DESIGN AND IMPLEMENTATION OF TPM SUP320

Jiangchun Ren, Kui Dai, Zhiying Wang, Xuemi Zhao and Yuanman Tong
School of Computer, National University of Defense Technology, Changsha, Hunan, P.R.China, 410073

Abstract: Security of computer in network is becoming more and more challengeable. The traditional way of applying a common smart card to application can not meet the requirement of high degree of security in critical systems. Trust Computing Group (TCG) drafts out specifications on trust computing platform, which have been acknowledged by specialists in this field. Following these specifications, we designed and implemented a chip named SUP320 with SOC technology. This paper gives the chip's hardware architecture, firmware modules and method for low power. Performance of SUP320 is tested in the end. We find that SUP320 is better than traditional smart cards in both security and efficiency.

Key words: TCPA(TCG); TPM; SUP320; SOC; low power; smart card; keys management.

1. INTRODUCTION

Computer systems in network are often attacked by viruses and trojan horses, the basal platform can not build up a trust and secure environment for applications on it. Most systems resist hacker's attack by technologies such as digital certificates and public key infrastructure to authenticate participants and provide cryptographic keys. But the arithmetic of cryptography reduces system's efficiency heavily. A smart card then is used to accelerate the arithmetic by hardware component.

The smart card surely increases the system security to some extent, but it can not meet the requirement of high degree of security. For it has some obvious disadvantages: firstly, the communication protocol between the card and the host is too easy that users' privacy may be hijacked in the middle; Secondly, the card's processor and memory are isolated and just connected by wires, the data in memory is possibly decrypted by hacker who can steal the card; and thirdly, the mode of smart card is single, administrator can not apply different security policies to different applications. Just for these reasons, Compaq, IBM, Intel, HP and Microsoft launched and formed Trust Computing Platform Alliance (TCPA, renamed TCG later), TCG drafts out a specification on the subsystem for security in universal platform[1]. The specification suggests a Trust Computer Base (TCB) should be used in the platform and the whole system's security infrastructure should be built up on it. The TCB combines a highly secure chip with its outside circuit. The secure chip is often named Trust Platform Module (TPM)[1][2][3].

In this year, we designed and implemented a TPM chip named SUP320, and designed an architecture of subsystem which can be embedded in the common computers. Platforms with this subsystem can get assistance for security in all kinds of levels: hardware, OS kernel and application[5].

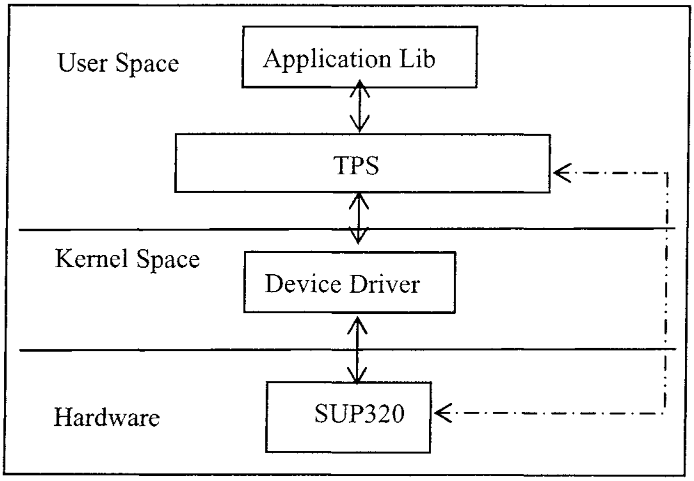

Figure 1. Framework of Secure Subsystem

The subsystem is composed of physical chip SUP320 including outside circuitry, device driver, Trust Platform Service (TPS) and libraries assisting for applications. Among those four levers, SUP320, a system on chip is the core of the subsystem. SUP320 has two kinds of important functions: accelerating security arithmetic and intercommunicating with host according to a robust protocol (explained in the broken line)[6][7]. This paper firstly presents the hardware architecture of SUP320; then describes the firmware modules in detail; following that, introduces the method for low power; Finally, tests the chip's performance.

2. HARDWARE ARCHITECTURE

The SUP320 was implemented by 0.18um 1P6M CMOS technology, Its die size is 4.9×4.9 mm^2, The cost of power is about 0.6W. Its hardware architecture is presented in figure 2.

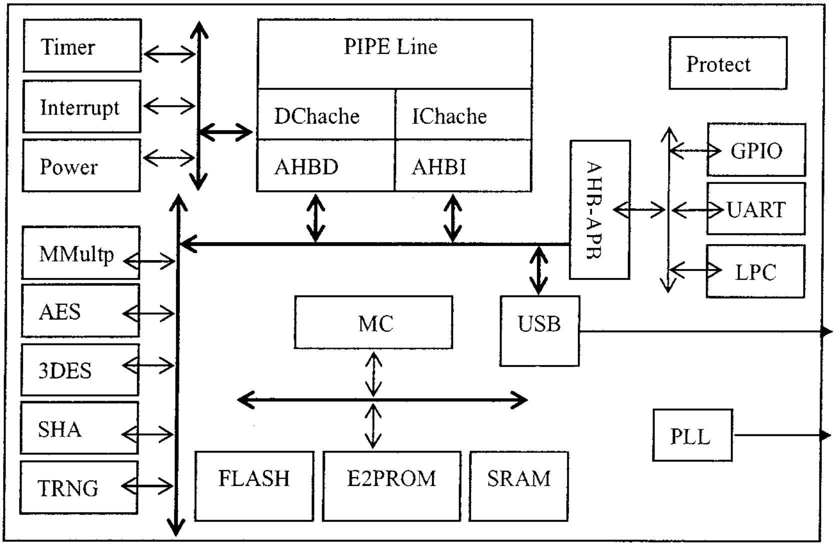

Figure 2. Hardware Architecture of SUP320

The core of SUP320 is a 32 bit RISC processor named "TengYue-1", which is designed by ourselves. The processor works well under frequency of 200 MHz, with a five-lever pipeline, an independent data cache and an instruction cache. Cache sizes are both 2K bytes.

MMultp, AES, DES and SHA are co-processors, which accelerate arithmetic of RSA/ECC, AES, DES and SHA. Because the operations of modular multiplication and modular power are frequently used in public-key arithmetic of RSA and ECC, we design a co-processor "MMultp" to speedup operation of modular multiplication, then speedup operation of modular power based on that. It is easy to realize the arithmetic of RSA (the length of key can be 512bit, 1024bit and 2048bit) and ECC (the length of key can be 160 bit and 192 bit) by different procedures[8][9][10].

TRNG is a true random numbers generator which produces numbers through noises in physical matter. These true random numbers are used in the module of communication protocol and public-key arithmetic to get big modular number pairs.

There are three kinds of memories on the chip: FLASH, EEPROM and SRAM. The size of flash is 128K bytes, in which the whole firmware is stored. The size of EEPROM is 128K bytes too, in which all kinds of keys and privacy data are stored. The SRAM's size is 16K bytes. It acts as a work room for the system on chip. These memories are all managed by a memory controller.

Peripheral interfaces include UART, GPIO, USB and LPC (Low Pin Count). LPC is designed according to TCPA specification, which can be connected with Intel CPU[4]. Other interfaces are designed to improve the flexibility of this chip so that SUP320 can also be used in other devices such as USB-KEY, secure disk and etc.

3. FIRMWARE

The firmware of SUP320 plays a critical role in the SOC. It is composed of many modules such as initialization, self test, interfaces abstract, arithmetic accelerating, community protocol, keys management, session management and method for low power. The ability of real-time and high efficiency always affects the system on chip to great extent. There are two approaches to this problem: one is to clip real-time OS such as RtLinux; another is to code sub-procedure for each module and set them up. The first method has an advantage that programmer can use all kinds of functions supported by OS and ignore the work of task scheduling, but the code size of OS is often too big. The second method has advantage inversely. Our chip would be used in a complex platform, efficiency is very important to the host. And the size of memory on chip is limited. For these reasons, we adopt the second method to organize the firmware. The firmware's architecture is presented in figure 3.

The main job of SUP320 is to wait for commands and do them, so the module of communication protocol is the schedule center of firmware. Command issued by outer entity enters the chip through the module of interfaces abstract, which is aroused by interrupts from peripheral interfaces. The module of communication protocol checks the integrity of command and explains it according to TCPA specification. After that, it arouses other modules to run. The result of operation is also sealed and sent out by it. In the following section, we give more details of some critical modules.

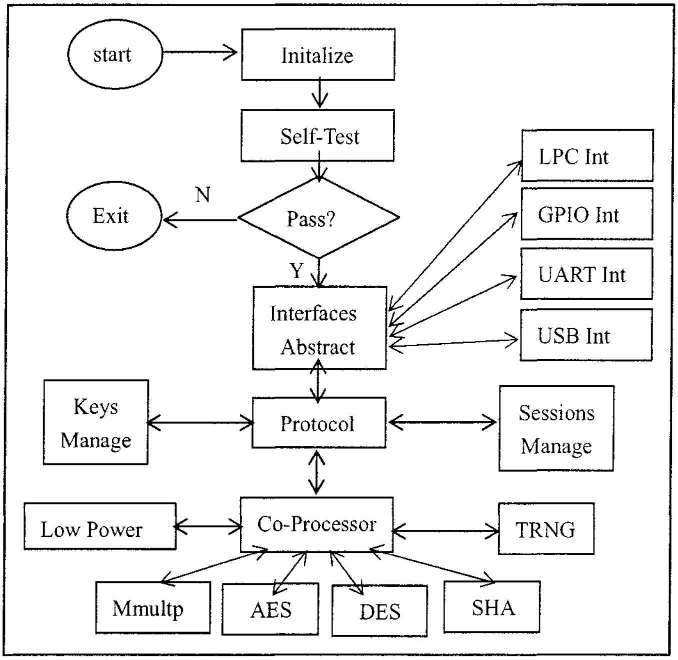

Figure 3. Firmware Modules

3.1 Interfaces Abstract

In operating system, all kinds of interfaces to peripheral devices are described abstractly as file interface. Sending and receiving data from device by applications are just like writing and reading data from a special file. The module of interfaces abstract uses a kind of data structure to describe communicating data in peripheral interfaces. The data structure is presented in figure 4.

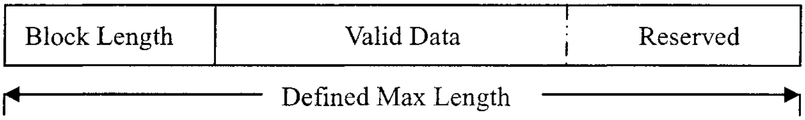

Figure 4. Data Structure Described In Interface Abstract Module

SUP320 has lots of peripheral interfaces, so it can be used in devices such as USB-KEY and secure disk. The module of interfaces abstract lets system on chip pay more attention to data processing but not data receiving or sending. Of course, the chip will use only one interface in a practical device, other interfaces is disabled. However, our method of abstracting interfaces could let the firmware reliable in a same copy for different cases.

3.2 Keys Management

The module of keys management is the securest module in SOC. Keys can been classified into two groups by their functions: stored keys and signing keys. They can also been classified by their capability in migration into two groups: migratable keys and non-migratable keys. Keys have the following attributes:

1. Some key is bound to a chip or a platform;
2. Each key has a multi-lever access control, one key may not be open to all processors in the platform;
3. All keys are managed in hiberarchy. Each key has a blob and naturally leads to a tree. The root of tree is the “Storage Root Key”(SRK) which is generated inside the TPM and is non-migratable.

Table 1. Data Structure of Key Blob

ID	ClassID	Content	Authorize	Parent ID	Next ID

To describe key blob in the key tree, a data structure presented in table 1 is adopted. All keys are managed in a thread tree (figure 5).

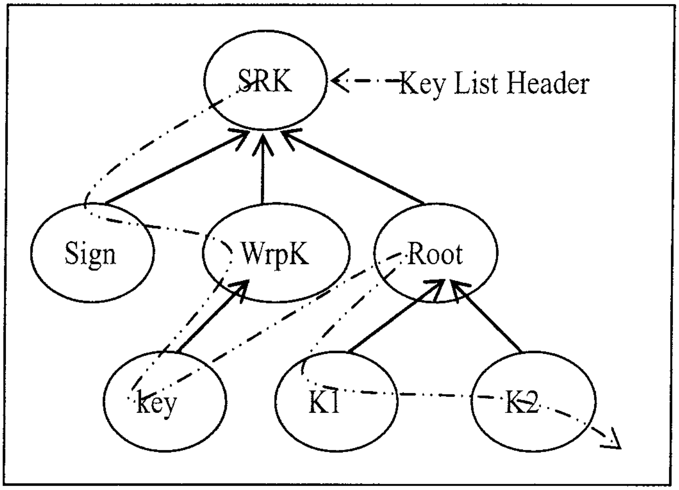

Figure 5. Key Tree

The key tree has a sub-tree marked "root". It managed all the migratable keys in the chip. The "root" key is assigned by user's command, for he always wants to set a "password" to protect his privacy.

The module of keys management is composed of three parts: keys generator, keys storage management and keys import and export. In asymmetrical arithmetic, the true random number generated by TRNG is used to produce big modular number pairs, and the user's public key and private key are produced by procedure according to them. In symmetrical arithmetic, the true random number is directly used for key. Key is managed by sub-module of keys storage management as soon as it is produced. Only public key and key of symmetrical arithmetic which are wrapped by parent key can be exported from chip. SOC provides two basic functions: Export_key and Load_key. In theory, SUP320 can produce unlimited keys through its capability of key generating. But the size of memory on chip is limited, we just design the chip as a portal for keys management. The host must use the two functions to maintain consistency of key view inside and outside the chip.

3.3 Session Management

The module of session management maintains the session information of processes in platform. Any process that wants to intercommunicate with SUP320 must start a session at the very start. A session is discarded when the conversation ends. But we must notice that the number of sessions is changing momentarily. There are lots of processes using the chip in the host, the capability of real-time must be considered. In this project, a list is adopted to manage the session handles. A new session is put ahead of the list when it is created, and a time-out session or an unused session is deleted from the list soon. When looking for a session which belongs to a special process, system searches the list from the head to the end. Other information belonging to a session is saved with the session handle.

3.4 Communication Protocol

The module of communication protocol does two kinds of things. One is to restrict SUP320 to explain input command in specific format. Another is to perform access control on the command according to current session, function and target key.

Different commands have different parameters. We don't intend to describe the format of each command. But each command includes fields such as TAG, Command ID, session information, key information, authorization data, nonce and etc. Table 2 shows most fields of a command.

Table 2. Fields of Command

TCG_TAG	ParamSize	CMD_ID	KeyHandle	InArg	Nonce	...

SUP320 provides two protocols for authorizing the use of entities without revealing the authorization data on the network or the connection to the SUP320. The first protocol is the "Object-Independent Authorization Protocol" (OI-AP), which allows the exchange of nonces with a specific SUP320. Once an OI-AP session is established, its nonces can be used to authorize any entity managed by the SUP320. The session can live indefinitely until either party request the session termination. The TPM_OIAP function starts an OI-AP session. The second protocol is the "Object Specific Authorization Protocol" (OS-AP)". The OS-AP allows establishment of an authentication session for a single entity. The session creates nonces that can authorize multiple commands without additional session establishment overhead, but is bound to a specific entity. The TPM_OSAP command starts the OS-AP session.

To depict problem easily, we suggest: inParamDigest is the result of the following calculation: SHA1(ordinal, inArg); outParamDigest is the result of the following calculation: SHA1(returnCode, ordinal, outArg); inAuthSetupParams refers to the following parameters, in this order: auth Handle, authLastNonceEven,nonceOdd, continueAuthSession; OutAuthSetupParams refers to the following parameters, in this order: auth Handle, nonceEven, nonceOdd, continueAuthSession. Then steps of OI-AP list below:

1. The caller sends command TPM_OIAP to start a conversation with SUP320;
2. SUP320 creates a new session, gets a new authHandle, associates session with authHandle. Gets a true random number to use for authLastNonceEven, saves authLastNonceEven with authHandle. Returns both authLastNonceEven and authHanle to the caller.
3. The caller receives anthHandle and authLastNonceEven and saves them. Gets a true random number to use for nonceOdd, computes inAuth = HMAC(Key.usageAuth, inParamDigest, inAuthSetupParams). Saves nonceOdd with authHandle, sends a command TPM_example, whose message include nonceOdd, authHandle and inAuth.
4. SUP320 receives command TPM_example, verifies authHandle is valid, retrieves authLastNonceEven from internal session storage, computes HM = HMAC (key.usageAuth, inParamDigest, inAuthSetupParams). Compares HM to inAuth, if anything is ok, executes TPM_Example and creates returnCode. Generates nonceEven to replace authLastNonceEven in session. Sets resAuth = HMAC (key.usageAuth,outParamDigest, outAuthSetupParams). Returns output parameters, message returned

includes nonceEven, resAuth and continueAuthSession. If continueAuthSession is FALSE, then destroys the session.

5. The caller saves nonceEven, HM = HMAC (key.usageAuth, outParamDigest, outAuthSetupParams). compares HM to resAuth. This verifies returnCode and output parameters.
6. The caller can use key authHandle to a new key, just follows step 3 to step 5 until the command of Terminate_Session is executed.

The difference of OS-AP and OA-IP is that: OS-AP names the entity to which the authorization session should be bound. More detail information can be found in [1].

3.5 Self Test

The task of self test module is to test the state of hardware components according to the configure parameters when the chip initializes. System on chip reports error information and exits once it finds any component is bad. To test a component effectually, we choose typical test programs for different components. Get test results after execution completed, then we can judge the state of component by the results.

4. METHOD FOR LOW POWER

The method for low power is paid more attention to the embedded system than any other problem. Although SUP320 has four co-processors, it nearly do not use two of them at the same time. So we can disable some co-processor to reduce the cost of chip power when it is not been used. The module of Low Power just does this kind of things, it manages all the states of co-processors and enables or disables the co-processors on behalf of other modules. To simplify the problem, we do not put the true number generator into consideration, for the true number is frequently used in module of communication protocol. In the following, a sub-routine of key-pair generating in RSA arithmetic is used as an example:

1. After SUP320 having done self test, all co-processors are disabled;
2. SUP320 receives the command of Key-Pair generating, procedure of RSA initialization calls module for low power, who enables the co-processor of MMultp;
3. When the operation finished, module for low power disables the co-processor of MMultp in the end of procedure of RSA;
4. SUP320 stores the keys and returns result to the caller.

The method is simple, but test result indicates that the cost of power is almost reduced by 50%.

5. PERFORMANCE

In a Pentium 2.4GHZ, Windows XP installed machine, SUP320 is tested by USB interface. We did some typical operations and compare efficiency of it with that of a common smart card, Compare results list in table 3.

Table 3. Performance Test

Arithmetic	KeySize	Function	SmartCard	SUP320
SHA	160	HASH	N.A.	500M bps
3DES	3*64	Encrypt/Decrypt	100Mbps	210Mbps
AES	128	Encrypt/Decrypt	N.A.	500M bps
RSA	1024bit	Sign	6/s	300/s
	1024bit(e=2^{16}+1)	Authentication	24/s	28000/s
ECC	160bitGF(p)	Sign	N.A.	1200/s
	160bitGF(p)	Authentication	N.A.	600/s

* The symbol "N.A." denotes the device has not this kind of function.

It is obvious that chip of SUP320 is better than a common smart card in both security and efficiency. It is a good choice to build up trust computing platform.

6. CONCLUSION

Building up a secure subsystem based on a physic chip, the whole platform gets security assistance in all levers of hardware, OS kernel and application. We can build a secure system out of insecure environment. Chip of SUP320 is designed by SOC technology, which can bind data and programs together in one chip. It can be used in many fields such as TPM, PKI and etc. The hardware architecture, software modules and method for low power is recommendable to design the system on chip. In the following days, we plan to consider the problems on cooperation of chip and platforms such as PC, PDA etc.

ACKNOWLEDGMENTS

The authors are extremely grateful to the members in trust computing group for their effort on the specifications. This research has been supported in part by The Chinese National Science Foundation(NSF 90104025).

REFERENCES

1. Trusted Computing Platform Alliance (TCPA), Main specification, February 2002. Version 1.1b.
2. Trusted Computing Platform Alliance (TCPA), PC Specific Implementation Specification version 1.0.
3. Trusted Computing Platform Alliance (TCPA), Trusted platform module protection profile, July 2002. Version 1.9.7.
4. Intel Low Pin Count (LPC) interface Specification Revision 1.1.
5. J.E.Dobson and B.Randell, Building Reliable Secure Computing Systems Out of Unreliable UnSecure Compinents, IEEE July 2003.
6. Ross Anderson, TCPApalladium frequently asked questions, http://www.cl.cam.ac.uk/users/rja14/tcpafaq.html accessed 13 March 2003.
7. W. A Arbaugh, D J Farber, and J. M Smith. A secure and reliable bootstrap architecture, In Proceedings 1997 IEEE Symposium on Security and Privacy, pages 65-71, May 1997.
8. Jean-Francois, Design of an Efficent Public-key Cryptographic Library for RISC-based smart cards. Ph.D. Thesis, University Catholique de Louvain,May 1998.
9. Koc,C.K,Acar,T., Burton S.kaliski Jr, Analyzing and Comparing Montgomery Multiplication Algorithms, IEEE Micro 16(3):26-33, june 1996.
10. Tung, C., "Signed-Digit Division Using Combinational Arithmetic," IEEE Trans. On Comp., vol. C-19, no. 8, pp. 746-748, Aug, 1970.