The Design of a Novel Video Encryption System based on ARM Processor

Ke Xiao, Pai Pang, Jiali Cui

No.5, Jinyuanzhuang Road, Shijingshan District, Beijing, China
* . tel: (010)88801585, fax: 88801585, e-mail: zehan_xiao@163.com

Abstract **— Compared with computer software encryption systems, video encryption system using an ARM processor has the advantages of higher speed and greater security, and the potential for further application developments in reliable encryption devices. In order to implement video encryption using ARM system, this paper presents a design solution with ARM processor equipped with Android System and the Tongfang TF32A09 encryption chip. Through the USB interfaces, the unencrypted video files from ARM processor are transferred to TF32A09 encryption chip. After encryption on the chip, the encrypted video file is transferred back to ARM via USB interface. The proposed system is able to complete video transmission and encryption individually. The system was fully tested and the results show the new design can be easily applied in professional systems that require fast and secure encryption.**

Keywords - ARM; Linux; TF32A09 encryption chip; video encryption

I. INTRODUCTION

With the increased need of higher video security and encryption speed, the traditional software video encryption can not meet the demands from modern applications. Therefore, in order to seek a safer, faster encryption, the entire industry has shown a trend of hardware encryption. Hardware encryption is characterized by high security and high speed.

In recent years, video hardware encryption is mainly used for commercial private videos. For private business video, in order to prevent the video from being accessed and used by unauthorized people, higher security is necessary, and at the same time we need to ensure high-speed encryption. Hardware encryption just has the characteristics of high-security and high-speed, so it can provide a safer encryption solution in commercial private video areas.

Hardware video encryption system based on ARM has lots of advantages including fast encryption and high security. Hardware video encryption has gradually replaced software encryption and become the mainstream of security encryption. Encryption systems with higher security are needed in today's society, and most of them use hardware encryption. Currently, ARM9 processor is primarily used in hardware video encryption systems. However, ARM9 core processor’s performance in practical application can not meet the existing need for speed anymore. Therefore, in this paper, we use ARM11 as a high-performance processor to interact with TF32A09 development board.

II. SYSTEM ARCHITECTURE DESIGN

The proposed system mainly consists of ARM development board, USB transmission module, TF32A09 development board encryption, and computer decryption module. The system’s main functions are implemented by the following steps. Firstly video files are imported to ARM development board to achieve video loading. The unencrypted video files in ARM development board will be transferred to TF32A09 development board via USB module to achieve video transmission between ARM development board and the TF32A09 development board. Then the video files received by TF32A09 development board will be encrypted and then transferred back to ARM development board. Finally the encrypted video files are imported to computer, decrypted, verified and played on computers.

III. SYSTEM HARDWARE AND SOFTWARE DESIGN

A. Hardware Structure

The hardware structure of video encryption system based on ARM mainly consists of ARM development board, USB transmission module and TF32A09 development boards. The system hardware block diagram is shown in Figure 1.

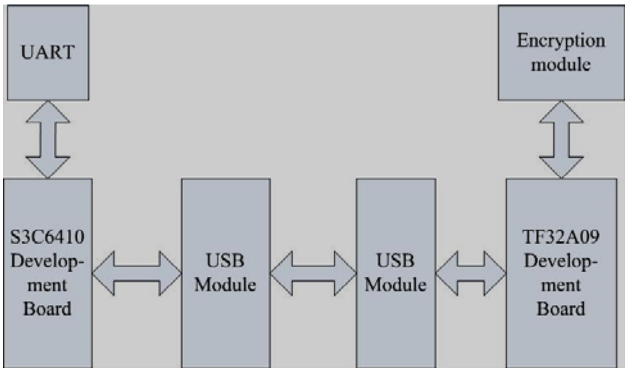

Figure 1. System hardware block diagram

S3C6410 development board consists of S3C6410 microprocessor, NAND Flash, SDRAM and USB modules, which are respectively used for curing stored program, programs running on the system and video files' transmission through USB.

TF32A09 development Board consists of 32-bit CPU core (C • CORE C320), the security module and USB modules which are respectively used for data encryption, data decryption, and video files' transmission through USB.

a . S3C6410 core processor

Samsung's S3C6410 is a 16/32 bit RISC microprocessor which uses ARM1176JZF-S as the core. This processor's core clock frequency can be up to 667 MHz, and it has NAND Flash, SDRAM controller, USB2.0 OTG (up to 480 MB/S) and other modules.

b. TF32A09 core processor

TF32A09 development board uses a 32 bit RSIC architecture CPU from Tongfang company as the core (C • CORE C320). Its maximum operating clock frequency is 100MHz. It has DES algorithm module which supports DES encryption/decryption and 64-bit DES encryption key. With a 80MHz clock, the encryption/decryption speed can be up to 33.9097 MB/s.

c. USB2.0 OTG

USB OTG (USB On-The-Go), a technology developed in recent years, is mainly used for the connection and data exchange between different types of devices and mobile devices. IEEE1394 and USB are the two main standards for such transfers. Both standards provide Plug and Play, hot-swap function, and they support power supply for outside devices and can be used to connect multiple devices. The launch of USB2.0 standard makes USB transfer speed of reach up to 480Mbps. And the launch of USB OTG technology can achieve data transmission between devices without host.

B. System Software Architecture

The software part of the system is mainly composed of two parts: embedded Linux operating system and the TF32A09 embedded chip development board.

a. BootLoader, Linux kernel and root file system migration

BootLoader is the boot program for the system. U-Boot-1.1.6 version provided by the development board company is used in the proposed system. Linux-2.6.28 is chosen as Linux kernel, and via TFTP, the BootLoader is downloaded from the PC Linux to NAND Flash on the development board.

This system design uses NFS root file system, which launchs the root file system stored in the NFS server using NFS. BusyBox-1.20.1 generates roofs directory. Commands under bin, sbin and usr, which are under roofs directory, and the initiator Linux C are mainly used in this process.

b. S3C6410 and TF32A09 use of USB-OTG

USB OTG (USB On-The-Go) is a technology developed in recent years. USB OTG was published by the USB Implementers Forum on December 18th 2011, and it is mainly used in connection and data exchange between different devices and mobile devices. The development of USB technology makes it possible that PC and various peripheral devices can be connected through a simple way at reasonable manufacturing costs. All these applications we mentioned above can be implemented by connecting all the devices, as PC's peripherals, through USB bus. The devices exchange data with each other under the control of PC. But, without PC, this convenient way of data exchange, can not work because these devices are unable to operate using USB port without a acting as PC Host. On-The-Go technology, namely OTG, is to achieve data transfer between devices in the absence of Host.

In summary, USB-OTG can freely choose which side is Host, which side is Drivice, so it is necessary to write the appropriate driver at the time of USB-OTG installation. Here, we will set USB-OTG of S3C6410 development board as Host, and set USB-OTG of TF32A09 development board as Drivice.

c. TF32A09 embedded chip encryption development

The main functions of this module are initializing USBC1, receiving video files from the USB module of ARM development board, setting the configuration control status registers of TF32A09 development board to enable encryption mode, loading the key to DAEKIN1R register, and loading video files to DAEDINR registers until the encryption completes.

C. Application Program Design

In order to do embedded software development, we need to build a cross-compiler environment in the host PC and use ARM-Linux-gcc-4.6.3 cross compiler to generate S3C6410 executable files. Application program mainly consists of video files' USB transfers and the usage of USB-OTG. Video encryption process is divided into three parts. The first part is the USB data transfer on S3C6410 development board. The second part is the USB receiver and the encryption part on TF32A09 development board. The third part is sending the files back to S3C6410 development board through the USB interface after the encryption is completed on TF32A09 board.

a. USB transmission of video data

Current methods of data transmission are mainly IIC, UART, USB1.0, USB2.0.

TYPE	RATE
IIC	100kbps
UART	1Mbps
USB1.0	12Mbps
USB2.0	480Mbps

All these four methods can be used as transmission interface, but in order to meet the real-time speed requirement, we select USB2.0 interface.

b. Video Encryption program of S3C6410 development board based on USB-OTG interface

USB-OTG interface is a system interface that Linux kernel provides for USB transmission. In order to transfer video files at high speed, the system uses USB2.0 interface whose rate can reach 480Mbps. USB transmission section of S3C6410 development board is shown in Figure 2.

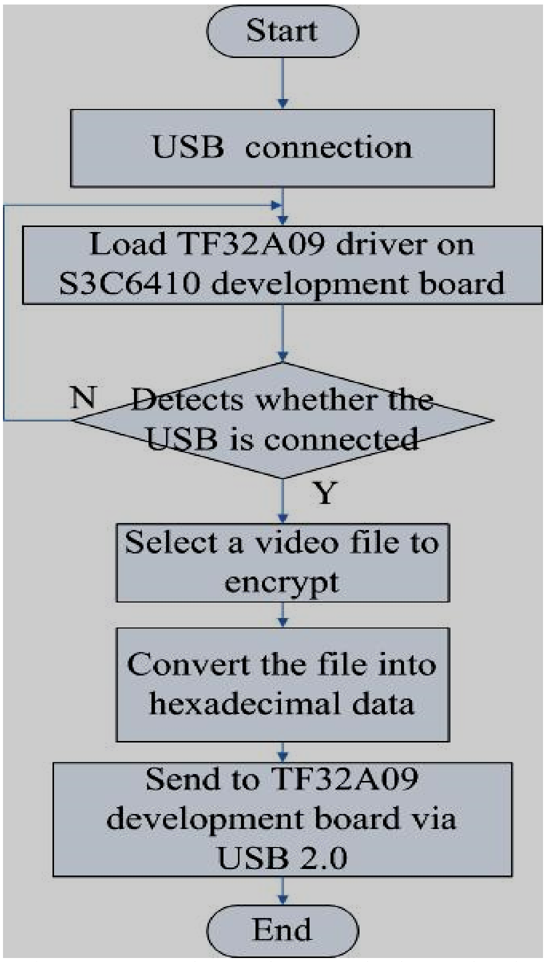
Start
USB connection
Load TF32A09 driver on S3C6410 development board
N
Detects whether the USB is connected
Y
Select a video file to encrypt
Convert the file into hexadecimal data
Send to TF32A09 development board via USB 2.0
End

Figure 2. USB transfer part of S3C6410 development board.

(1)TF32A09 USB driver under Linux has been previously written. It is automatically loaded to S3C6410 development board.

(2)Call process_dev (SCSI_USB_DEV * dev) to detect whether USB is mounted or not.

(3) Call open (const char * pathname, int flags) to open the video file.

(4) Call a2x (const char c) to convert Strings in the file to hexadecimal data.

(5) Use USB as file to read and write:Open mounted USB, call open ("/ dev / ttyS0", O_RDWR).Write in hexadecimal data to USB, call write (fd, buffer, Length).

c. video encryption program based on TF32A09 development board

TF32A09 Development Board contains a DES algorithm module, supporting DES encryption/decryption and 64-bit DES encryption key. With a 80MHz clock, the encryption/decryption speed can reach up to 33.9097MB/s. It supports USB2.0 OTG (up to 480 MB/S). The flow chart of USB receiver and encryption part of TF32A09 development board is shown in Figure 3).

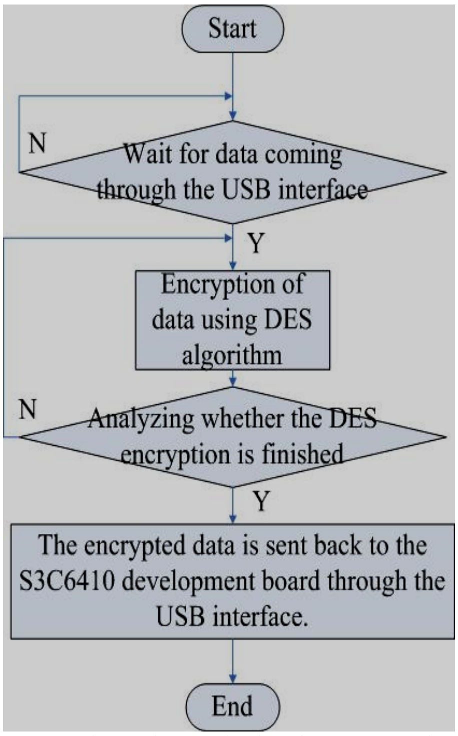
Start
N
Wait for data coming through the USB interface
Y
Encryption of data using DES algorithm
N
Analyzing whether the DES encryption is finished
Y
The encrypted data is sent back to the S3C6410 development board through the USB interface.
End

Figure 3. USB receiver and encryption based on TF32A09 development board.

(1) Firstly, determine whether there is data is sent over via USB and or received.

Call if((g_byUsbMsgFlags&EP2_RX_INTR)==EP2_RX_INTR) ReceiveUsbPack((UINT8 *)g_abyUsbIntBuf);

(2) Encrypt hexadecimal data using DES algorithm

Call while((TFCS_DAECSR_B & DAECSR_BUSY) == 0x01);

(3)Determine if the string pointer is '\0' or not. If yes, it means DES encryption is completed.

(4)Send the encrypted video file to S3C6410 development board.

CallSendUsbPack((UINT8)g_abyUsbIntBuf,sizeof(g_abyUsbIntBuf));

d. Video reception and encryption program based on S3C6410 development board

Video files are encrypted on TF32A09 development board and then sent back through the USB interface to S3C6410 development board. The flowchart of this process is shown in Figure 4.

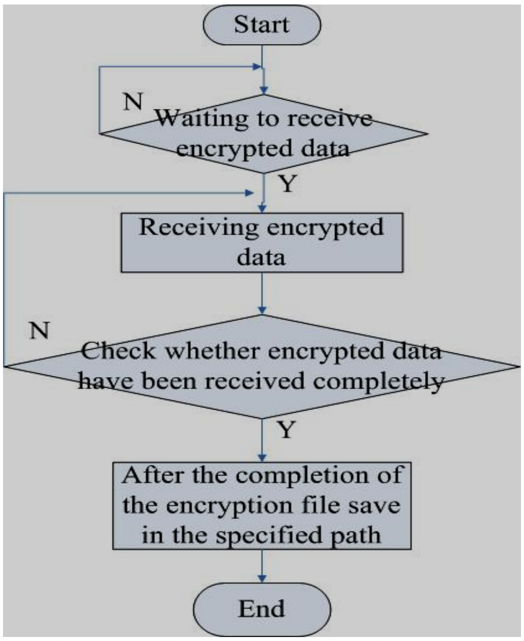
Start
N
Waiting to receive encrypted data
Y
Receiving encrypted data
N
Check whether encrypted data have been received completely
Y
After the completion of the encryption file save in the specified path
End

Figure 4. Video reception and encryption section.

(1) Call if (read (fd, buff, Len)> 0) detect whether there is data received.

(2) Call while ((nread = read (fd, buff, 512))> 0) to receive the data and determine whether the data reception has finished.

(3) Call write (int fd, const void * buf, size_t count) function to store the completely encrypted video files in specified location.

D. Video decryption and play

A corresponding key has been set up in the system during DES encryption. Then after the encrypted video files are stored to the specified location, this key can be used to decrypt and play the video.

a. Decryption of video under the Linux

In Linux system, we use its built-in GnuPG software for video DES decryption.

GnuPG stands for GNU Privacy Guard, which is usually called GPG. It combines a set of encryption software. It was written in the C programming language in the GNU project.

GPG uses command "$ gpg + target absolute path" to decrypt files.

b. Video display under Linux

In Linux system, we often use VLC multimedia player. VLC multimedia player, which was initially called Video LAN client, is the multimedia player in VideoLAN project. It can support numerous voice and video decoders as well as various file formats. Meanwhile, the displaying of DVD, VCD and different types of Real Time Streaming protocols can be supported too. Moreover, it can be used as unicast or multicast's streaming server with high-speed Internet connection in IPv4 and IPv6. The VLC multimedia player combines the decoder from FFmpeng and libdvdcss program library, which makes it gain more functionality such as multimedia files displaying and DVD video disks encryption.

Cross-platform is one of the characteristics of VLC multimedia player. It has various versions such as Linux, Microsoft Windows, Mac OS X, BeOS, BSD, Pocket PC, Solaris and so on.

IV. RESEARCH RESULTS AND PERFORMANCE ANALYSIS

This design uses USB2.0 port to transfer data between two development boards, constructing a system that is shown in the following Figure 5.

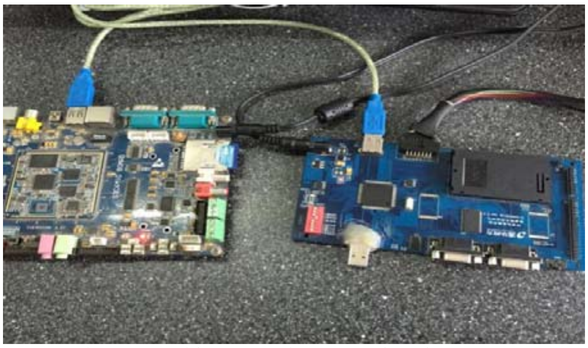

Figure 5. Video encryption system based on ARM.

On the left side of the picture is S3C6410 development board, and the right side is encryption development board. USB is used to transfer video data between these two boards.

The size of the video used during this test is 130MB. We test DES encryption operation on this video 3 times and the results show that the encryption time for each test is 4 seconds and the average encryption speed can reach 30MB/s.

Playing the video after decryption is shown in Figure 6.

Figure 6. Video Encryption Test Results.

V. CONCLUSIONS

Based on S3C6410 development board, TF32A09 development board, USB2.0 interfaces, embedded LINUX operating system, the proposed system achieves video USB transfer and video encryption. The experimental test results show that the entire system has the advantages of high-speed encryption, high-speed and stable transmission. The system can be extended to many areas, such as enterprise-class real-time video encryption, military video real-time encryption and so on, and it has broad application prospects.

REFERENCES

[1] Zheyu Zhang. Design and Implementation of Video Surveillance System based on ARM-Linux. [Beijing Jiaotong University] .2009

[2] Min Zhao. Research and application based on S3C2440 embedded Linux systems. [Xiangtan Science] .2009

[3] Zhenquan Xu, Zhiyun Yang, Wei Li, Lin Chen. Development Status and Prospects about Digital video encryption technology. Learned journal of Wuhan University. 2005.30(7):570-574

[4] Shiguo Lian, Zhongxin Li, Zhiquan Wang. The performance comparison of several typical video encryption algorithm. Learned journal of China Image and Graphics. 2004.4(9):485-490

[5] Mingjun Han, Wei Wang. A program of data encryption on remote monitoring. Technical application of 200 cases about Fieldbus. 2005.21(7):69-71

[6] Fan Yang. Research and design of remote video surveillance system based on secure and real-time. [Jilin University]. 2009

[7] Liming Wang, Shuangjiao Chen. Development and practice based on ARM9 Embedded Systems. Beijing: Beihang University press. 2008:4-16

[8] Lingxiang Zheng. Design and application development based on Embedded Systems. Beijing: Beihang University press. 2006:20-80

[9] Inter.Intel PXA270 Processor.DATA SHEET.2005

[10] Jikun Sun. The detail of development technology based on Embedded Linux system.

[11] Alan Cox,Video4Linux Programming,2000

[12] Kun Song. The treasured book of Visual C++ Video Technology Programme. Beijing: The People's Posts and Telecommunications Press. 2008:10~20

[13] JAYANT N S, NOLL P. Digital coding of waveforms[M]. Englewood Cliffs, NJ: Prentice-Hall,1984

[14] I-M Pao, M T Sun.Modeling DCT Coefficient for Fast Video Encoding．IEEE Tran．CSVT,1999，9:608-616

[15] Wrote by (US)Andrew N. Sloss, translated by Jianhua Shen. Software design and optimization based on ARM Embedded Systems. Beijing: Beihang University press. 2005: 1-207

[16] Quangang Wen. The design of assembly language programming based on ARM architecture. Beijing: Beihang University press. 2007: 30-170

[17] Xiaoxin Wang, Yifang Weng, Rong Zheng. The research of network video encryption algorithm based on Logistic chaotic system. Learned journal of Beijing Technology and Business University. 2009,2:42-46

[18] Fridrich J. Symmetric Ciphers Based On Two-Dimcmional Chaotic Maps．International Journal of Bifureafion and Chaos，1998，8(6)：1259~1284

[19] IETF．A Transport Protocol for Real-Time Applications(RFC3550)，2003

[20] IETE RTP Payload Format for MPEG-4 Audio／Visual Streams(RFC3016)，2000

[21] Wrote by (CA)Tom St Denis, (UK)Simon Johnson, translated by Xiaobin Fu. Programmer's Cryptography. Beijing: China Machine PRESS. 2007:10-300

[22] Yunxiang Zhu, Ping Hu. Proficient in C programming language and project practice under UNIX. Beijing: Electronic Industry Press, 2007:195-372

[23] BenQ．M22／M23／M23G AT Command User Guide．2006