on the command word sequence, which is extracted by the historical bus data. The extraction algorithm we designed can get the sequence intelligently in various IES deployments without knowing the specific designs of the systems. Using this command word sequence, we can detect the anomaly of periodic messages without knowing the workflow of the system. As for aperiodic messages, we propose a method to construct decision trees of each RT by analyzing the packets in application layer. The decision trees can describe the behavioral rules of each RT to predict aperiodic messages.

A. Communication messages

The 1553B protocol specifies that a complete message comprises with the command words and data words sent by the BC, and the corresponding status words or data words returned by the RT. Each message contains at least one command word, so we only take the command word in messages as the detection object for lightweight purpose.

In periodic messages, the sequence of periodic command words is predefined and each word will be sent at least once in every cycle. While in aperiodic messages, an aperiodic command word may be triggered by the RT status or sent by a GS. If it is triggered, the aperiodic command word will immediately follow a periodic command word so that its appearance is traceable. But if it is controlled by a GS, this command word usually does not have identifiable features such as time or sequence. Therefore, in our solution, such messages sent by the GS will be identified as abnormal and reported to the GS for identification.

We build an IES simulation platform (which will be introduced in section IV) and to collect experimental data. The statistical result of numbers of command words in 200 minutes is shown in Fig. 3, where Cmd_1 to Cmd_6 are periodic commands, and Cmd_7 to Cmd_11 are aperiodic commands. We can see that numbers of periodic command words differs by one because the last polling was not completed when the data was collected. While numbers of aperiodic command words varies and does not show any correlation. In addition, the total number of aperiodic command words is also much smaller than that of periodic command words.

B. Feature Definition

In the Markov-based anomaly detection algorithm proposed by Stan *et al.* [5], the characteristics of the periodic message is defined as an 8-tuple, that is (*source terminal address, source sub-address, destination terminal address, destination sub-address, channel, number of data words, mode code, time interval*). The description of the aperiodic message removes the "time interval" field from this 8-tuple. The required storage space for this representation is 137 bits.

In our scheme, we simplify the feature tuple by analyzing the 1553B protocol. The protocol stipulates that each communication must be initiated by the BC, so the communication between RTs must be forwarded by the BC. That is, for the feature tuple of the command word, the source address, including the source terminal address and the source sub-address, is fixed to the address of the BC. Therefore, we do not use the source address as the characteristic field, but only use one bit to indicate whether the command word is sent or received. The periodic message feature tuple in our method is defined as a 6-tuple, that is (*terminal address, sub-address/mode code, transmission/reception, number of data words, channel A or B, minimum time interval*). The feature for aperiodic messages is a 5-tuple, that is (*terminal address, sub-address/mode code, send/receive, number of data words, channel A or B*). The details of features are shown in Table I. It can be seen that the required storage space for our method is 42 bits, which is only 30% of that in Stan's scheme.

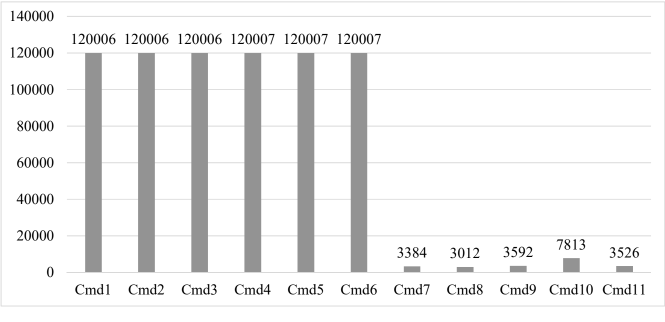


Fig. 3. Numbers of command words in 200 minutes (partial).

C. Detection algorithm for periodic messages

The periodic messages of the 1553B bus are strictly transmitted according to a predefined sequence of command words. The key premise of using sequence analysis to detect anomalies is to obtain the complete sequence of periodic command words, and then detect whether there are anomalies based on the extracted message sequence.

1) Message sequence extraction: According to the characteristics of periodic messages, we just need to count the number of messages in each interval to find out the polling period. Then the messages will be reordered by traversing the bus log. So that we can extract the sequence of periodic messages through bus logs without knowing the specific design of the system. This also adapts to system changes. For example, if any of the RT is added or removed, our extraction method can also modify the periodic message sequence.

2) Detection phase: It can be checked quickly whether a new message from the bus is in the order of extracted sequence. If so, the "time interval" field of the message will be checked again to detect whether the message is a fake command word inserted before the normal command word. Otherwise, the message enters the next detection phase as an aperiodic message.

D. Detection algorithm for aperiodic messages

Here we mainly detect aperiodic messages triggered by certain states. For the aperiodic message sent by the GS, we have already mentioned that it cannot be detected because it does not have any time and order characteristics. This type of aperiodic messages will be reported and the system administrator will determine if there is an exception.

We use the algorithm based on behavioral rules to process aperiodic messages. The main idea of this method is to find out the system operations after reaching a certain state. The first step is to parse the communication packets to get the data field that represents the state changes of the system. System behavior decision trees are then built for each application of subsystems (e.g., the "temperature control" application of the

TABLE I
DEFINITION OF FEATURES.

Types of features	Fields	Values	Storage (bit)
Features of periodic command word	Terminal address	0-31	8
	Sub-address or Mode code	0-31	8
	Channel	0 (A), 1 (B)	1
	Transmission or Reception	0 (Recicve), 1 (Send)	1
	Number of data words	0-32	8
Features of time	Time interval	Numeric	16
Total storage			42

thermal control subsystem) with the system state value before each aperiodic message as the critical state. This decision trees can be used to detect whether aperiodic messages appearing in bus traffic are normal.

1) Parsing packets and feature extraction: In the reference model, the data words of the application layer are divided into telemetry data packets and remote command packets. More specifically, we analyze telemetry data packets to obtain the states of the system, and combine the analysis of remote control packets to construct system behavior decision trees. The two packet formats are different. The next question is how to extract the required data from them.

a) Telemetry data packets: The detail of telemetry data packets is shown in Fig. 4(a). The telemetry data packet usually is a periodic message, which is sent by a RT in response to the polling message from the BC to obtain the status of the RT. There are three parts in the packet, that is, packet header, packet content and checksum for error controlling.

In most Internet protocols, since the packet size and content of the application layer can change dynamically, it is usually necessary to use a complex attack feature vector when employing a machine algorithm for its packet content anomaly detection. Differently, in the 1553B protocol, since the data field has a maximum of 32 data words, that is, the data field has a maximum of 32 features, a simpler algorithm can be used to detect abnormal data.

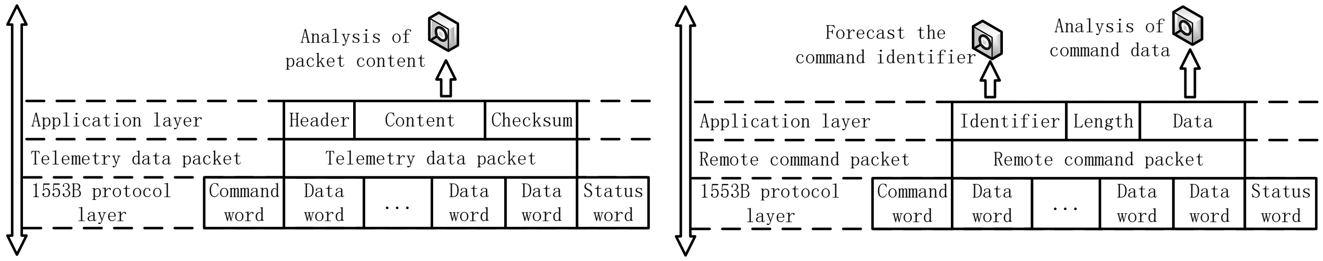
By analyzing data fields of the application layer protocol, we find that the size of the data field is fixed, and only some of the data words are different in different messages. Therefore, the telemetry data packets are sorted by different RTs. The telemetry data of each RT is extracted from a plurality of aperiodic messages. These data will be arranged according to the sending time. The contents of packets of each RT are arranged according to the time period. Using the packet content variation as the feature values, the next remote command packet can be predicted.

The feature extraction process for the telemetry data packet is as follows. Let the periodic messages $M_1 = \{m_{11}, m_{12}, ..., m_{1n}\}$ represent the n messages in the $1st$ period. Set $M_i = \{m_{i1}, m_{i2}, ..., m_{in}\}$ as the n messages in the $i-th$ period, where $m_{ij} = \{commandword_{ij}, dataword_{ij}, statusword_{ij}\}$, j represents the $j-th$ message in the period. $dataword_{ij} = \{data_1, data_2, ..., data_p\}$, where $1 \leqslant p \leqslant 32$. Each $data_p$ is 2-byte long. When analyzing the content of the packet in the periodic message, some of the bytes are unchanged, and other changed bytes which are continuous or discrete values. We extract the changed data words in the telemetry data package to obtain the critical states.

Next, we will introduce how to extract the characteristics of the $j-th$ periodic message. The $j-th$ periodic message of M in k periods are extracted to form $S_j = \{m_{1j}, m_{2j}, ..., m_{kj}\}$, and each m_{kj} may trigger an aperiodic message. Let $[AP]_j$ denote the aperiodic message triggered by periodic message j. We use $rawdata_{ij} = \{data_1, data_2, ..., data_p\}$ as the raw data of message j in the i_{th} period, where $1 \leqslant i \leqslant k$, $data_q \in \{fixed, continuous, discrete\}$, $1 \leqslant q \leqslant p$. Only the continuous and discrete data need to be extracted. Let OD_j as the original data of the $j-th$ periodic message, $OD_j = \{rawdata_{1j}, rawdata_{2j}, ..., rawdata_{kj}\}$. Then extract continuous and discrete values from OD_j, that is $rawdata_{ij} - rawdata_{(i-1)j} = \{..., Nonezero, ..., Zero, ...\}$. The data words corresponding to the non-zero values are extracted as the feature. Finally, the feature of the $j-th$ periodic message is represented as $T_j = \{..., data_p, ...\}$, where $data_p \in \{continuous, discrete\}$.

b) Remote command packets: The analysis of the remote command packet is shown in Fig. 4(b). In an IES, aperiodic messages usually are remote command packets for controlling the system. The remote command packet is triggered when the data field of the telemetry packet exceeds the pre-set threshold. The remote command packet has three fields, including command identifier, data length and command data (this part is optional). The command identifier indicates the function of the command, and its length is 8 bits. The data length indicates how many bytes of the data of the command. A length of 0 indicates that the instruction has no data. If the data length is n, the data length is followed by n bytes of data.

One of the RT has a variety of remote commands, and some of them have command data. We use the feature T_j extracted above to predict whether the $[AP]_j$ is abnormal. The aperiodic message j is expressed as $[AP]_j = \{command, len\}$, where the $command$ is a discrete value and len is a continuous value. The $[AP]_j$ is the unique label of the remote command packet. When $len = 0$, $[AP]_j = \{command\}$. In order to make the algorithm more lightweight, the tag of the remote command packet does not contain command data.



(a) The telemetry data packet.

(b) The remote command packet.

Fig. 4. Structure of the two kind of packets in application layer.

2) Building decision trees and detecting aperiodic messages: In order to intuitively understand the behavioral rules of each terminal of the bus, we adopt decision trees to represent the behavioral rule specifications. Taking the application of the thermal control subsystem, “temperature control”, as an example, we explain the specific behavior through the trained decision tree of the subsystem. As shown in Fig. 5, when the temperature is higher than -17.5 °C, no remote command is sent. That is, there is no aperiodic message after the periodic message. In the case where the temperature is lower than or equal to -17.5 °C, if the heater is on, then a command will be sent to turn off the heater. That is, an aperiodic message follows the periodic message. Otherwise, there is no remote command. In the case where the temperature is lower than or equal to -19.5 °C, if the heater is on, there is no remote command; otherwise, a command to turn on the heater is sent. The decision tree we build has the same behavioral rule specifications for each RT, so it can also be analyzed by the designers to verify the behaviors.

Each specific application of RT corresponds to a behavior specification decision tree. The training process includes two stages of feature selection and node generation. We choose the CART (Classification And Regression Tree) algorithm [16] to select a feature from the training data as the split criterion. The CART algorithm divides the features of the current data set according to the Gini coefficient, which is used to measure the uncertainty of random variables. When the Gini coefficient is 0, all data in the data set belong to the same category. Sub-nodes are then recursively generated from top to bottom according to the features selected by CART. The decision tree is stopped when no features satisfy the split condition. For the current application training set s, the Gini coefficient of each existing feature for this data set is calculated. For each feature A, take its value a, and divide s into s_1 and s_2 according to the sample point when $A \leqslant a$ is tested as “yes” or “no”, and calculate the Gini coefficient when $A \leqslant a$.

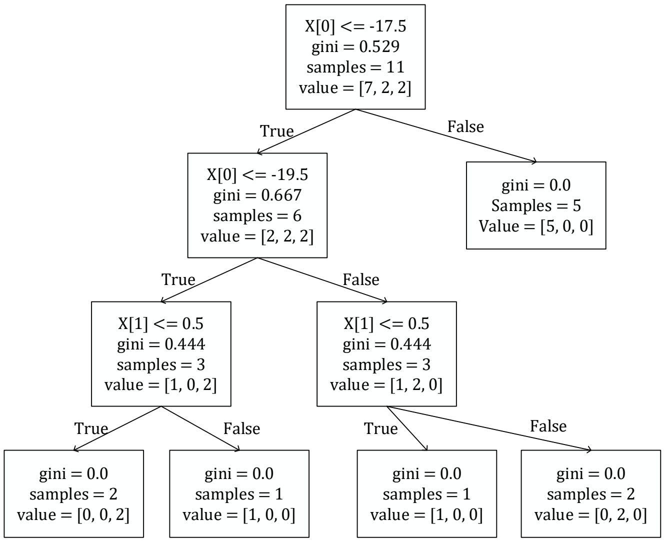


Fig. 5. A decision tree of the thermal control subsystem.

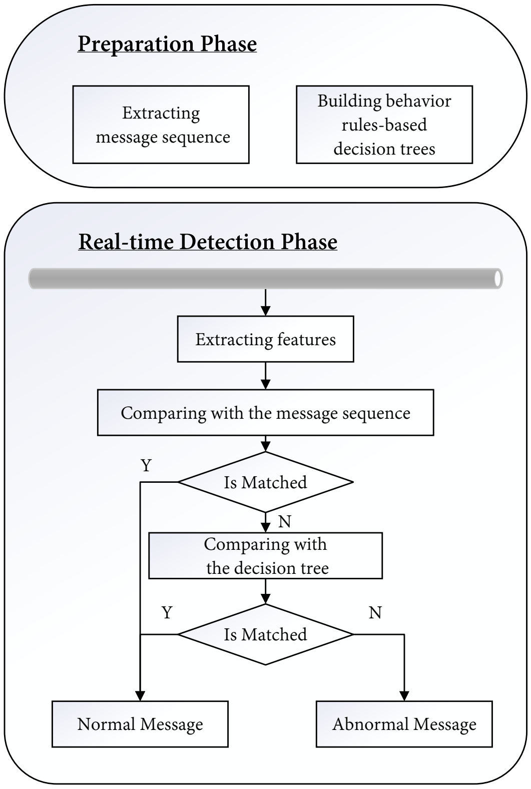

Preparation Phase
Extracting message sequence
Building behavior rules-based decision trees
Real-time Detection Phase
Extracting features
Comparing with the message sequence
Y
Is Matched
N
Comparing with the decision tree
Y
N
Is Matched
Normal Message
Abnormal Message

Fig. 6. Flow chart of proposed detection method.

In the real-time detection phase, we use the trained decision tree to determine whether an aperiodic message conforms to its expected behavior. If yes, the aperiodic message is normal.

E. Complete process of our detection method

As shown in Fig. 6, the process of our detection method has two phases. In the preparation phase, through analyzing the historical bus log, the period message sequence are extracted, and decision trees of each subsystem are built. In the real-time detection phase, a message from the bus traffic is compared with the extracted sequence. If the matching fails, it will be identified as an aperiodic message temporarily. Then the detection of aperiodic messages will start. If the detection still fails, it is determined to be an abnormal message and an alarm is issued. For data re-training, on the one hand, since the IES rarely changes after deployment, periodic message sequences and the behavior rules of subsystems also rarely change. On the other hand, retraining is not practical because it is difficult to evaluate the ground truth.

IV. Experiments and Comparison

We implemented a simulation system combining hardware and software to evaluate our proposed detection method. We also re-implemented a baseline comparison scheme and demonstrate that our method has better performance.

A. Simulation System with a real 1553B bus

Our software-and-hardware combined simulation method realizes the terminals through programming, and the communication data between the terminals is using a real 1553B bus [17]. This approach not only ensures that the system

retains the hardware characteristics of the real bus, but also makes the simulation system more flexible and can meet different experimental needs.

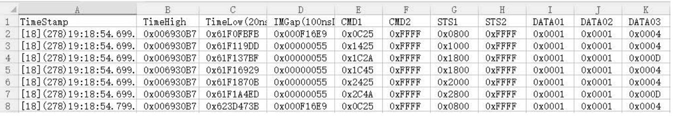
We implemented the simulation system based on the AltaView board. The AltaView board encapsulates the 1553B bus and its underlying protocol, while providing users with interfaces. The hardware part has three main components, including the 1553B bus, the coupler and the dual channel. The dual channel is used to communicate. In our simulation system, one channel for the normal system communication, the other channel transmits attack data. The coupler is used to connect the two channels, so that the attacker can connect in.

For the test set, we constructed a malicious adversary (as a RT) in the simulation system, but it also has permissions for BC and BM. That is, it has the highest authority to initiate communication arbitrarily and falsify data packets for attack. We implemented DoS attacks, forgery attacks, tampering attacks, and replay attacks. The specific implementation manners are as follows: each time a DoS attack is initiated, we randomly selected a message to send continuously without leaving the interval between messages, occupying the bus bandwidth so that the bus cannot work normally. For forgery attacks, we intercepted bus messages or generated new messages conforming to the 1553B bus protocol, and sent them in a period of 80ms. The tampering attack is to modify control commands in remote control packets, and send it in a cycle of 100ms. The replay attack replays the previous messages with a period of 80ms. Our test set contains multiple attacks of each type.

B. Experiments on the simulation system

1) Preprocessing data: The BM module is available independently from the AltaView board. Fig. 7 shows the raw data we collected by the BM, where the data words are too long to be fully displayed. We used the normal bus traffic to generate the command word sequences and build the behavior specification trees. There are 12,217 periodic messages and 446 aperiodic messages in the training set. They are all normal data. For the test set, it contains a total of 153,865 items, wherein 110,523 are normal periodic messages, 3,776 normal aperiodic messages, 33,422 DoS attacks, 2,423 forgery attacks, 1268 tampering attacks, and 2,453 replay attacks.

2) Extracting features:



	A	B	C	D	E	F	G	H	I	J	K
1	TimeStamp	TimeHigh	TimeLow(20ns	IMGap(100nsI	CMD1	CMD2	STS1	STS2	DATA01	DATA02	DATA03
2	[18](278)19:18:54.699.	0x006930B7	0x61F0FBFB	0x000F16E9	0x0C25	0xFFFF	0x0800	0xFFFF	0x0001	0x0001	0x0004
3	[18](278)19:18:54.699.	0x006930B7	0x61F119DD	0x00000055	0x1425	0xFFFF	0x1000	0xFFFF	0x0001	0x0001	0x0004
4	[18](278)19:18:54.699.	0x006930B7	0x61F137BF	0x00000055	0x1C2A	0xFFFF	0x1800	0xFFFF	0x0001	0x0001	0x000D
5	[18](278)19:18:54.699.	0x006930B7	0x61F16929	0x00000055	0x1C45	0xFFFF	0x1800	0xFFFF	0x0001	0x0001	0x0004
6	[18](278)19:18:54.699.	0x006930B7	0x61F1870B	0x00000055	0x2425	0xFFFF	0x2000	0xFFFF	0x0001	0x0001	0x0004
7	[18](278)19:18:54.699.	0x006930B7	0x61F1A4ED	0x00000055	0x2C4A	0xFFFF	0x2800	0xFFFF	0x0001	0x0001	0x000D
8	[18](278)19:18:54.799.	0x006930B7	0x623D473B	0x000F16E9	0x0C25	0xFFFF	0x0800	0xFFFF	0x0001	0x0001	0x0004

Fig. 7. Raw data collected from the simulation system (partial).

TABLE II
FEATURES OF PERIODIC MESSAGES.

Index	Command word	Terminal address	Mode code	Channel	Send/ Recive	Number	Time period
0	0x0C25	1	1	0	1	5	100
1	0x1425	2	1	0	1	5	100
2	0x1C2A	3	1	0	1	10	200
3	0x1C45	3	2	0	1	5	100
4	0x2425	4	1	0	1	5	100
5	0x2C4A	5	2	0	1	10	100
6	0x0841	1	2	0	0	1	-
7	0x1041	2	2	0	0	1	-
8	0x1866	3	3	0	0	6	-
9	0x2043	4	2	0	0	3	-
10	0x2865	5	3	0	0	5	-

TABLE III
FEATURES OF APERIODIC MESSAGES.

Index	Terminal address	Application No.	Data3	Heater state	Remote command
1	1	1	0x10	0	0
2	1	1	0x12	0	0
3	1	1	0x14	0	0
4	1	1	0x17	0	0
5	1	1	0x19	0	0
6	1	1	0x20	0	0xF2
7	1	1	0x21	1	0
8	1	1	0x18	1	0xF1
9	1	1	0x17	0	0
10	1	1	0x21	0	0xF2
11	1	1	0x19	1	0xF1

a) For periodic messages: In order to reduce the storage overhead, our proposed method combines the characteristics of 1553B bus communication, extracts only the command words as the detection data. Now we need to extract the required features from these raw data. The 4-tuple feature *(terminal address, sub-address/mode code, send/receive, number of data words)* is extracted from the command word. The *minimum time interval* features are extracted by the $TimeHigh$ and $TimeLow$ fields. The *channel A or B* features are extracted through channel information. The final features of experimental data are shown in Table. II.

b) For aperiodic messages: Whether aperiodic messages are abnormal is predicted by analyzing packets of the application layer. To be specific, we need to extract the characteristics of the telemetry data packets and generate labels for remote command packets individually, which constitute a training set for the aperiodic message detection method. And finally the training set is obtained. Taking the "temperature control" application of the thermal control subsystem as an example, the data pre-processing results are shown in Table. III.

C. Performance indicators for detection scheme

The performance indicators for evaluating our proposed intrusion detection method are accuracy rate (AR), detection

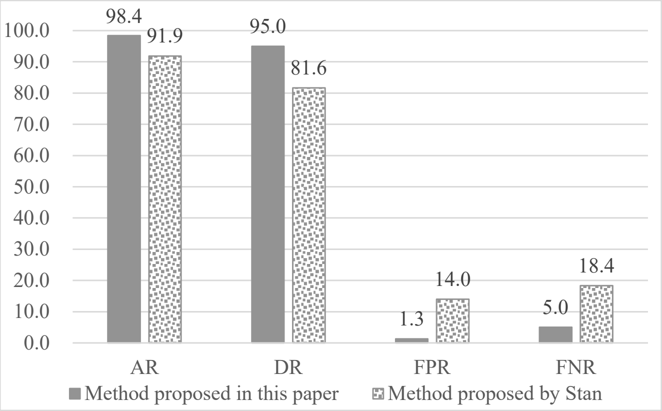


Fig. 8. Efficiency comparison between the two methods.

rate (DR), false positives rate (FPR), and false negatives rate (FNR). AR indicates that the anomaly detection method classifies the data correctly. DR indicates that the detected attack behaviors accounts for the percentage of the total number of attack behavior samples. High DR means that the method can detect more attacks. FPR indicates the percentage of the normal behaviors among the detected attacks and FNR indicates the percentage of the attacks among the detected normal behaviors.

D. Results and comparison

1) Comparison: Our baseline method is the work reported by Stan *et al.* [5]. They also process periodic messages and aperiodic messages separately, but both their algorithms are based on the Markov model.

We implemented these two schemes on the above simulation platform and used the same data set for comparison. The data set has 153,865 messages. As shown in Fig. 8, our method is more efficient than the method of Stan *et al.*, with 98.4% of AR, 95.0% of DR, 1.3% of FPR and 5.0% of FNR. Because the collected data is not enough to eliminate uncertainty, their method cannot judge certain ambiguous messages, and can only treat them as normal traffic, which leads to a low DR. Moreover, due to the limitations of the Markov model, the FPR and FNR of their scheme are both relatively high. But in the same experimental environment, our method addresses these problems through strict sequence comparison and decision tree detection, which greatly reduces FPR and FNR, and also improves AR and DR.

2) Discussion: The detection efficiency for different attacks is shown in Table. IV.

TABLE IV
THE DETECTION RESULTS OF DIFFERENT ATTACKS.

Attack type	AR (%)	DR (%)	FPR (%)	FNR (%)
DoS attacks	98.3	95.1	0.7	4.9
Forgery attacks	98.7	96.0	5.5	4.0
Tampering attacks	98.6	95.7	5.7	4.3
Replay attacks	98.4	92.1	3.6	7.9

For DoS attacks, because the attack messages are sent in large quantities within a short period of time, they can be easily identified as intruding traffic through the time signature field in the periodic message detection module. Forgery attacks are often difficult to meet the expected value of the message sequence specification or decision tree, so AR and DR are high, while FPR and FNR are low. Detection of tampering attacks also performed well. A tampering attack is usually sent as an aperiodic message, and the attack succeeds if and only if the system is in a state that can trigger this message. This is why our method can effectively detect tampering attacks.

Replay attacks can also be effectively detected in most cases. The replay of periodic messages violates the message sequence specification and decision trees. However, when replaying aperiodic messages, because the system is in a certain state to trigger aperiodic messages, it may conform to a decision tree, and therefore the FNR in our experiment is higher than other attacks.

V. Related works

Stan *et al.* [5] reported an attack on MIL-STD-1553B bus, and proposed a method based on Markov model to detect attacks (already described in the previous section). In contrast to the manual analysis by McGraw *et al.*, Stan *et al.* used a hardware and software combination test platform composed of the actual 1553 communication bus to perform bus attacks instead of manipulating data by exploiting various vulnerabilities in the protocol. However, the attack model established is very simple, which can only produce different attack effects by forgery of command words.

Losier *et al.* [18] proposed an intrusion detection scheme based on temporal features. In their work, RT response time, interval time between messages, periodic time, etc., are used as the judgment basis for intrusion. The intrusion detection scheme has been proved to be effective and feasible through experiments, but their method relies too much on temporal characteristics and ignores other characteristic information of messages, hence not good enough in accuracy.

He *et al.* [19] researched the security vulnerabilities of 1553B again, and analyzed various types of attacks, such as tampering attacks and DoS attacks, from the perspective of confidentiality, integrity and availability. The vulnerability of 1553B bus was demonstrated by simulation experiments. The authors introduced a more rigorous and detailed attack model, so as to require future researchers to provide solutions that can properly handle these attacks.

Another area of relevance to our work is the research on CAN bus security [20], [21]. CAN bus is a vehicle bus standard designed to enable micro-controllers and devices to communicate with each others' applications based on messages. Different types of attacks have proven that the bus lacks effective security mechanisms. Therefore, many researchers have proposed effective intrusion detection schemes. Song *et al.* [22] proposed a lightweight intrusion detection scheme that uses message intervals as the main feature. Kang *et al.* [23]

proposed the use of deep neural networks to improve the detection rate. In addition, some other research conducted intrusion detection by means of CAN bus message sequence [24] or construction of gradient boosting decision tree [25]. While the CAN bus and 1553B bus share some similarities, there are still many differences in protocol and physical hardware [26]–[28]. The most important one is that the 1553B protocol is more restrictive than the CAN bus protocol. The 1553B bus system must conform to the command response mechanism and communicate with terminals by the polling mechanism. From this perspective, the CAN bus is more powerful than the 1553B bus. Therefore, the existing CAN bus intrusion detection schemes cannot be applied to the 1553B bus directly.

VI. Conclusion and future work

This paper has proposed a method based on the message sequence and behavior rules to detect possible attacks in IESs. For periodic messages, we have used a smart method to extract the command word sequence for detection. For aperiodic messages, decision trees are constructed to predict system behavior. By comparing with with the previous work, we have demonstrated that our proposed method has better accuracy and detection rate, and the false positives and false negatives also remain low. This method is suitable for resource-constrained systems based on bus communication.

In our future work, we plan to deploy a specific application scenario of an IES (such as armored vehicles) to build a fully functional simulation test platform, to further analyze the cost of the proposed detection method in terms of memory and computing load. Then, according to the specific deployment environment and experimental results, we may optimize the method to more comprehensively protect the system based on bus communication.

Acknowledgment

This research is supported by the National Key R&D Program of China (2017YFB0802805 and 2017YFB0801701); the National Natural Science Foundation of China (Grants: U1936120 and U1636216); Joint Fund of the Ministry of Education of China for Equipment Preresearch (No. 6141A020333); the Shanghai Knowledge Service Platform for Trustworthy Internet of Things (No. ZF1213); the Fok Ying Tong Education Foundation of China (Grant 171058); and the Fundamental Research Funds for the Central Universities. Daojing He is the corresponding author.

References

[1] P. Marth, “Timed integrated electronics module (IEM),” in *Johns Hopkins APL Tech. Dig.*, vol. 24, Jan. 2003, pp. 194–200.

[2] M. E. Fraeman, “Research on digital simulation of satellite integrated electronic system,” in *Proc. 11th Annual AIAA/USU Conference on Small Satellites*, Sep. 1997, pp. SSC97–I–3.

[3] A. Gillen, J. Shelton, “Introduction of 3910 high speed data bus,” in *MILCOM 92 Conference Record*, USA, Oct. 1992, pp. 956–960.

[4] C. Deshu, W. Jixiang, “Guilin institute of optical communications; fiber-optic mechanization of an aircraft internal time division command/response multiplex data bus (come up for discussions),” *Optical Communication Technology Z*, vol. 1.

[5] O. Stan, Y. Elovici, A. Shabtai, G. Shugol, R. Tikochinski, S. Kur, “Protecting military avionics platforms from attacks on mil-std-1553 communication bus,” *arXiv preprint arXiv:1707.05032*, Jul. 2017.

[6] J. P. Farwell, R. Rohozinski, “Stuxnet and the future of cyber war,” *Survival*, vol. 53, no. 1, pp. 23–40, Jan. 2011.

[7] F. Schuster, T. Holz, “Towards reducing the attack surface of software backdoors,” in *Proc. of the 2013 ACM SIGSAC conference on Computer & communications security*, Berlin, Germany, Nov. 2013, pp. 851–862.

[8] J. Chen, C. Su, K. H. Yeh, M. Yung, “Special issue on advanced persistent threat,” *Future Generation Computer Systems*, vol. 79, no. 1, pp. 243–246, Feb. 2018.

[9] I. Ghafir, M. Hammoudeh, V. Prenosil, L. Han, R. Hegarty, K. Rabie *et al.*, “Detection of advanced persistent threat using machine-learning correlation analysis,” *Future Generation Computer Systems*, vol. 89, pp. 349–359, Dec. 2018.

[10] X. Hu, Q. Liang, “Research on digital simulation of satellite integrated electronic system,” in *11th IEEE International Conference on Control & Automation (ICCA)*, Taichung, Taiwan, Jun. 2014, pp. 112–116.

[11] M. Bowyer, L. Erup, H. P. Lexow, “Security in dvb-rcs2,” *International Journal of Satellite Communications and Networking*, vol. 31, no. 5, pp. 263–276, Sep. 2013.

[12] V. D. Gligor, “A note on denial-of-service in operating systems,” *IEEE Transactions on Software Engineering*, pp. 320–324, May 1984.

[13] B. D. Cullity, *MIL-STD-1553 designer’s guide*, 1998.

[14] M. Du, F. Li, G. Zheng, V. Srikumar, “Deeplog: Anomaly detection and diagnosis from system logs through deep learning,” in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, Texas, USA, Oct. 2017, p. 1285–1298.

[15] R. Chalapathya, S. Chawla, “Deep learning for anomaly detection: A survey,” *arXiv preprint arXiv:1901.03407*, Jan. 2019.

[16] A. Liaw, M. Wiener, “Classification and regression by randomforest,” *R. news*, vol. 2, no. 3, pp. 18–22, Dec. 2002.

[17] V. D. Gligor, “D. he, q. qiao, j. gao, s. chan, k. zheng, n. guizani,” *IEEE Network*, vol. 34, pp. 159 – 165, Aug. 2019.

[18] B. Losier, R. Smith and V. Roberge. (2019, Jan.) Design of a time-based intrusion detection algorithm for the mil-std-1553. Royal Military College of Canada, Kingston. Project number DTAES-8 2102. [Online]. Available: http://roberge.segfaults.net/joomla/files/publications/Project_Report_2102.pdf.

[19] D. He, X. Li, S. Chan, J. Gao, M. Guizani, “Security analysis of a space-based wireless network,” *IEEE Network*, pp. 36–43, Jan. 2019.

[20] P. Kleberger, T.Olovsson, E.Jonsson, “Security aspects of the in-vehicle network in the connected car,” in *Proc. 2011 IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, Jun. 2011, pp. 528–533.

[21] M. Wolf, A. Weimerskirch, C. Paar, “Security in automotive bus systems,” in *Workshop on Embedded Security in Cars*.

[22] H. M. Song, H. R. Kim, H. K. Kim, “Intrusion detection system based on the analysis of time intervals of can messages for in-vehicle network,” in *Proc. 2016 international conference on information networking (ICOIN)*, Kota Kinabalu, Malaysia.

[23] M. J. Kang, J. W. Kang, “Intrusion detection system using deep neural network for in-vehicle network security,” *PloS one*, vol. 11, no. 6, p. e0155781, Jun. 2016.

[24] M. Marchetti, D. Stabili, “Anomaly detection of can bus messages through analysis of id sequences,” in *Proc. 2017 IEEE Intelligent Vehicles Symposium (IV)*, Los Angeles, CA, USA.

[25] D. Tian, Y. Li, Y. Wang, X. Duan, C. Wang, W. Wang *et al.*, “An intrusion detection system based on machine learning for can-bus,” in *Proc. International Conference on Industrial Networks and Intelligent Systems*, Ho Chi Minh City, Vietnam.

[26] T. Hoppe, S. Kiltz, J. Dittmann, “Security threats to automotive can networks–practical examples and selected short-term countermeasures,” in *International Conference on Computer Safety, Reliability, and Security*, Newcastle upon Tyne, UK.

[27] R. M. Ishtiaq Roufa, H. Mustafaa, S. O. Travis Taylora, W. Xua, M. Gruteserb, W. Trappeb *et al.*, “Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study,” in *Proc. 19th USENIX Security Symposium*, Washington DC, USA.

[28] C. Miller, C. Valasek, “Remote exploitation of an unaltered passenger vehicle,” *Black Hat USA*, vol. 2015, p. 91, Aug. 2015.