An Algorithm for Detection of Malicious Messages on CAN Buses

Congli Ling
State Key Laboratory of Industrial Control Technology
Institute of Cyber-Systems and Control, Zhejiang University
Hangzhou, China, (86)18768188589,
lingdatabase@sina.com,

Dongqin Feng*
State Key Laboratory of Industrial Control Technology
Institute of Cyber-Systems and Control, Zhejiang University
Hangzhou,China,(86)13957166054
*dqfeng@iipc.zju.edu.cn

***Abstract*—Control systems are encountering increasing security threats; as one of the common systems, CAN control system is very facile to be attacked. Aiming at the improvement of CAN control system security, an algorithm for detection of malicious CAN messages is given, and it has been implemented in the CANoe simulation environment. The result shows that this algorithm has powerful detection function and valuable practice signification.**

Keywords- CAN bus; detection algorithm; malicious messages; CANoe

I. INTRODUCTION

Safety is the degree to which accidental harm is prevented, reduced, and properly reacted to; and security is the degree to which malicious harm is prevented, reduced, and properly reacted to [1], so this paper aims at the security of CAN control systems.

CAN control systems have long been used in areas like automobile, shipbuilding, industrial automation, aerospace, medical equipment, industrial equipment etc. Recently, ever-increasing general protocols, hardware and software have been the dominant parts of CAN control systems. Meanwhile, in present competitive markets, isolated control system networks are being inter-connected. Due to connecting these networks, and introducing IT components into the control systems, security problems arise. CAN protocol have been designed by Robert Bosch in 1986 for automotive applications as a method for enabling robust serial communication[2].The inherited vulnerabilities of such a protocol constitute the main threats of CAN control system. Some security problems in embedded systems are introduced[3]. Vulnerabilities of the automotive CAN control system are analyzed and verified by experiments[4].A simulation of sniffer and replay attack on CAN buses was carried out[5].The security of automobile systems was summarized[6]. It pointed out the security threats, especially DoS attack and information leakage in CAN systems, in FlexRay, LIN and MOST Systems, and put forward a few general defensive strategies. The security problems become sharply severe, but the counterparts are awfully scanty.

This paper will analyze the inherited security problems of CAN control systems from the perspective of technical features of CAN protocol and design an algorithm for detection of malicious CAN messages. In the end, the algorithm is implemented in CANoe simulation environment.

"Industrial Control System Network Security Technology Research " 863 project .

II. CAN PROTOCOL

A. Main Features of CAN Protocol

CAN protocol has defined the physical layer and data link layer of ISO/OSI reference model. It has two kinds format of message: one that has 11bits identifier is called standard message, and the other one that has 29bits identifier is called extended message. The difference between them is the length of arbitration field, as shown in Fig.1.

1) the Messages' Priority .

The sending sequence of messages on CAN bus depends on a message's identifier. When two or more nodes send messages, a message with smallest ID will get the priority. So it can meet different requirements of real-time.

2) CSMA/CD.

When two or more nodes send messages, nodes with low priority will automatically quit from the bus. Nodes with topmost priority can continue to send messages without being disturbed. Though under heavy network load, it can avoid network paralysis.

3) Nodes Identified by Node Number.

CAN nodes have no code information similar to "address". Adding or removing nodes will not influence the others on the bus.

4) Multi-master.

Besides the master-slave mode, CAN bus also has multi-master mode. When the bus is idle, all nodes can access it, and the first one will get the right of use. In case of collision, the right will depend on the value of messages' ID.

5) Effective Eerror Detection Mechanism.

An error detected by either the sender of a message, or receiver stations of the message, is signaled to the sender station. The sender then re-transmits the message. So the reliability of data transmission is greatly improved. Furthermore, it contains mechanisms for automatic fault localization including disconnection of the faulty controller.

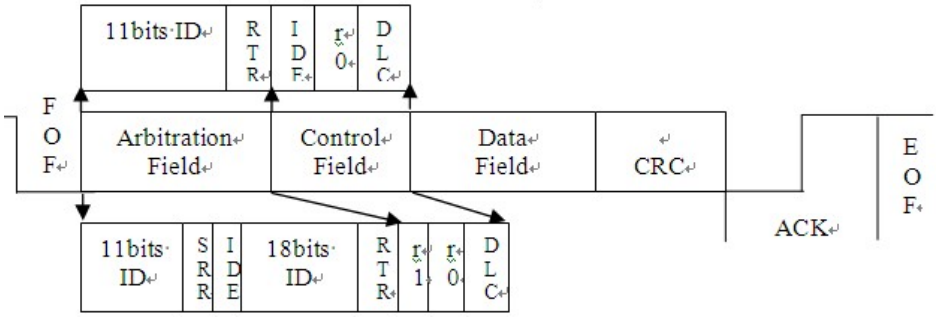


Figure 1.Standard and extended CAN Frame

B. *Vulnerabilities of CAN Protocol*

1) Jam the Bus or Modify Messages

From the perspective of technical features of CAN protocol, the transmission mechanism based on CSMA/CD on the one side guarantees that high priority messages can be transmitted preferentially, on the other side enables attacks that jam the communication channel. Constantly introduced topmost priority nonsense messages may be forwarded always first, even though they will be immediately discarded by the receiving controllers, and permanently prevent the transmission of all the other normal CAN messages. The attacker may modify the messages with malicious data to compromise the bus or sniffer data from the bus.

2) Well-directed Error Flags

Whenever a transmitter detects five consecutive bits of identical value in the bit stream to be transmitted, it automatically inserts a complementary bit in the actual transmitted bit stream. Error bit sequences destroy the bit stuffing rules between the Start of Frame and CRC delimiter or the fixed form from ACK field to the End of Frame field. Utilizing the CAN error detection mechanism, attackers can deliberately manufacture CAN bus error. The controllers will response this error without discrimination; hence the current transmission is interrupted. Malicious CAN messages may disturb normal operation of the bus or allow the disconnection of every single controller by posting several well-directed error flags.

III. THE DETECTION ALGORITHM

A. *Declaration of the Algorithm*

According to the content discussed above, the length of CAN ID regardless of 11bits or 29bits is far less than the maximum of 64bits data length. In addition, once the used IDs are configured, they will stay the same. This drastically contributes to detecting the malicious messages. Combined the ID transmitted on the bus with its uninterruptible occurrence frequency, the alarm threshold can be calculated. Meanwhile it avoids using large amounts of compute and storage resources, which are limited in the embedded CAN controllers.

B. *Detection Algorithm*

1) step1

- Take each ID and its threshold (ID_threshold) configured in the CAN control system to be detected, as well as the unknown ID's threshold(UN_threshold) as known input;
- Then set all counters and flags to 0;

2) Step2

- Listening the CAN bus to get the messages' IDs;

3) Step3

- Make judgment to the obtained ID;

4) Step3.1

- If the ID belongs to the input set, then increment the ID's counter (ID_counter)by one, set the ID's continuity flag(ID_flag), clear the unknown ID's counter(UN_counter) and its continuity flag(UN_flag);
- Do further judgment;
- If ID_counter exceeds ID_threshold and ID_flag equals 1, Then do alarm operation , clear the ID_counter and ID_flag;
- Else return to the step2;

5) Step3.2

- Else Increment the unknown ID's counter by one and set the unknown ID's flag, clear the known ID's counter and its continuity flag;
- Do further judgment;
- If UN_counter exceeds UN_threshold and UN_flag equals 1, Then do alarm operation , clear the UN_counter and UN_flag
- Else return to the step2.

IV. ALGORITHM AND DETECTION SYSTEM IMPLEMENTATION BASED ON CANOE

A. *Detection System Implementation*

CANoe is a special bus simulation tool, developed by the German VECTOR company. A series of CAN network system level of design, analysis and development tools it provides can help researchers to complete the whole system simulation work of bus network. CANoe can simultaneously

simulate network bus communication and nodes' various control functions. The detection system implemented by CANoe is shown in Fig.2. Device #D responses to the messages sent by the trigger, and periodically sends out heartbeat message. Malicious device Intruder sends out malicious messages. The ID of malicious messages can be detected by the Detector, then send out the ID and alarm flag within a message.

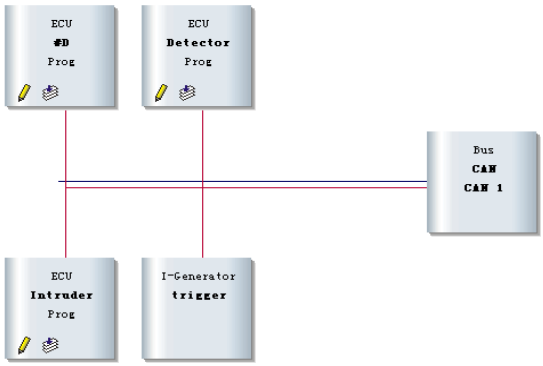


Figure 2. System implementation based on CANoe

B. *Messages Design*

Here to make a statement, all the messages in this paper just have research meaning but no practical significance.

1) *Trigger*

We suppose that the Trigger which may not exist in practice and is just used for research periodically send messages whose IDs are 0x180 and 0x181.

2) *#D*

0x280, 0x380 and 0x500，respectively, represents the messages ID of speed, frequency and heartbeat forwarded by #D .

3) *Intruder*

After 100ms delay from the beginning of system powered on, the designed malicious message 0x444 2C 2C will be forwarded by the Intruder. Then after 1000ms delay, it will consecutively forward malicious frequency message 0x380 FF FF FF twice.

4) *Detector*

When alarm threshold is exceeded, the Detector will

forward message 0x555 byte0 byte1 byte2 FF. Among them, byte0 and byte1 represent the ID of the detected malicious message; byte2 represents the alarm threshold of this ID; while FF represents the alarm flag.

C. *Algorithm Implementation*

Vector CAN Communication Application Programming Language (CAPL) is a C-like language, which is the programming language foundation of Vector CANoe and a rich, robust tool used to extend the power of CANoe beyond the tool's interfaces and to customize tool functionality based on CAN protocol to the user's requirements. The algorithm is just implemented by CAPL. Fig.3 shows the definitions of used variables and Fig.4 shows the flow chart.

```
variables
{
  int spd_counter=0;    // configured ID Speed counter
  int frq_counter=0;    // configured ID Frequency counter
  int htb_counter=0;    // configured ID Heartbeat counter
  int UN_counter=0;     //unknown ID counter
  int spd_flag=0;       // configured ID Speed
  int frq_flag=0;       // configured ID   Frequency continuity flag
  int htb_flag=0;       // configured ID   Heartbeat continuity flag
  int UN_flag=0;        // unknown   ID   continuity flag
  int t_spd=3;          //speed      threshold of 0x280
  int t_frq=2;          //frequency   threshold of 0x380
  int t_htb=1;          //heartbeat   threshold of0x500
  int t_UN=0;           // threshold of   unknown
}
```

Figure 3. Variables used in the implementation

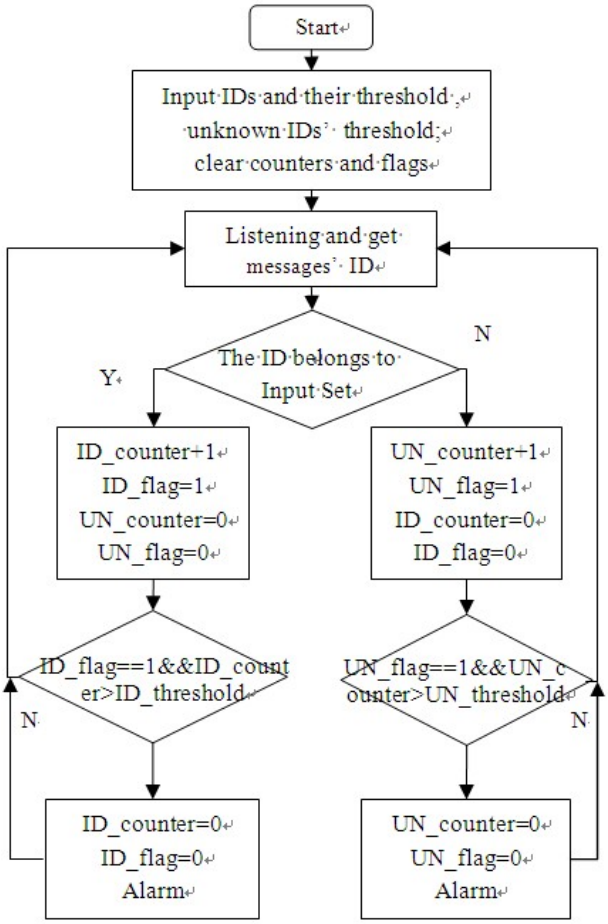
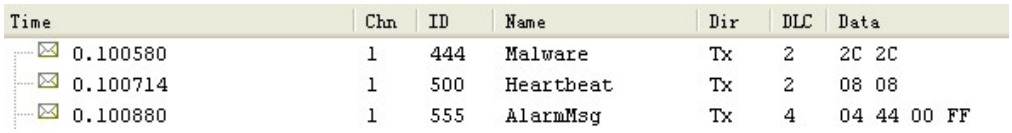
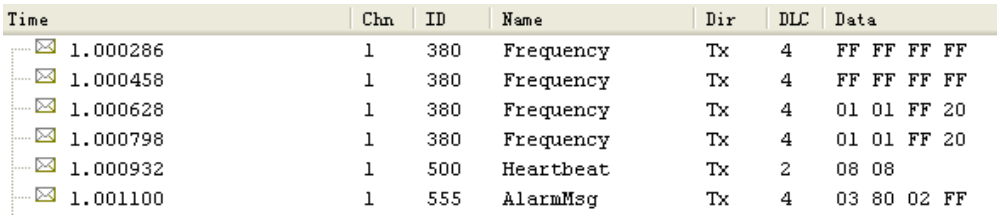


Figure 4. Flow chart of Algorithm Implementation



Time	Chn	ID	Name	Dir	DLC	Data
0.100580	1	444	Malware	Tx	2	2C 2C
0.100714	1	500	Heartbeat	Tx	2	08 08
0.100880	1	555	AlarmMsg	Tx	4	04 44 00 FF

Figure 5. Alarm triggered by malicious messgage ID 0x444



Time	Chn	ID	Name	Dir	DLC	Data
1.000286	1	380	Frequency	Tx	4	FF FF FF FF
1.000458	1	380	Frequency	Tx	4	FF FF FF FF
1.000628	1	380	Frequency	Tx	4	01 01 FF 20
1.000798	1	380	Frequency	Tx	4	01 01 FF 20
1.000932	1	500	Heartbeat	Tx	2	08 08
1.001100	1	555	AlarmMsg	Tx	4	03 80 02 FF

Figure 6. Alarm triggered by malicious fequency messgage ID 0x380

D. Test results

After the implementation of all modules, the whole system is tested. The results are shown in Fig.5 and Fig.6. After the Intruder sent out the malicious message 0x444 2C 2C or malicious frequency message 0x380 FF FF FF FF, and when the threshold was exceeded, the Detector alarmed.

V. CONCLUSION

This paper puts forward an algorithm for detection of CAN malicious messages based on CAN identifier for the problems that CAN control systems are very vulnerable when encounter malicious messages. By virtue of CAPL and CANoe simulation environment, the algorithm is verified, and the results show that both wrongful IDs and right IDs exceeding alarm threshold will trigger the alarm. The research in this paper supports the defense of CAN control systems significantly.

ACKNOWLEDGMENT

I am very grateful for my teacher and colleagues, Thanks for their guidance and help. I would like to express my gratitude to Zhejiang University for the support of the "Industrial Control System Network Security Technology Research" 863 project.

REFERENCES

[1] D.G. Firesmith, Common concepts underlying safety, security, and survivability engineering, Tech. Rep., CMU/SEI-2003-TN-033, Software Engineering Institute, Pittsburgh, PA, December 2003.

[2] Dongqin Feng, You Wang, Lei Xie. Industrial automation network. China Electric Power Press.2011.07

[3] P. Koopman. "Embedded system security". IEEE Computer, 37(7),pp:95-97, July 2004.

[4] K. Koscher, A. Czeskis, F. Roesner, et al. Experimental security analysis of a modern automobile. In D. Evans and G. Vigna, editors, IEEE Symposium on Security and Privacy. IEEE Computer Society, May 2010.

[5] Tobias Hoppe , Jana Dittman. Sniffing/Replay attacks on CAN Buses: a simulated attack on the electric window lift classified using an adapted CERT taxonomy. In Proceedings of the 2nd Workshop on Embedded Systems Security (WESS), Salzburg, Austria, 2007.

[6] Marko Wolf, André Weimerskirch, and Christof Paar. Security in automotive bus systems. In Workshop on Embedded IT-Security in Cars (escar), 2004.