A Fast and Secure Transmission Method Based on Optocoupler for Mobile Storage

Lu Zou[1], Dejun Zhang[1(✉)], Fazhi He[2], and Zhuyang Xie[1]

[1] College of Information and Engineering,
Sichuan Agricultural University, Yaan 625014, China
djz@sicau.edu.cn
[2] School of Computer Science, Wuhan University, Wuhan 430072, China

Abstract. This paper presents a one-way data transmission method in order to ensure the safety of data transmission from mobile storage to secure PC. First, an optocoupler is used to achieve the one-way transmission of physical channel, so that data can only be transmitted from mobile storage to secure PC, while the opposite direction is no physical channel. Then, a safe and reliable software system is designed which contains one-way communication protocol, fast CRC check method and packet retransmission algorithm together to ensure the safety of data transmission. After that, to obtain the maximum transmission rate, the frequency of data bus($slwr$) and the packet size(num) which effect on transmission rate are detailed analyzed. Experimental results show the proposed method is high-efficiency and safe.

Keywords: Data security · Mobile storage · Protocol · CRC check

1 Introduction

With the advent of global information and digital era, mobile storage technology has penetrated into every corner of the society, greatly improving the efficiency of people's work. However, the leakage events caused by the use of mobile storage [1] have posed a serious challenge to the national security system and corporate security system, which makes state secrets, financial information, business production and operation face a huge threat [2]. What's more, the recent outbreak of the Symantec event and PRISM [3] even exacerbate people's concerns about the safety of PC information. Although leakage events can be temporarily resolved by firewalls, proxy servers, intrusion detection or other security measures, it still cannot meet the requirements of government and sensitive departments for information security. In departments with high confidentiality requirements, the mobile storage even cannot be directly inserted to secure PC [4].

How to realize that not only can the data be transmitted from mobile storage to secure PC, but also prevent the data in the secure PC leaked to mobile storage. Hence, one-way transmission technology is one of the research hotspots in the field of information security in the world. To solve the leakage problems

B. Zou et al. (Eds.): ICPCSEE 2017, Part I, CCIS 727, pp. 251–261, 2017.
DOI: 10.1007/978-981-10-6385-5_22

and ensure the safety of secure PC, this paper presents a one-way transmission method based on optocoupler for mobile storage to access to secure PC. In this way, data from mobile storage can be transmitted into secure PC, but that in secure PC cannot be leaked out.

The proposed one-way transmission system is efficient and safe with double guarantee. To achieve the goal, at hardware-level, the characteristics of optocoupler is well used to achieve the one-way physical channel. At software-level, a unique program flow and a new communication protocol is designed, which guarantees device at the bottom only send data out, so as to ensure the transmission between mobile storage device and secure PC is one-way, safe and fast.

2 Previous Work

At present, the main researches about one-way transmission can be classified into: software solution [5] and hardware solution [6–9].

At software-level, Kang et al. proposed pump technology [5], which uses a reverse validation mechanism to prevent the flow of data from the outside to the inside. The technology is one-way, but the corresponding protocol is bi-way, once the protocol exists loopholes, it can be used to fetch the data in secure PC. Wan et al. [7] developed an optical shutter data protocol suitable for one-way communication, of which the protocol is similar to TCP, but there is no connection, no cache and no variable distribution consumption as TCP in the establishment of the connection requires for. And in this way, data sending is direct and more efficient. Zhao [8] analyzed the shortcomings of one-way transmission device, and put forward a more secure and reliable solution which is mainly realized by modifying UDP.

In recent years, with the rapid development of security attack technology, software-level isolation is far from meeting the requirements of confidential information security, and hardware-level isolation with corresponding device came into being. It's said that hardware-level isolation can guarantee the unilateralism of data transmission physically [9].

At hardware-level, Li [10] proposed a new secure data transmission method based on the traditional isolation card which is called SGAP. Xiao et al. [11] integrated USB controller and SPI controller of MCU to realize one-way transmission in hardware-layer and firmware layer. Wang and Meng [12] proposed a fiber-based one-way data transmission system of which the receiving end is connected to mobile storage, while the sending end is connected to PC. This system makes full use of a dedicated hardware device whose sending end and receiving end are composed of optical fiber connection.

Most of the transmission methods above are "blindly sending", which means the one is only for sending while the other is only for receiving, and whether the received data is incorrect or complete is not concerned. Because of there is no interactive control protocol, it always leads to error seriously. Furthermore, the use of dedicated device (e.g. fiber optic transceivers) is costly and usually limited by the lack of flexibility. Apart from this, infrared communication technology is

low transmission rate of which the highest transmission rate is 4 Mbps that is far from meeting the requirements of high-speed data transmission.

The proposed method in this paper is not only inexpensive, but also can ensure the safety of transmission between mobile storage and secure PC. In our method, an optocoupler is used as the one-way transmission medium, and then a safe transmission mechanism by software programming is designed to ensure the security and reliability of the data in the transmission channel.

3 Hardware Scheme and Overview

The one-way data transmission system aims at achieving a one-way data flow, specifically, it attempts to send data from the mobile storage to the secure PC. The overall framework of hardware component is shown in Fig. 1, which is divided into three parts: master controller, one-way physical channel and USB Device. (a) The master controller (ARM Cortex-A5) provides the USB host for reading the files of mobile storage. (b) The one-way physical channel is used to connect the master controller and USB Device. (c) The USB Device takes CY7C68013A as the core and it communicates with secure PC via USB bus.

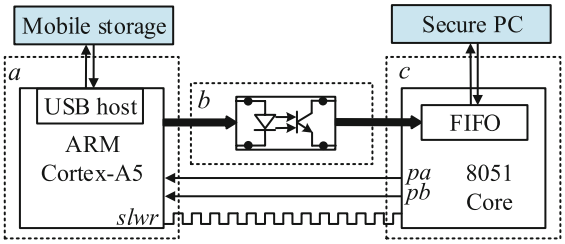

Fig. 1. The overall framework of hardware system.

From Fig. 1, both the USB host and the FIFO are set as input mode only which guarantees data security from hardware-level. In addition, an optocoupler is innovatively adopted in the one-way physical channel, which guarantees the data can only be transmitted from master controller to USB Device.

Data flow of the hardware-level is described as follows: (1) Since mobile storage has been inserted into USB host, the master controller reads basic information of the mobile storage. (2) Secure PC sends the transmission command to 8051 core via USB bus. (3) 8051 core parses the received command and triggers *pa*. (4) The master controller launches the one-way transmission procedure when it detects *pa* signal. (5) The one-way transmission procedure creates a transmission queue according to the file information of mobile storage. (6) The data packet is sent to optocoupler by master controller, and then passes the FIFO, eventually reaches to secure PC. (7) Secure PC parses the received packet, then recoveries the data packet.

4 One-Way Communication Protocol

Based on the hardware scheme, it is necessary to design a software system according to the one-way data transmission demand. Therefore, communication protocol for one-way transmission is the key and difficulty of this paper. UDP [13] based on the broadcast transmission, does not need to receive any confirmation message from the receiving end and supports for one-way transmission. However, UDP is generally used in the case where the amount of data for transmission is not large and the requirements of reliability is not high. Faced with large-capacity mobile storage, the problems of reliability must be solved. Therefore, this paper presents a new one-way communication protocol.

In the proposed protocol, the data packet required to send are defined as three types: *folder information packet*, *file information packet* and *file block packet*. In the one-way transmission process, each packet in the transmission channel is sent according to the order of *folder information packet*, *file information packet*, *file block packet 1*, *file block packet 2*, $\cdots$, *file block packet N*. The packet in the one-way transmission channel is composed of data block and CRC block.

4.1 Packet Format

Folder Information Packet. Folder in the mobile storage carries various attributes of the delivery folder, including packet type, operation permission, pathname, etc. And the receiving end creates the folder according to the structure of the *folder information packet*. The complete information as shown in Table 1.

Table 1. Structure of folder information packet

Variable name	Variable type	Field description
$PackType$	int	Packet type
$DesNum$	int	File index
$Mode$	int	Operation permission
$FoLength$	int	Pathname length
Folder[num]	unsigned char	Pathname

File Information Packet. *File information packet* is the starting packet which notifies the receiving end of the starting of a new transmission process and it carries various attributes (packet type, file size, access time, etc.). The receiving end checks the validity of the packet and recoveries the file according to the attributes. The complete information as shown in Table 2.

Table 2. Structure of file information packet

Variable name	Variable type	Field description
$PackType$	int	Packet type
$DesNum$	int	File index
$FileSize$	int	File size
$Access$	time_t	Access time
$Modify$	time_t	Modify time
$Mode$	int	Operation permission
$FiLength$	int	Filename length
File[num]	unsigned char	Filename

File Block Information Packet. When one file transmits, it is divided into several blocks by a predetermined value (num). As a result, a specific packet is responsible for sending a specific block of file. *File block information packet* is composed of packet type, block length, block contents these three parts. The complete information as shown in Table 3.

Table 3. Structure of file block information packet

Variable name	Variable type	Field description
$PackType$	int	Packet type
$DaLength$	int	Block length
Data[num]	unsigned char	Block contents

4.2 16-Bit CRC Check

Due to the use of block strategy, block packet may loss at the receiving end, and it's difficult to be reorganized into a complete original file. Thus, append check information to the end of folder information packet, file information data packet and file block information packet is necessary. The CRC check theory [14] is employed in this paper, according to the content of the packet (n-bit), a 16-bit check redundancy code is generated, which is appended to the packet structure to form a new packet ($n+16$ bit). The procedure of generating 16-bit check redundancy code is as follows.

The n-bit packet $D(X)$ to be sent is shifted 16-bit to the left and then divided by the polynomial $G(X)$, and the resulting remainder is the 16-bit check redundancy code. As shown in Eq. (1), which, $Q(X)$ is an integer.

$$\frac{D(X) \cdot 2^{16}}{G(X)} = Q(X) + \frac{R(X)}{G(X)} \tag{1}$$

The Mod-2 addition and subtraction used in Eq. (1) is bitwise with carry and borrow free. In fact, it is logical exclusive-OR operation. The multiplication and division operations conforms to the same rule. Thus, the polynomial of the 16-bit check redundancy code is shown below:

$$G(X) = X^{16} + X^{12} + X^5 + 1. \tag{2}$$

Ultimately, the received packet will be checked according to the rules of CRC at secure PC. Specifically, the received packet (including the packet information and CRC information) is divided by the polynomial. If the remainder is 0, it means there is no error, otherwise, there is error.

5 Structure of System

The one-way data transmission system is composed of bottom device and secure PC, as shown in Fig. 2, the bottom device is responsible for sending while the secure PC is responsible for receiving. These two components together to achieve the one-way transmission.

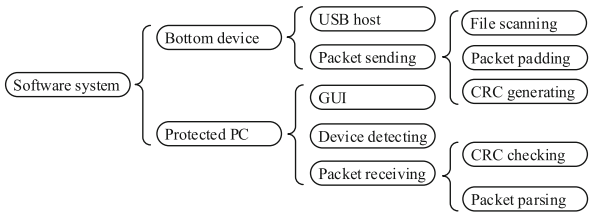

Fig. 2. The structure of software system.

The bottom device is divided into USB host module and packet sending module, and packet sending module consists of file scanning module, packet padding module, and CRC generating module. The USB host monitors the state of the USB host of the master controller in real-time, and when the state of the USB host is MOUNTED (see in Sect. 5.1), the packet sending module runs.

The secure PC is divided into GUI, device detecting module and packet receiving module, and packet receiving module consists of CRC checking module and packet parsing module. The device detecting module is for detecting whether the bottom device is connected or whether the optocoupler is normal.

5.1 USB Host

The status of USB host are described as three states: (a) no mobile storage inserted, denoted as DISKOUT, (b) mobile storage inserted but not mounted; in this situation, the contents of mobile storage cannot be read by OS(Ubuntu

14.04); denoted as IN, (c) mobile storage inserted and mounted; in this situation, the contents of mobile storage can be read; denoted as MOUNTED.

The USB host runs automatically with OS until the bottom device is powered off. Figure 3 shows the flow chart based on the three states of the USB host. The steps as follows: (1) Detecting the status of USB host. (2) If mobile storage is detected, mount it to the directory /mnt/usb in OS, otherwise, continue to detect. (3) Sending 2048 bytes of zero to test the physical channel and check whether there is hardware (optocoupler) failures, according to the status of *pa*. (4) Starting the one-way data sending process. (5) Unmount mobile storage, return to step (1).

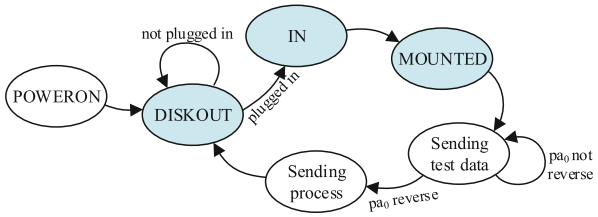

Fig. 3. The state machine of USB host.

5.2 Packet Sending

The packet sending module performs two deep recursive scans on the directory /mnt/usb. The first scan records the total number of leaf nodes whose type value is the initial value of $DesNum$ in the packets of $FOLDER$ or $FILE$. The second scan reads the leaf node and encapsulates it according to the three packet structures (*file information packet*, *folder information packet*, and *file block information packet*), and then appends the 16-bit CRC code to the packet to send. After each $FOLDE$R or $FILE$ has been transmitted, then $DesNum$ is decremented by one to zero. Since the communication protocol proposed in this paper is unreliable, a retransmission algorithm is designed to ensure the packet can be retransmitted in time. The retransmission algorithm is shown as below, when one packet is sent, the status of *pa* and *pb* is read and compared with *sa* and *sb*, respectively. If *pa* is reversed, it indicates that the secure PC has received correct packet. If *pb* is reversed, it indicates that the secure PC has received incorrect packet, and the retransmission algorithm is executing.

```
int sa=1,sb=1;
for(i=1;i<N;i++)
{
    SendingPacket(i);
    while(1)
    {
        if(pa!=sa)
```

```
            {
                sa=1-sa;
                SendingPacket(i);
            }
            elseif(pb!=sb)
            {
                sb=1-sb;
                break;
            }
        }
    }
```

5.3 Packet Receiving

The packet sending module provides a retransmission algorithm, therefore, the packet receiving module needs to provide a corresponding response algorithm. When the packet receiving module has received a packet, the CRC checking module executes: (1) packet errors, *pa* will be reversed; (2) packet corrects, the packet parsing module will be executed; (3) another packet is waiting to receive, *pb* will be reversed.

The steps of packet parsing as follows: (1) identify the type of the packet. (2) type value is $FILE$, parse the structure of *file information packet* and create a new file; type value is $FOLDER$, parse the structure of *folder information packet* and create a new folder; type value is $DATA$, parse the structure of *file block information packet* and write the binary stream into the created file. (3) since the file is completed recovered, the write stream is closed.

6 Implementation and Performance

In this section, to further analyze various characteristics of our method, we discussed the effect of *slwr* signal frequency and *num* value on the transmission rate. The program of bottom device was implemented in C and performed on Raspberry Pi 3 Model B, running Ubuntu14.04. The program of secure PC was implemented in C# 4.0 and performed on a computer with Inter (R) core (TM) i7-4470 (3.4 GHz) and 8 GB of Main Memory, running Win7.

The master controller (ARM Cortex-A53), as shown in Fig. 1, provides *slwr* signal which generates write clock to (falling-edge enable) the FIFO of USB Device [15]. Therefore, the *slwr* signal indirectly affects the transmission rate. In its best case, the shorter the period of *slwr* is, the faster the transmission rate is. However, in practical applications, when the main controller generates excessive frequency of GPIO, packet loss is seriously due to the problem of soft delay in Ubuntu, which affects the transmission rate of the system.

The relationship between *slwr* and transmission rate is shown in Fig. 4. The transmission rate of the system is improved with the increasing of the period of

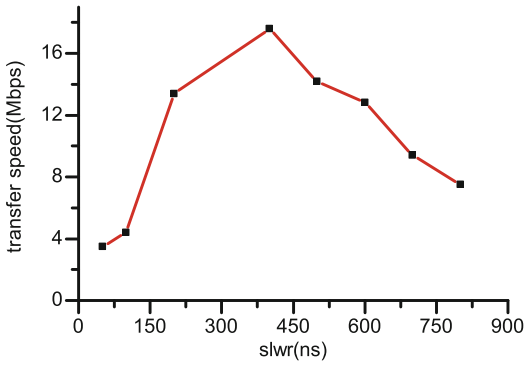

Fig. 4. The relationship between *slwr* and transmission rate.

slwr, and the maximum transmission rate is obtained when *slwr* is increased to 2.5 MHz. Moreover, with a higher period of *slwr*, the transmission rate is decreased.

Apart from the period of *slwr*, the size of the packet also plays a critical role in the transmission rate. In its best case, when there is no packet loss, *num* is not the bigger the better. For example, when mobile storage contains files of different sizes, set *num* as 100 Mbytes or larger, a large number of invalid data will be filled, which seriously affects the transmission rate of the system. Meanwhile, when there is packet loss, *num* is not the smaller the better. For example, set *num* as 256 bytes, as for the identical file, the rate of packet loss will be great with the increasing of *num*, which effectively reduce the rate of transmission loss of the whole system.

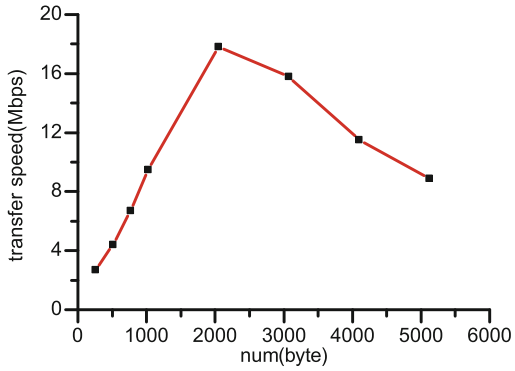

Fig. 5. The relationship between *num* and transmission rate.

The relationship between *num* and transmission rate (set *slwr* = 2.5 MHz) is shown in Fig. 5. The transmission rate of the system is improved with the

increasing of *num*, and the maximum transmission rate of up to 17.8 Mbps is obtained when *num* is set as 2048 bytes. Moreover, with a larger *num*, the transmission rate is decreased due to the waste of transmission channel.

7 Conclusions

In this paper, we proposed a one-way transmission system based on optocoupler as the physical channel. The novel aspects of our work can be summarized as follows: (1) We employed optocoupler to achieve a one-way transmission of physical channel. (2) According to the one-way environment, we construct a new communication protocol. (3) We designed an efficient CRC check method and a reliable retransmission algorithm. When packet transmission failures or packet losses, the algorithm ensures the reliability of transmission. (4) We improved the transmission rate up to 17.5 Mbps, basically meet the requirements of using mobile storage.

Acknowledgments. This work is supported by the National Science Foundation of China (Grant No. 61472289) and Hubei Province Science Foundation (Grant No. 2015CFB254).

References

1. Reddy, S.T., Lakshmi, D.L., Deepthi, C., et al.: USB_SEC: a secure application to manage removable media. In: 2016 10th International Conference on Intelligent Systems and Control, pp. 1–4. IEEE (2016)
2. Walker, S.J.: Big data: a revolution that will transform how we live, work, and think. Am. J. Epidemiol. **17**(9), 181–183 (2014)
3. Guardian, T.: NSA collecting phone records of millions of Verizon customers daily. Commun. ACM (2013)
4. Shah, N.N., Kumar, G.N., Raval, J.A.: Web based framework for data confidentiality in removable media ensuring safe cyber space. Int. J. Sci. Eng. Technol. Res. **4**(5) (2015)
5. Kang, M.H., Moskowitz, I.S.: A pump for rapid, reliable, secure communication. Proc. ACM Conf. Comput. Commun. Secur. **39**(7), 119–129 (1993)
6. Prost, W., Auer, U., Tegude, F.J., et al.: Tunnelling diode technology. In: International Symposium on Multiple-Valued Logic, pp. 49–58. IEEE (2001)
7. Wan, Y.L., Zhu, H.J., Liu, H.Z., et al.: Reliability of one-way transmission system base on optical shutter. Netinfo Secur. **12**, 25–27 (2010)
8. Zhao, B.: Study and design of safe one-way information transmission equipment. Comput. Appl. Softw. **27**(6), 98–99 (2010)
9. García-Dorado, J.L., Mata, F., Ramos, J., Santiago del Río, P.M., Moreno, V., Aracil, J.: High-performance network traffic processing systems using commodity hardware. In: Biersack, E., Callegari, C., Matijasevic, M. (eds.) Data Traffic Monitoring and Analysis. LNCS, vol. 7754, pp. 3–27. Springer, Heidelberg (2013). doi:10.1007/978-3-642-36784-7_1
10. Li, G.: Implementation of file transfer system based on physical isolation. Comput. Eng. Appl. **18**, 166–168 (2004)

11. Xiao, Y.J., Fang, Y., Zhou, A.M., et al.: Design of data unilateral transmission based on USB 2.0. J. Comput. Appl. **26**(6), 1481–1490 (2006)
12. Wang, H.Y., Meng, F.Y.: Design and implementation of one-way data transmission system based on optical fibre. Netinfo Secur. **3**(1), 68–72 (2011)
13. Singh, R., Tripathi, P., Singh, R.: A survey on TCP (transmission control protocol) and UDP (user datagram protocol) over AODV routing protocol. J. Appl. Physiol. **31**(1), 63–9 (2014)
14. El-Khamy, M., Lee, J., Kang, I.: Detection analysis of CRC-assisted decoding. IEEE Commun. Lett. **19**(3), 483–486 (2015)
15. Sowmya, M.S., Shwetha, H.N., Savitha, A.P.: USB interface with FPGA for data acquisition system using in X-ray applications. Int. J. Sci. Technol. **2**(5), 270 (2014)