論文2003-40CI-6-5

임베디드 생체 인식 기술 구현 : 지문 보안 토큰 사례

(Implementation of Embedded Biometrics Technologies : A Case of a Security Token for Fingerprints)

金 榮 陳 *, 文 大 城 **, 潘 聲 範 **, 鄭 容 和 ***, 鄭 敎 逸 ****

(Young-Jin Kim, Daesung Moon, Sung Bum Pan, Yongwha Chung, and Kyoll Chung)

요 약

지문 정보 등의 생체 정보를 이용하는 생체 인식 기술은 컴퓨터 시스템의 로그인, 출입 ID, 전자 상거래 보안 등의 여러 서비스에서 사용자의 안전한 인증(authentication)을 위해 널리 사용되고 있다. 근래에 이르러, 생체 기술은 비밀 번호와 같은 기존의 개인 인증 방법에 비해 안전하면서도 자동화를 가져 올 수 있다는 장점으로 인해, 보안 토큰, 스마트 카드와 같은 소형의 임베디드 시스템에 탑재되고 이용되는 추세이다. 본 논문에서는 임베디드 시스템의 형태인 보안 토큰을 개발하고 지문 정보를 이용한 사용자 인증 시스템을 구현하며 시험한 결과를 기술하였다. 보안 토큰과 호스트와의 통신은 USB를 이용하여 시험 및 검증하였으며 보안 토큰 상에서의 지문 정합 프로그램의 수행 시간과 수행 메모리를 측정하고 성능 개선에 대해 기술하였다. 또한, 보안 토큰에서 매치온 카드(match-on-card)로의 전이를 위해 필요한 내용을 언급하였다.

Abstract

Biometric technologies using biometric information like fingerprints features are in wide use for the secure user authentication in many services including log-in of computer systems, entrance ID and E-commercial security. Nowadays, biometric technologies are ported into small embedded systems like securiy tokens or smart cards due to the merit of being secure and automatic in comparison with the previous method in user authentication such as using a PIN. In this paper, the security token developed as an embedded system and the user authentication system implemented and tested using fingerprints information are described. Communications between the security token and the host are tested and verified with USB. And, execution time and runtime memory on the security token board was measured and performance improvement was described. In addition, requisites for the transit from the seurity token to the match-on-card was mentioned.

Keywords : Biometrics, Fingerprint, Security Token, Embedded System

* 正會員, 서울大學校 電氣,컴퓨터工學部
(*School of Computer Science & Engineering, Seoul National University*)
** 正會員, ETRI 情報保護硏究本部
(Information Security Technology Division, ETRI)
*** 正會員, 高麗大學校 컴퓨터情報學科
(Dept. of Computer Information, Korea University)
**** 終身會員, ETRI 情報保護硏究本部
(Information Security Technology Division, ETRI)
接受日字:2002年7月22日, 수정완료일:2003年10月20日

Ⅰ. 서 론

컴퓨터 시스템 로그온이나 출입 ID에 더하여, 근래에는 전자상거래, 전자 화폐 등의 폭넓은 사용에 따라 개인 인증 처리가 많아 지고 있어, 개인 정보의 해킹에 대한 위협도 커지고 있다. 따라서, 개인 정보의 안전한 관리 및 안전한 사용자 인증이 요구되고 있는 실정이다. 현재 주로 사용되고 있는 개인 인증 방법인 패스워에 따른 노출이 쉬워서, 보다 안전한 사용자 인증 방법이 필요하다. 보안 토큰 시스템은 지문, 얼굴, 음성 등의 생체 정보를 개인 식별을 위한 인증 자료로서 이용하는데, 일종의 임베디드 시스템인 match-on-token을 사용한다. Match-on-token은 기준이 되는 사용자 생체 정보를 미리 토큰 내에 가지고 있고, 인증이 필요할 때 외부에서 생체 정보를 받아 들여 토큰 내에서 정합(matching)을 수행한다. 이로써, 호스트를 포함한 토큰 외부에서 일어 날 수 있는 사용자 정보에 대한 해킹을 방지하게 된다. 또한, match-on-token은 스마트 카드를 이용한 match-on-card로 연계되어 발전하고 있다. 또한, 보다 안전성이 강화된, 카드에 지문 입력 센서를 부착한 sensor-on-card도 개발되고 있는 추세이다.

기존의 생체정보를 이용한 개인인증 시스템들은 대부분 PC기반이었다. 본 논문에서 개발한 보안 토큰은 USB I/F를 이용하는데, 소형의 임베디드 시스템이라는 관점에서 기술적으로 스마트카드와 유사하지만, 스마트카드에 비해 다소 충분한 하드웨어 자원을 가지도록 설계가 가능하다. 또한, 스마트카드가 카드리더기와 같은 부가적인 장치를 필요로 하는 반면, 열쇠 크기의 보안 토큰은 대부분의 컴퓨터에 존재하는 USB I/F를 이용할 수 있다는 장점이 있다.

본 논문에서는 match-on-token을 포함하는 보안 토큰 시스템 개발에 대해 기술하였다. Ⅱ장에서는 보안 토큰 에뮬레이터 보드와 호스트를 중심으로 하는 보안 토큰 시스템의 구성 요소 및 각 요소별 설계 사항에 대해 언급하고 Ⅲ장에서는 보안 토큰 시스템에 적합한 지문 인증 알고리즘의 설명과 보안 토큰 시스템을 구현하고 지문 정합 프로그램을 탑재하여 시험하는 내용에 대해 기술하였다. Ⅳ장에서는 지문 정합 프로그램을 수행하여 수행 시간과 수행 메모리를 측정하고 그 내용을 살펴보았다. 마지막으로, Ⅴ장에서는 성능 개선 및 최적화를 위한 내용과 match-on-card으로의 전이를 위해 필요한 내용 등의 향후 연구 방향에 대해 언급하였다.

Ⅱ. 보안 토큰 시스템 설계

일반적인 보안 토큰 시스템은 서론에서 언급한 바와 같이 생체 정보를 이용하여 시스템 내에서 사용자 인증을 수행하므로 인증 과정상의 안전성을 보장하게 된다. 즉, match-on-token상에서 생체 정보 정합을 수행한다. 본 논문에서의 보안 토큰 시스템에서 사용되는 생체 정보는 지문으로 한정하도록 한다.

지문을 이용한 보안 토큰 시스템은 <그림 1>과 같이 사용자 등록(enrollment) 과정과 사용자 인증 과정(authentication)으로 이루어 진다[1]. 사용자 인증 과정은 지문 특징점의 정합을 포함한다.

등록 및 인증 과정에서 입력된 지문 영상을 이용한 특징점 정보 추출과정까지는 호스트에서 수행한다. 등록 및 인증 지문 특징점 정보는 USB 통신 채널을 통해 보안 토큰에 저장한 후에는 외부로 전달되지 않고 개인 기기 내부에서 정합 과정까지 수행하여 인증 결과만을 외부로 출력하도록 한다. 보안 토큰의 메모리 크기 및 CPU 연산 성능에 따라 보안 토큰의 지문 정합 수행 성능이 달라지므로 이에 대한 고려가 반드시 필요하다. 본 장에서는, 보안 토큰 에뮬레이터 보드에서의 H/W 사양 및 S/W 수행 환경 설계, 호스트 프로그램의 설계, USB 통신 및 지문 정합 프로그램의 수행에 대해 기술한다.

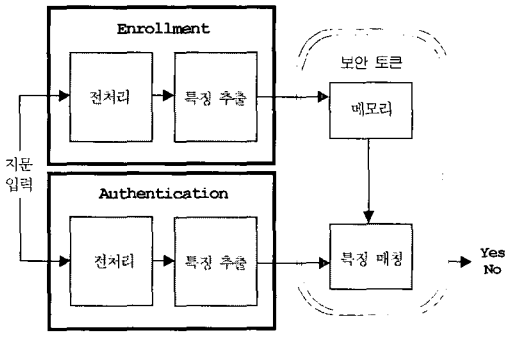


그림 1. 지문을 이용한 match-on-token 시스템
Fig. 1. Match-on-token system using fingerprints.

1. 보안 토큰 에뮬레이터 보드

보안 토큰 보드의 H/W 구조는 <그림 2>와 같다. 보

안 토큰은 에뮬레이터 보드의 형태로 구현하며 ARM7-TDMI를 core로 하는 상용 프로세서를 CPU로 사용한다. 메모리 뱅크를 이용하여 메모리 맵 설정 및 초기화를 하며 소스 레벨의 디버깅을 위해 JTAG을 이용한 AXD 디버거를 사용하도록 하고 상위 레벨의 간단한 디버깅 또는 모니터링을 위해 직렬 통신을 사용한다. 또한, 보드와 호스트 사이의 통신을 위해서 USB 통신을 지원하도록 한다.

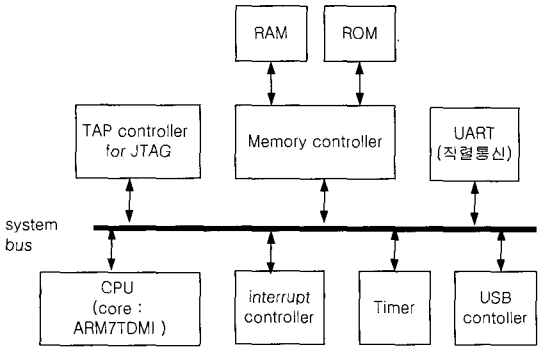


그림 2. 보안 토큰 에뮬레이터 보드 구조
Fig. 2. Architecture of the security token emulator board.

보안 토큰 보드에서의 수행 S/W는 H/W와 직접 인터페이스하는 장치 구동 루틴을 포함하고 이용하는 부팅 코드와 응용 프로그램으로 나뉜다. 장치 구동 루틴은 메모리 설정 관련 레지스터에 대한 동작, 직렬 통신 장치 및 USB 장치에 대한 초기화 및 인터럽트 처리 함수 등이 포함된다. 이를 수용하는 부팅 코드 및 응용 프로그램의 수행 내용은 다음과 같이 구성된다.

① 메모리(SDRAM 등) 설정 및 초기화
② 관리자 모드 설정
③ 인터럽트 제어 register 초기화 및 설정
④ 전역 코드 영역 및 데이터 영역 초기화
⑤ 각 모드별 스택포인터 설정
⑥ 예외 벡터 테이블 설정
⑦ 사용자 모드 설정 및 스택포인터 설정
⑧ 응용 프로그램(지문정합 프로그램)의 메인함수 호출

2. 호스트 프로그램

보안 토큰 보드는 호스트에서 동작하는 클라이언트(client) 프로그램에서 넘겨주는 데이터에 대해 지문 정합을 수행한다. 이 데이터는 기준이 되는 template 지문 특징점 정보와 인증을 위해 입력을 받는 input 지문 특징점 정보이다. 이 지문 특징점 정보들은 호스트에 연결된 지문 스캐너에 연결되어 영상 이미지로부터 특징점 추출을 담당하는 프로그램으로부터 얻어지게 된다.

호스트측에서의 USB 통신을 위한 프로그램은 USB 허브를 제어하고 USB 1.1 spec.에 맞도록 통신 내용을 제어하는 드라이버와 상위 클라이언트 프로그램으로 나뉜 다. USB 드라이버는 Microsoft에 의해 배포된 Windows Driver Development Kit(DDK)를 이용, 수정하여 재작성할 수 있다[2].

3. 지문 정합 프로그램

지문 정합 프로그램은 보안 토큰의 운영 프로그램에 기반하여 USB 통신을 통해 전달되는 template 지문 특징점과 input 지문 특징점을 입력으로 하여 정합을 수행하도록 한다. 지문 정합 프로그램은 특징점 정보의 정렬(alignment)와 정합(matching)의 모듈로 구성되며 수행된 결과는 지문의 일치 또는 불일치를 판별할 수 있는 숫자로 호스트에 반환하도록 한다. 특히, 수행 메모리의 제한을 고려해서 특징점 저장 자료 구조는 전역 변수로 하며, 불필요한 반복 연산을 줄이기 위해 루프 내에 적절한 탈출 조건을 제시하도록 한다.

Ⅲ. 보안 토큰 시스템 구현 및 시험

1. 보안 토큰 시스템 구현

현재 개발중인 보안 토큰 시스템은, ARM core를 내장한 마이크로프로세서 에뮬레이터 보드상에서 ARM Developer Suite(ADS) 1.1로 부팅 코드 및 시스템 소프트웨어를 구현하고 있으며 호스트상에서는 Windows 98 운영체제를 기반으로 MS Visual C++ 6.0으로 클라이언트 환경을 구축하였다. <그림 3>은 구축된 USB 이용 보안 토큰 시스템의 구성을 보이고 있다. <그림 4>는 구현된 보안 토큰 시스템의 양대 구성 요소인 보안 토큰 보드와 호스트의 프로토콜 스택을 나타내고 있다.

에뮬레이터 보드는 ARM7TDMI를 core로 하는 마이크로프로세서인 S5N8946을 사용하며[4], 부팅롬으로 flash ROM 256K bytes, 프로그램 수행 공간으로 SDRAM 16M bytes를 가진다. 특히, 메모리에 대해 보드의 H/W 사양과 보안 토큰의 구현 spec.을 정리하면,

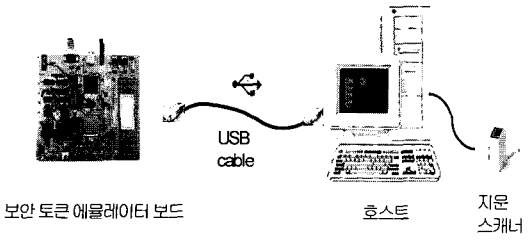


그림 3. 보안 토큰 시스템 구성
Fig. 3. Configuration of the security token system.

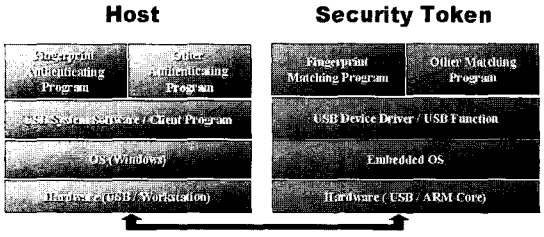


그림 4. 보안 토큰 시스템 프로토콜 스택
Fig. 4. Protocol stack of the security token system.

표 1. 보안 토큰 구현 spec.
Table. 1. Implementation spec. of the security token.

항목 / 메모리	H/W	구현 spec.
ROM	256 k	128 k
RAM	16 M	64 k

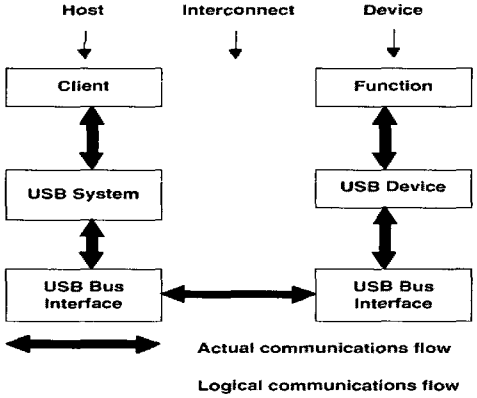


그림 5. USB 통신 모델[3]
Fig. 5. USB communication model[3].

<표 1>과 같다. 구현 spec.의 메모리는, H/W적인 메모리를 부팅 코드에서 제한하고 응용 프로그램을 최적화함으로써 만족시키도록 하였다.

또한, 보드는 디버깅 및 모니터링을 위한 직렬 통신 포트, 지문 추출 정보의 송수신과 이를 이용해 사용자 인증을 하기 위한 USB controller 및 hub를 가지고 있다. 부팅 코드는 SDRAM의 인식 설정 및 초기화를 시작으로 관리자 모드 및 인터럽트 모드에서의 stack point 설정, 인터럽트 제어 register 초기화와 예외 벡터 테이블의 RAM 설치 등을 포함한다.

USB 소프트웨어는 1.1 spec.에 따라 호스트의 드라이버와 bus enumeration을 수행한 뒤에 인터럽트에 의해 송수신을 하도록 구현되었다. USB 통신을 위한 호스트 프로그램은 보드의 USB 디바이스 드라이버에서 지원하는 Bulk 및 interrupt transfer 타입 통신을 모두 지원하도록 구현되어 있다. <그림 5>는 호스트와 보안 토큰에서의 USB 구성 계층 간의 통신 모델을 나타낸 것이며, 최상위에 존재하는 계층, 즉 클라이언트 프로그램과 보안 토큰상의 USB 함수가 구현되어 사용되고 있다. <그림 6>은 지문 정합 프로그램과의 USB 통신을 고려한 지문 정보를 이용하는 보안 토큰 시스템의 동작을 개략적으로 보인 것이다. 이 동작도를 기반으로 실제 보안 토큰 시스템의 수행 절차를 간략히 나타낸 내용이 <그림 7>과 <그림 8>에 나타나 있다.

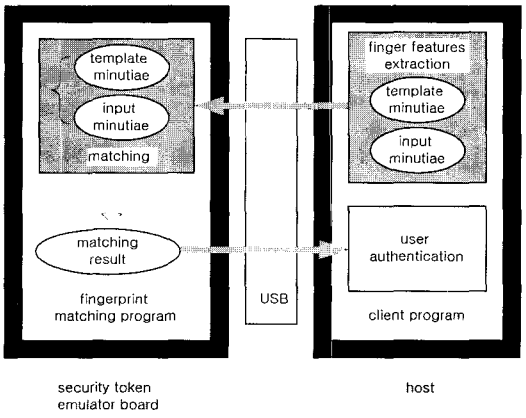


그림 6. 보안 토큰 시스템 동작도
Fig. 6. Execution view of the security token system.

2. 보안 토큰 시스템에 적합한 지문 정합 알고리즘

기존의 일반적인 지문 인증 알고리즘들은[5, 6] 많은 연산량과 하드웨어 자원을 필요로 하여, 대부분 128MBytes 이상의 메모리와 1GHz 이상의 CPU 환경에서 수행하도록 설계되었다. 반면에, 본 논문에서 개발한 보안 토큰은 12MHz CPU, 128KBytes 플래쉬 메모

리, 64KBytes RAM의 하드웨어 자원을 가진다. 따라서, 기존의 알고리즘으로는 현재 개발되고 있는 보안 토큰에서 직접 수행이 불가능하다. 본 연구에서는 보안 토큰과 같은 제한된 하드웨어 자원을 가진 환경에 적합한 새로운 지문 정합 알고리즘을 제안하였다[7]. 본 논문에서 제안한 지문 정합 알고리즘은 비교하기 위한 두 지문을 정렬할 때 피라미드 기법을 이용하여 단계적으로 정확한 정렬 파라미터를 찾아가는 방법을 사용하였다. 실험에 의해 제안한 알고리즘은 적은 메모리 사용으로도 실시간 동작이 가능하다는 것을 확인하였다.

3. 보안 토큰 시스템 시험

보안 토큰 시스템의 시험 절차는 크게 호스트에서의 USB 장치 설치 및 인식 과정과 USB설치 이후의 지문 정합 프로그램 수행으로 나뉜다. <그림 7>과 <그림 8>은 이러한 과정을 보여주고 있다. <그림 7>의 과정은, 보안 토큰 보드에 전원이 가해지고 난 뒤에 보드의 USB 장치 드라이버를 구동하는 부팅 코드와 호스트 프로그램 간에 USB를 사용하기 전에 설치하고 인식하기 위해 통신하는 단계이다. 호스트에 보드의 USB 장치에 대한 정보가 등록되고 이에 대한 주소가 할당된 뒤에 보드에게 통보되면 USB를 이용한 통신이 가능하게 된다. USB를 이용한 보안 토큰상의 지문 정합 시험은, <그림 8>의 수행 절차에 따라 <그림 3>의 보드와 호스트 PC를 연결하여 실시하였다.

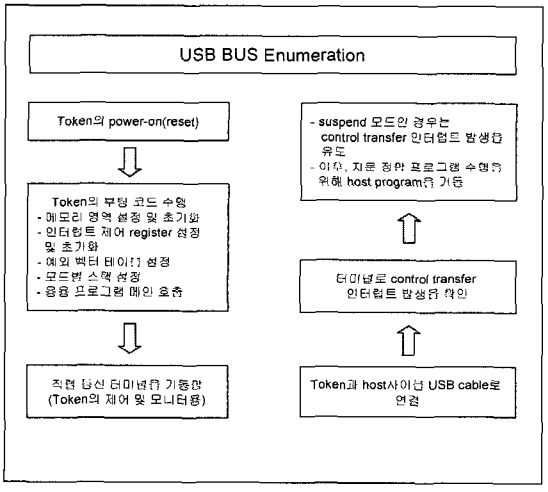


그림 7. 보안 토큰 시스템에서의 USB 설치
Fig. 7. USB installation on the security token system.

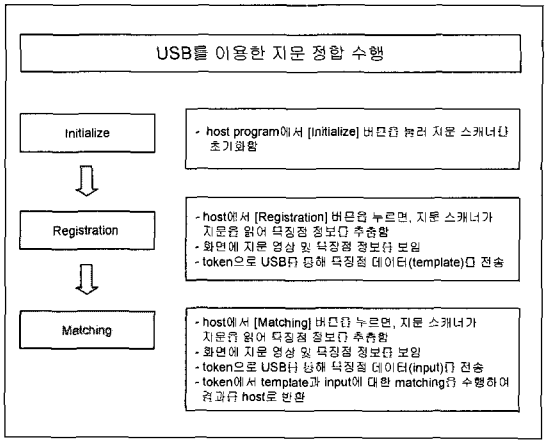


그림 8. USB 이용 지문 정합 수행
Fig. 8. Fingerprint matching execution with USB.

USB 설치 이후, 지문 정합 수행시 Initialize 단계에서는 지문 스캐너를 초기화하며 지문 정합에 필요한 변수들을 초기화한다. Registration 단계에서는 지문 스캐너에 입력된 지문 영상으로부터 특징점 추출 프로그램이 지문 특징점 정보들을 추출한다. 이 정보들을 가지고 있다가 정합 수행에 이용하게 된다. 이때, 기준이 되는 지문 정보는 보통 15 ~ 70개의 특징점으로 구성되는데, 하나의 특징점은 x변위, y변위, 각도 변위, 방향 및 특징점 종류로 구성된다. 또, 입력 특징점 정보는 같은 사용자의 특징점이나, 기준 정보에 대해 특징점 구성요소들의 값들이 상이한 15 ~ 70개의 특징점으로 구성되어 있다. <그림 9>는 직렬 통신을 통해 보안 토큰에서의 프로그램 수행을 제어할 수 있는 터미널 화면을 보이고 있으며, <그림 10>은 Matching 단계에서, 호스트 프로그램이 앞에서 기술한 지문 특징점 정보 2

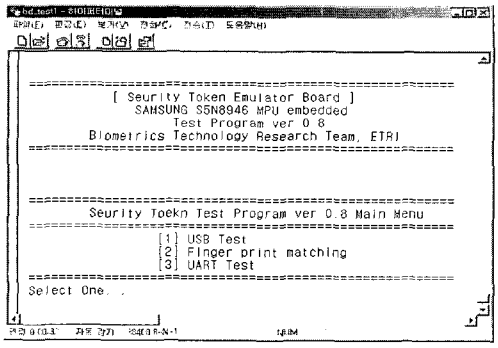


그림 9. 보안 토큰과의 직렬 통신을 위한 터미널 화면
Fig. 9. Terminal capture for serial communication with the security token system.

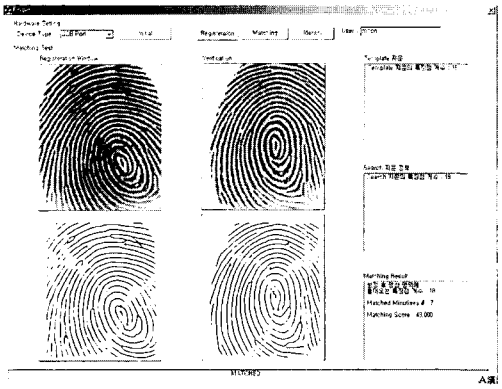


그림 10. 호스트 프로그램 화면
Fig. 10. Host program capture.

쌍의 정보를 보드로 USB를 통해 송신하고 정합 결과를 다시 수신하여 그 결과를 보이는 모습을 나타내고 있다.

Ⅳ. 보안 토큰 시스템 성능 측정 및 비교

본 논문에서 제안한 보안 토큰 시스템의 성능 측정을 위해서 NitGen[8]사의 광학 지문 센서로부터 획득한 영상을 사용하여 실험하였다. 지문 영상의 해상도는 500dpi이며, 크기는 248×292이다. 또한, 실험 영상은 100명에 대해 1인당 4개의 지문 영상을 수집하여 총 400개의 지문 영상을 사용하였다. 실험 영상을 획득할 때 손가락의 위치와 회전 및 압력에 관하여 어떠한 조건도 가하지 않았기 때문에 다양한 화질의 지문 영상으로 실험할 수 있었다.

1. 수행 시간

보안 토큰 보드상에서의 지문 정합 프로그램의 수행 시간은 timer interrupt를 이용하여 측정하였다. 측정하는 내용은 interrupt 횟수와 현재의 timer counter값이며 이를 이용하여 다음과 같이 수행 시간을 계산한다.

$$elapsed\ time = 1\text{주기의 시간} \times \left[interrupt\ 수 + \left(1 - \frac{counter값}{timer초기값}\right)\right] \quad (1)$$

여기서, timer 초기값은 interrupt 구간 길이를 지정해주는 값이다. 시험 data에 대해, 1주기 시간이 1 sec인 경우에 수행 시간 측정 값은 약 1.9sec로 나타났다.

2. 수행 메모리

수행 메모리를 측정하기 위해서 부팅 코드와 라이브러리 및 지문 정합 프로그램 코드의 수행 image가 RAM에서 이용하는 전역 변수를 확인하였다. 그 다음, 전체 RAM 메모리 한계를 제한한 뒤에 사용자 heap 메모리를 초기화하여 크기를 조절하면서 시험하였다. 부팅 코드에서는 지문 정합 프로그램이 사용자 heap을 사용하기 전에 0으로 초기화 하여 heap의 최대 사용 크기를 알 수 있도록 하였다.

시험 data에 대해 사용자 heap을 중심으로 측정된 수행 메모리 및 코드 크기는 다음과 같다.

- RAM 영역 : 6.6k (전역 변수 = 16 bytes, heap = 6 k, 사용자 stack = 600 bytes)
- ROM 영역 : 43k (정합 코드 크기 = 43 k)

3. 성능 비교

측정한 지문 정합 수행 시간 및 수행 메모리에 대해 [7]의 시스템에서의 수행 결과와 비교한 내용은 <표 2>에서 보는 바와 같다. [7]의 시스템에서는 약 60 MIPS 즉, 65 Mhz의 CPU[9]를 사용하고 있는 반면, 보안 토큰 보드에서는 12 Mhz의 clock을 사용하고 있다. CPU성능의 차이에 기인한 수행 시간은 보안 토큰 보드에서 더 많이 요구되는 반면에, 수행 메모리는 보안 토큰 보드가 더 작게 사용하는 것으로 나타났다. 같은 정합 프로그램에 대해 수행 메모리가 달라진 것은, 메모리 관리에 대한 시스템 프로그램의 최적성이 보안 토큰 상에서 더 잘 이루어 져 있다고 볼 수 있다.

표 2. 지문 정합 수행 시간 및 메모리 비교
Table. 2. Comparison of execution time and memory in fingerprint matching.

항목 / 시스템	수행 시간 (sec)	명령어수 (instruction)	수행 메모리 (bytes)
보안 토큰	1.9	17 M	6.6 k
iSAVE[5]	0.29	17 M	10 k

Ⅴ. 결론 및 과제

보안 토큰 시스템은 안전한 사용자 인증의 요구를 만족시키기 위해 사용되며 사용자의 생체 정보를 보안 토큰 상에서 인증함으로써 정보 유출이나 해킹에 대해 안전하다.

본 논문에서는 USB를 이용한 보안 토큰 시스템을 설계 및 개발하고 시험한 내용을 기술하였으며, 특히

지문 정합 프로그램 수행시에 수행 시간 및 메모리를 측정하고 고찰하였다.

최근 스마트 카드의 RAM 크기는 4 ~ 8 kbytes 정도이므로 본 보안 토큰 시스템 상의 결과를 이용한 match-on-card 시스템에의 전이를 위해서는, 수행 메모리 제약에 대한 연구가 필요하다. 또한, 스마트 카드는 카드 리더를 통한 호스트와의 통신에서 규정된 시간 제약성이 존재하므로 수행 시간에 대한 연구 또한 필요하다. 즉, 시간과 메모리에 대한 제약을 동시에 가지는 연구가 필요하다고 볼 수 있다.

현재, 지문 정합 프로그램에서 사용하는 RAM의 크기는 약 6.6 kbytes 인데, 특징점 정보를 암복화하기 위해 암호 알고리즘을 사용하는 경우에는 key 및 중간 결과 data에 이용되는 수행 메모리 사용이 최고 수백 byte이상이 요구될 것으로 예상된다. 따라서, 특징점 저장 공간과 암호 알고리즘의 사용 공간에 대한 효율적인 메모리 공간 이용이 필요하다.

참 고 문 헌

[1] 반성범 외, "사용자 인증을 위한 Match-on-Card 시스템에 관한 연구", 제2회 생체인식기술 워크샵, 2002. 1

[2] 장승석, Bulk transfer를 위한 USB host (Windows 98 PC) 프로그램 구성, 한국전자통신연구원 기술문서(TM-1700-1999-095), 1999.10

[3] Universal Serial Bus Specification 1.1, Sep. 23. 1998.

[4] S5N8946 ADSL/Cable Modem Microcontroller User's Manual, Rev. 1.1, SAMSUNG, 2001.

[5] A. Jain, L. Hong, and R. Bolle, "On-line Fingerprint Verification", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 19, No.4, pp. 302~313, 1997.

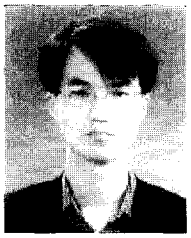
[6] N. Ratha, K. Karu, and A. Jain, "A Real-Time Matching System for Large Fingerprint Databases", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.18, No. 8, Aug. 1996.

[7] DaeSung Moon et al., "A Fingerprint Matching Algorithm using A Multi-Resolution Scheme for Memory-Constrained Environments", IC-AI '02, Vol. I, pp. 124-128, Las Vegas, 2002.

[8] NitGen, http://www.nitgen.com

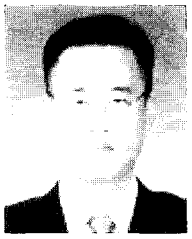
[9] http://www.arm.com.

저 자 소 개



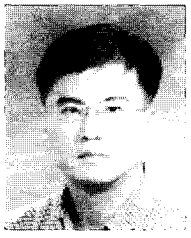
金 榮 陳(正會員)

1997년 : 서울대학교 전기공학부 학사. 1999년 : 서울대학교 전기공학부 석사. 1999년 10월~2003년 1월 : 한국전자통신연구원 정보보호연구본부 연구원. 2003년 2월~현재 : 서울대학교 전기,컴퓨터공학부 박사과정. <주관심분야 : Embedded System, Real-Time System, Java Virtual Machine, IC Card, 정보보호>



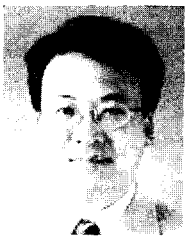
文 大 城(正會員)

1999년 : 인제대학교 전산학과 학사. 2002년 : 부산대학교 컴퓨터공학과 석사. 2002년 12월~현재 : 한국전자통신연구원 정보보호연구본부 생체인식기술연구팀 연구원. <주관심분야 : 생체인식, 영상처리, 정보보호>



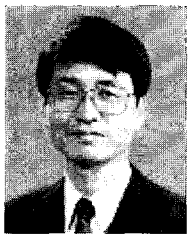
潘 聲 範(正會員)

1991년 : 서강대학교 전자공학과 학사. 1995년 : 서강대학교 전자공학과 석사. 1999년 : 서강대학교 전자공학과 박사. 1999년~현재 : 한국전자통신연구원 정보보호연구본부 생체인식기술연구팀 팀장. <주관심분야 : 생체인식, 영상처리, VLSI 신호처리>



鄭 容 和(正會員)

1984년 : 한양대학교 전자통신공학과 학사. 1986년 : 한양대학교 전자통신공학과 석사. 1997년 : 미국 Univ. of Southern California 컴퓨터공학과 박사. 1986년~2003년 : 한국전자통신연구원 정보보호연구본부 생체인식기술연구팀장. 2003년~현재 : 고려대학교 컴퓨터정보학과 부교수. <주관심분야 : 생체인식, 암호알고리즘, 병렬처리 등>



鄭 敎 逸(終身會員)

1981년 : 한양대학교 전자공학과 학사. 1983년 : 한양대학교 산업대학원 전자계산학과 석사. 1997년 : 한양대학교 전자공학과 박사. 1981년~현재 : 한국전자통신연구원 정보보호연구본부 정보보호기반연구부 부장. <주관심분야 : IC카드, 정보보호, 생체인식, 신호처리>