FINGERPRINT AUTHENTICATION IN WIRELESS APPLICATIONS USING EMBEDDED SYSTEMS

N. B. Kalani [1], A. C. Suthar [2], A. M. Kothari[3] ,Dr. G.R.kulkarni[4]

[1]Head Of The Department In E. & C., V.V.P.College Of Engg., Rajkot, Gujarat,
[2] Asst. Prof., Department Of E. & C., C. U. Shah College Of Engg. & Tech., Gujarat,
[3] Lecturer, Department Of E. & C., A.I.T.S., Rajkot, Gujarat,
[4]Principal, C. U. Shah College Of Engg. & Tech., Gujarat,

[1]nbkalani@rediffmail.com, [2]acsuthar@ccetvbt.org, [3]amkothair@aits.edu.in, [4]grkulkarni29264@yahoo.com

ABSTRACT:
Fingerprints are the oldest and most widely used and commonly employed in forensic science to support criminal investigations, and in biometric systems such as civilian and commercial identification devices. The fingerprint of an individual is unique and remains unchanged over a lifetime. However, fingerprint images are rarely of perfect quality. They may be degraded and corrupted due to variations in skin and impression conditions.
In this paper, we describe different methodologies for fingerprint authentication in embedded systems, namely the DSP and System-on-Chip approaches, for Wireless Applications. It allows easy control of the peripherals and fast software reengineering time. This can achieve a fast development. Also, the simplicity of the hardware also enables manufacturers to embed fingerprint authentication systems in the future intelligent devices with low costs.
.

Keywords- Authentication, Finger prints, System On Chip, Embedded Systems

1. INTRODUCTION

Embedded devices like PDAs, cell-phones, etc. are the recent development in mobile commerce, authentication technologies. Fingerprint matching is the most common method in such technologies. Fingerprint authentication involves substantial amounts of computation. Several methods can be used to implement fingerprint authentication in like the Application Specific Integrated Circuit (ASIC), Digital Signal Processor (DSP) and System-on-Chip (SoC) methods in embedded systems. We describe the fingerprint algorithms and its implementation using DSP and SoC methods.

2. EMBEDDED SYSTEMS FOR FINGERPRINT

This system can be divided in two types, according to their purpose of application: identification and verification applications. Identification application means identification of a person from a group of individuals. It often searches a database for the identity of a person using the person's fingerprint as index. The database size can range from a few persons to hundreds of persons. The performance of such system is database size dependent. Access control is a typical identification example. Figure 1. shows the block diagram of a common access control system. Enrollment is conducted when a user first registered a fingerprint in the system. The core part of the system is the authentication device. It needs to handle different connections such as Ethernet and RS232 in order to communicate with the enrollment machine, database server and security controller.

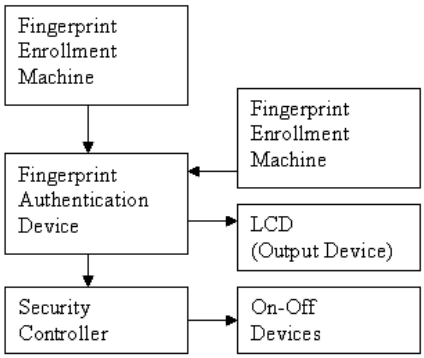


Fig. 1 Block Diagram of physical access control system

Since our concern is limited to embedded systems, we concentrated on those applications involving only a few fingerprint records so that all the components in Figure 1. for one single embedded system Fingerprint authentication often refers to the verification of a person's identity using his/her fingerprints. It is actually a 1-to-1 fingerprint matching problem. Such a problem can be well handled by a desktop PC in real time. The block diagram of fingerprint verification system is shown in Figure 2.
In mobile activities, the authentication will be conducted in a PDA or cell-hone. When implementing such an application in an embedded device like a PDA, we encounter difficulties due to

slower CPU speeds, absence of cache and most important of all, absence of a floating point unit. While optimizing the authentication, we cannot afford to sacrifice reliability. Thus, a fast, low cost and accurate methodology must be developed for fingerprint matching for embedded applications.

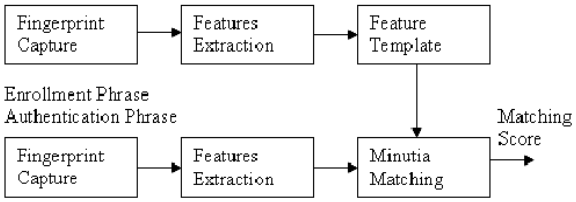


Fig. 2 Fingerprint Verification System

3. BASICS OF FINGER PRINT

A human being's fingerprint contains numerous ridgelines. The end of a ridgeline forms a termination and the merge of two ridgelines forms a bifurcation as shown in Figure 3. A termination or a bifurcation is called a minutia point. The set of the minutia points constitute the features characterizing a fingerprint. The matching of two fingerprints is based on comparing the locations of the minutia points.

In this paper, our focus aims at accomplishing the fingerprint verification in embedded processors in real time while for supporting other security [1]. As shown in the lower part of Figure 2, the authentication process is made of three steps. Fingerprint capture is basically an I/O process that can be accomplished in real time without difficulty.

Fig. 3(a) Bifurcation (b) Termination

Minutia matching is point-matching technique. The most time consumption processing is the feature extraction step. It requires the accuracy and performance of the authentication process.

Feature extraction can be divided into several parts [2]. They are image enhancement, segmentation, image orientation, core point location and minutia extraction.

1) Image enhancement/filtering
   This part aims at enhancing the image quality before feature extraction. Common techniques like the application of a directional filter or image normalization is often employed.
2) Image segmentation and orientation
   This part computes the orientation of the fingerprint image and identifies the Region of Interest.
3) Core point location
   This part locates a core point for fingerprint alignment and matching. It is computed from the orientations of the ridgelines obtained from Image segmentation and orientation.
4) Minutia extraction
   This part traces the ridgeline with the helps of orientation information and find out the bifurcation and termination features.

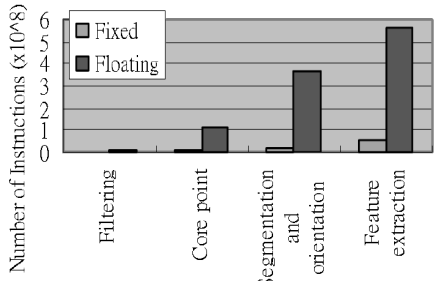
In general, the feature extraction process is a n^3 process, assuming that that fingerprint is a n by n image. Steps (1) to (4) involve intensive floating point calculations, especially approximations of trigonometric functions.

4. IMPLEMENTATION

The design of a fingerprint matching system requires the considerations of cost, development time and runtime performance. A real time response is the critical factor.

The DSP approach

As the cost of the embedded fingerprint system should be as low as possible, while using a fixed



point DSP rather than a floating point DSP [3].

There are two methods to implement a fingerprint program on the DSP platform:

1) Translate all the C source code to assembly language manually. This method gives a very high performance. But it is a very time consuming process.
2) Translate all the C source code to machine language under TI's Code Composer Compiler.

To shorten the integration time, to use the second method. With this approach, simulate the interface of software and hardware components in a short period of time. First, the size of the fingerprint image had to be processed had to be reduced to 64*64 pixels because of memory-limitation. Second, our original program generated a lot of intermediate data. These data that were previously stored in dynamically allocated memory locations were reallocated to static memory locations since dynamic memory allocation is a very time consumption procedure in DSP processing.

Memory consumption is especially severe. Since the DSP and the host embedded processors operate at different speeds, it would not easy for them to share common memory. Moreover, the DSP accesses its built-in memory much faster than external memory. Therefore, fingerprint image data captured from a

sensor must be transferred from the embedded processor's memory to the DSP before the DSP can process them. In this way, memory requirement is doubled. Figure 4. Shows the number of instructions needed for the DSP to process the algorithm.

The total number of instructions needed for the fixed-point fingerprint algorithm is about 76 million while floating-point fingerprint algorithm needs about 1000 million because the floating-point instructions are emulations only. The time measurement is about 1.9s and 26.1s on fixed-point and floating-point respectively. Obviously, more time will become necessary to run the fingerprint authentication algorithm when the image size is 256*256.

Besides the DSP hardware, the embedded Linux system also contains extra hardware like the sensor, display, etc. The additional hardware connections increase the system complexity and reliability. This arrangement increases the overall cost directly.

SoC Approach

Another approach for embedded fingerprint system is the use of System-on-Chip. SoC is a new type of hardware architecture with complex hardware that integrates different IP cores together. Besides the core processor is inside a SoC, it is also equipped with hardware controllers for different peripherals such as LCD and input/output device controllers. In fact, a SoC processor is more or less like the general-purpose processor but without floating point coprocessor because of cost and power consumption consideration. The requirement that all processing is performed on the SoC helps to reduce power consumption, minimize chip areas and simplify the hardware and software development process [4]. The ultimate question is to identify a suitable SoC processor and build an associated system capable of verifying fingerprints in real time.

Figure 5. shows the software architecture using the SoC design. Like a regular computer system, it can be divided into three layers, hardware layer, kernel layer and application layer. With this approach, multiple applications on a single kernel for extensibility of the embedded fingerprint system. If the layered approach will bring in an overhead that can delay the system response time.

To use embedded Linux as the operating system because of its open source code allows us to customize the software architecture for system. With the use of Linux kernel, standardize the API for device drivers, scheduling and messaging services so as to save a lot of development time and effort.

For the application layer, to translate the fingerprint algorithm. Direct cross-complication of the algorithm is possible but can result in poor runtime performance. Therefore to develop a software reengineering process to port the software to its new environment.

Basically, the reengineering process is made up of five steps:

1) Whenever possible, replace floating-point data by integers,
2) Replace the remaining floating-point calculations by fixed point calculations,
3) Buffer all reusable complex calculations,
4) Avoid the use of subroutines,
5) Recompile using an optimized compiler.

The above procedure appears routine and straightforward. Yet, there are numerous domain specific conversions that must be taken care of. The precisions of the fixed-point system as well as good approximations of the trigonometric functions were our great concerns. Ultimately, the verification time was cut to one second or less.

Many factors, like cost, psychology, number of service providers, etc., contribute to the late coming of mobile commerce activities. Among them, security in mobile is an important one.

The involvement of the Internet in mobile commerce, considering the real life mobile commerce transactions involves business-to-customers, business-to-business.

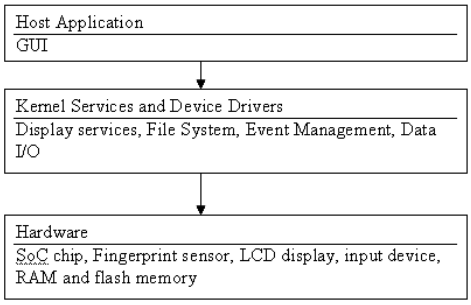


Fig. 5 Software architecture on embedded fingerprint system design

Under such a consideration, security in mobile involves: protection against unauthorized users, protection during data transmission, protection of data on lost devices, protection of mobile assets, and protection of existing security investment. Some issues are covered by choices of the wireless protocols like iMode, HDML, WAP 1.1, SAT (SIM Application Toolkit), and/or the service providers.

To extend the scope of mobile commerce beyond close systems, commercial mobile commerce solutions must migrate smoothly from symmetric algorithms to algorithms, involving Public Key Infrastructure (PKI), for open system frameworks [5].

The current trend of merging the PDAs and wireless telephones merge call for the use of faster embedded processors, to support more features, especially those multimedia ones. In this way, these mobile devices are equipped with the hardware features to verify fingerprints. The next problem is to find a safe place to store the fingerprint features template. The SIM card, which is a smart card itself, is very secure. In fact, the SIM card stores important secret keys, digital certificates as well. The use of fingerprint to protect the SIM card from misuse enhances the confidence of

the confidence of the wireless telephones users to use the devices for mobile transactions [6].

5. CONCLUSION

To model the software development cost and time can be expressed in the following equation.

$$S_E = A * B * S * \prod_{i=1}^{15} Fi \qquad (1)$$

where S_E is software development effort, A is the constant used to compensate the effect of increasing project size, S is the software size in thousands of source lines of code, B is the scale factor accounts for the relative economics or diseconomies of scale, Fi is the cost driver (multiplier) representing such project attributes like time constraints, reliability, software tools and application experience.

Optimization in DSP development process requires extensive programming experience in the operation of the DSP. This increases the cost driver Fi value in above equation and the development effort. When we switch to another type of DSP, utilization of the developed DSP code is very low because the algorithm for DSP is usually hardware dependent.

From the result, we conclude that the use of SoC is suitable for embedded fingerprint system development. It allows easy control of the peripherals and fast software reengineering time. This can achieve a fast development. Also, the simplicity of the hardware also enables manufacturers to embed fingerprint authentication systems in the future intelligent devices with low costs.

6. REFERENCES

[1]. Ho, H.C.; Moon, Y.S.; Ng, K.L.; Wan, S.F.; Wong, Collaborative fingerprint authentication by smart card and a trusted host, Electrical and Computer Engineering, 2000 Canadian Conference on, Volume: 1, 2000 Page(s): 108 - 112 vol.1.

[2]. Maio, D.; Maltoni, D., Direct gray-scale minutiae detection in fingerprints, Pattern Analysis and Machine Intelligence, IEEE Transactions on,Volume: 19 Issue: 1 , Jan 1997 Page(s): 27 – 40.

[3]. Pandey, R, Advances in DSP Development Environments, ELECTRO '96. Professional Program. Proceedings, 1996, pp299-301.

[4]. Chang, H.; Cooke; L.; Hunt, M.; Grant. M.; McNelly, A.; Todd, L., Surviving the SOC Revolution, A Guide to Platform-Based Design, Kluwer Academic Publishers, 1999, pp207-216.

[5]. Mobile Commerce Security: Essential and Analysis, Gemplus, http://www.wmrc.com/businessbriefing/pdf/mcommerce 2001/tech/Gemplus.pdf .

[6]. Asker M. Bazen and Sabih H. Gerez, Extraction of Singular Points from Directional Fields of Fingerprints, Mobile Communications in Perspective, Annual CTIT Workshop, Enschede, The Netherlands, February 2001.