情報保護學會論文誌
第13券 第4號, 2003. 8

지문 인증을 이용한 보안 토큰 시스템 구현

문 대 성*, 길 연 희*, 안 도 성*, 반 성 범*, 정 용 화**, 정 교 일*

Implementation of A Security Token System using Fingerprint Verification

Dae-Sung Moon*, Youn-Hee Gil*, Do-Sung Ahn*, Sung-Bum Pan*,
Yong-Wha Chung**, Kyo-Il Chung*

요 약

급속한 정보화 및 인터넷의 발달로 인해 네트워크를 통한 정보의 교류가 활발해지고 온라인 뱅킹 등 전자상거래와 관련된 산업의 규모가 커지면서 정확한 개인 인증에 대한 요구가 그 어느 때 보다도 커지고 있다. 이러한 환경에서 가장 일반적인 인증수단으로 사용되고 있는 PIN(Personal Identification Number) 또는 패스워드 방식은 유출 및 망각의 위험이 상존하므로, 이런 문제를 해결할 수 있는 생체 인증에 관한 연구가 활발히 진행되고 있다. 특히, 생체 인증 시스템의 보안 수준을 좀더 향상시키기 위해서 생체 정보의 저장뿐만 아니라 인증까지도 사용자가 휴대 할 수 있는 보안 토큰(스마트카드, USB 토큰) 내부에서 수행하는 연구가 진행되고 있다. 그러나, 보안 토큰의 제한된 하드웨어 자원(메모리, CPU) 때문에 기존의 생체 인증 알고리즘으로는 동작이 불가능하다. 본 논문에서는 206MHz StrongARM CPU, 16MBytes Flash Memory 및 1MBytes RAM의 하드웨어 자원을 가지는 지문 인증 보안 토큰 시스템 구현에 대하여 기술하고, 이러한 보안 토큰 시스템에서 수행이 경량화시킨 지문 인증 알고리즘의 성능을 분석하였다. 실험결과, 본 논문에서 제안한 지문 인증 알고리즘은 6.8KBytes의 메모리를 사용하여 1.7%의 EER(Equal Error Rate)을 제공할 수 있음을 확인하였다.

ABSTRACT

In the modern electronic world, the authentication of a person is an important task in many areas of online-transactions. Using biometrics to authenticate a person's identity has several advantages over the present practices of Personal Identification Numbers(PINs) and passwords. To gain maximum security in the verification system using biometrics, the computation of the verification as well as the store of the biometric pattern has to be taken place in the security token(smart card, USB token). However, there is an open issue of integrating biometrics into the security token because of its limited resources(memory space, processing power). In this paper, we describe our implementation of the USB security token system having 206MHz StrongARM CPU, 16MBytes Flash memory, and 1MBytes RAM. Also, we evaluate the performance of a light-weighted fingerprint verification algorithm that can be executed in the restricted environments. Based on experimental results, we confirmed that the RAM requirement of the proposed algorithm was about 6.8 KBytes and the Equal Error Rate(EER) was 1.7%.

keyword : *Biometrics, Fingerprint authentication, Match-on-Token*

Ⅰ. 서 론

컴퓨터 보급의 확산과 인터넷 등 전자매체의 발달로 인해 온라인 뱅킹과 같은 전자거래의 비중이 점차 높아지고 있다. 이로 인해, 정확한 개인 인증에 대한 요구 또한 커지고 있다. 현재 가장 일반적인 인

* 한국전자통신연구원 정보보호연구본부({daesung,yhgil,dosung,sbpan,kyoil}@etri.re.kr)
** 고려대학교 컴퓨터정보학과(ychungy@korea.ac.kr)

증 수단으로 사용되고 있는 PIN(Personal Identification Number) 또는 패스워드 방식은 유출 및 망각의 위험이 상존하므로, 이에 따른 보안상의 문제가 최근 들어 크게 부각되고 있다. 이러한 PIN 또는 패스워드 방식의 단점을 해결할 수 있는 개인 인증 기술로서 생체 인식이 등장하였다.[1~5] 생체 정보를 이용한 사용자 인증 방법이란 사용자의 지문, 얼굴 모양, 음성 등의 정보를 이용하여 사용자를 인증 하는 것으로, PIN 또는 패스워드 방식에 비해서 타인에 의해 도용되거나 사용자가 암기해야 하는 문제가 발생하지 않는 장점이 있다.

전통적인 생체 인증 시스템의 경우 중앙의 데이터베이스에 생체 정보를 저장한 후, 네트워크를 통해 입력된 사용자의 생체 정보와 비교하여, 사용자를 인증하게 된다. 그러나, 단말기의 센서로부터 입력된 사용자의 생체 정보가 네트워크를 통해 서버로 전달될 때, 불순한 목적을 가진 자에 의해 가로채일 경우 문제가 발생하게 된다. 이런 문제를 해결하기 위해서 OTT(One Time Template)기술 등을 제공하지만, 생체 정보는 PIN 또는 패스워드와 달리 사용자가 필요시마다 변경 할 수 없다. 따라서, 중앙 데이터베이스에 저장된 생체 정보가 외부로 유출된다면 심각한 문제가 발생할 수 있으며, 생체 인증 시스템 도입을 꺼리는 큰 이유인 "Big Brother" 문제가 발생하게 된다.

이러한 문제점을 해결하기 위해서 최근에는 지문과 같은 개인의 생체 정보를 중앙 데이터베이스에 저장하지 않고 보안 토큰(스마트카드, USB 토큰)과 같은 소형의 임베디드 시스템에 탑재하여 사용하고 있다.[6~7] USB 토큰은 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, [그림 1]과 같이 열쇠 크기의 USB 토큰은 대부분의 컴퓨터에 존재하는 USB I/F를 이용한다.

Store-on-Token은 개인의 생체 정보를 단지 저장만 하고, 인증 과정은 호스트 컴퓨터에서 수행한다. 즉, Store-on-Token에서 인증 과정을 수행 할 때 중요한 개인의 생체 정보가 호스트 컴퓨터로 전송된다. 이때, 불순한 사용자에 의해 생체 정보가 가로채일 경우 보안을 보장 할 수 없다. 이러한 문제점을 해결하고 높은 보안 수준을 제공하기 위하여 저장 뿐만 아니라 인증 과정 또한 보안 토큰 내부에서 수행되어져야 하는데, 이런 시스템을 Match-on-Token이라 부른다.

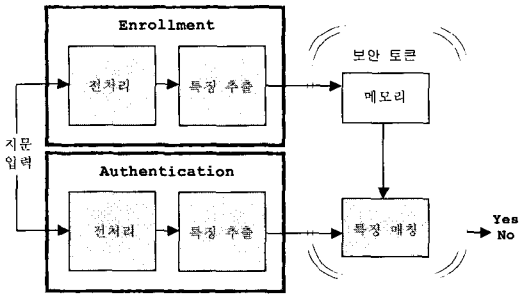
[그림 1] USB 토큰의 예

본 논문에서는 보안 토큰 내부에서 생체 정보의 저장뿐만 아니라 인증 과정이 함께 이루어지는 Match-on-Token 시스템을 구현하였다. 특히, 본 논문에서는 생체 정보 중 가장 일반적으로 사용되고 있는 지문 정보를 이용하였다. 그러나, 기존의 일반적인 지문 인증 알고리즘들은 많은 연산량과 하드웨어 자원을 필요로 하여, 대부분 128MBytes 이상의 메모리와 1GHz 이상의 CPU 환경에서 수행하도록 설계되었다. 반면에, 본 논문에서 개발한 보안 토큰은 206MHz CPU, 16MBytes 플래쉬 메모리, 1MBytes 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. 본 논문에서 제안한 지문 정합 알고리즘은 비교하기 위한 두 지문을 정렬할 때 피라미드 기법을 이용하여 단계적으로 정확한 정렬 파라디터를 찾아가는 방법을 사용하였다. 실험에 의해 제안한 알고리즘은 적은 메모리 사용으로도 실시간 동작이 가능하다는 것을 확인하였다.

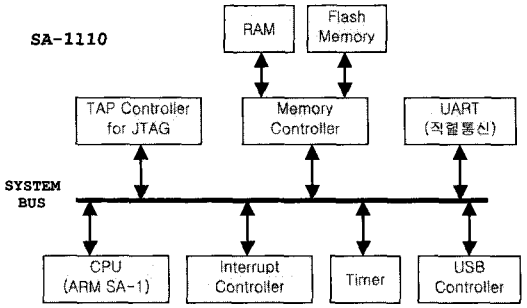
본 논문의 구성은 2장에서 보안 토큰 에뮬레이터 보드와 호스트를 중심으로 하는 보안 토큰 시스템의 구성 요소 및 각 요소별 설계 사항에 대해 언급하고, 3장에서는 제한된 하드웨어 자원을 가진 환경에 적합한 새로운 지문 정합 알고리즘을 설명한다. 4장에서 실험 결과를 분석하고, 5장에서 결론을 내린다.

II. 보안 토큰 시스템 설계

본 논문에서는 다양한 생체 정보 중에서 지문 정보를 이용하여 사용자 인증을 한다. 지문이란 인간의 손바닥에 존재하는 땀구멍이 융기한 선으로 형성된 문형을 말하는 것으로, 융기되어 나타나는 융선(ridges)과 두 융선 사이의 패인 골(valleys)로 나타내어진다. 지문 인식의 방법으로는 영상을 기반으로 하는 방법[8]과 영상 내에 존재하는 특징점을 이용하는 특



〔그림 2〕 Match-on-Token 시스템



〔그림 3〕 보안 토큰 에뮬레이터 보드 구조

징점(minutiae) 기반[5]으로 나눌 수 있다. 본 연구에서는 특징점 기반 사용자 인증 보안 토큰 시스템을 구현하였다.

지문을 이용한 특징점 기반 사용자 인증 시스템은 사용자 등록(enrollment) 과정과 사용자 인증(authentication) 과정으로 수행된다. 사용자 등록 과정은 획득된 지문 영상에서 특징점 정보들을 추출하여 저장하는 과정이며, 사용자 인증 과정은 입력된 지문 영상에서 특징점 정보를 추출한 후 미리 저장된 특징점과 정합(matching)을 수행함으로써 입력된 지문이 저장된 지문과 동일한 지문인지를 판단하는 과정이다.

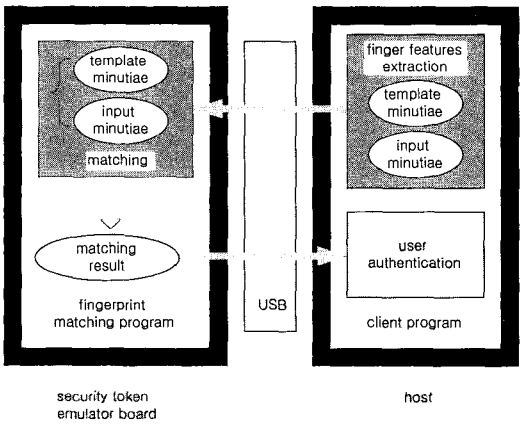
[그림 2]와 같이 Match-on-Token 시스템은 등록 및 인증 과정에서 지문 영상을 입력받아 전처리과정을 거친 후 특징점을 추출하는 과정은 호스트 컴퓨터에서 수행하고, 이미 등록된 특징점과 인증을 위해 새로 입력된 특징점 사이의 유사도를 측정하는 특징 매칭 과정은 보안 토큰 내부에서 수행한다. Gil등[9]에서 보고된 것처럼 전처리 과정과 특징점 추출 과정은 많은 메모리 사용과 명령어(instruction) 수를 요구하여 보안 토큰과 같은 하드웨어 자원이 제한적인 환경에서는 수행이 현실적으로 불가능하기 때문이다. 또한, 사용자의 정보보호를 위해 외부로 유출되지 않아야 할 정보는 지문 영상 자체가 아니라 지문 영상으로부터 추출된 특징점 정보이다.

등록 과정에서 보안 토큰에 저장된 사용자 지문의 특징점 정보는 외부로 전달되지 않고 보안토큰 내부에서 정합 과정을 수행한 후 최종 인증 결과만을 호스트로 전송하기 때문에 개인의 고유한 생체 정보가 외부로 유출되지 않도록 하였다.

개발한 보안 토큰의 H/W 구조는 [그림 3]과 같이 Intel StrongARM SA1110 RISC Processor[10]를 사용하며, Serial port, JTAG interface 그리고, 내장된 USB controller를 통하여 호스트와의 인터페이스를 제공하도록 설계되었다. 전원은 별도의 공급없이 호스트에서 제공하는 전원을 그대로 사용하였다.

〔표 1〕 USB 보안 토큰의 시스템 명세

CPU	32-bit RISC Processor (StrongARM, 206MHz)
Flash Memory	16 Mbyte
RAM	1 Mbyte
크기	7cm×2cm×1cm



〔그림 4〕 보안 토큰 시스템 동작도

표 1은 개발한 보안 토큰의 시스템 명세를 보여준다. 7cm×2cm×1cm 크기의 보안토큰은 206MHz CPU를 탑재하고, 메모리는 16MBytes Flash Memory, 그리고, 1MBytes RAM을 가진다. 특히, 지문 인증 이외에 추가로 화자 인증, 얼굴 인증 및 PKI 등의 많은 다른 응용들이 실시간으로 실행이 가능하게 하기 위하여 성능이 뛰어난 StrongARM processor 사용하였다. 또한, 1MBytes RAM은 지문 인증 수행만을 위해서는 충분한 메모리 공간이지만, 운영체제와 다른 응용들이 사용하는 메모리 공간을 제외하면 실제 지문 인증에 사용되어질 수 있는 메모리 공간은 50KBytes 이하이다.

[그림 4]는 보안 토큰 시스템의 동작도를 보여준다. 호스트에서 추출된 특징점은 USB 인터페이스를 통하여 보안 토큰으로 전송된다. 보안 토큰에서는 미리 저장되어 있던 등록 특징점과 유사도를 측정하여 정합 결과만을 다시 USB 인터페이스를 통하여 호스트로 전달하게 된다.

Ⅲ. 보안 토큰에 적합한 지문 인증 알고리즘

기존의 일반적인 지문 인증 알고리즘들은 AFIS(Automatic Fingerprint Identification System) 등의 응용에 초점을 맞추어 개발되었기 때문에 CPU나 메모리 등의 하드웨어 자원에 제한이 없는 PC 이상의 환경에서 수행하도록 설계되었다. 따라서, 기존의 지문 인증에 관한 연구를 수정없이 보안 토큰에서 직접 적용할 수 없으며, 지문 인증이 제대로 동작하기 위해서는 하드웨어 자원을 적게 사용하는 새로운 지문 인증 알고리즘의 개발이 필요하다. 본 장에서는 일반적인 지문 인증 알고리즘을 설명하고, 이를 기초로 하여 적은 메모리 사용을 위해 피라미드 기법을 이용한 새로운 알고리즘을 설명한다.

3.1 일반적인 지문 인증 알고리즘

일반적으로, 특징점 기반 지문 인증 시스템[5]은 전처리, 특징점 추출, 지문 정합의 세 단계로 이루어진다. 본 논문에서는 보안 토큰 내부에서 수행되는 지문 정합에 관하여 설명한다. 지문 정합 단계는 다시 특징점 정렬(alignment)과 포인트 매칭의 두 과정으로 구성된다. 획득된 지문은 회전(rotation), 천이(translaticn), 잡영 등에 의한 왜곡이 발생하여 단순 정합을 할 수 없으며, 등록 지문 특징점과 인증 지문 특징점 사이의 변화량을 계산해서 좌표계를 맞추어 주는 특징점 정렬 과정이 필수적으로 필요하다. 특히, 특징점 정렬 과정은 기준점의 부재로 인해 복잡하고 메모리 요구량도 클 뿐 아니라, 정렬의 정확성 여부에 따라 전체 지문 인증 시스템의 정확도가 좌우되기 때문에 중요한 부분을 차지한다. 특징점 정렬 과정 이후 정렬된 두 지문의 특징점 정보를 이용하여 유사도를 계산하는 포인트 매칭 과정을 수행한다.

호스트(PC)로부터 추출된 지문 특징점은 아래와 같이 세 가지 구성원소를 가진다.

(1) 특징점의 x,y 좌표
(2) 특징점의 각도(θ)
(3) 특징점의 type(분기점, 끝점)

정렬 과정은 등록 지문의 특징점 집합 P와 인증 지문의 특징점 집합 Q를 입력으로 사용하며, 설명의 편의를 위해서 P,Q를 아래와 같이 정의한다.

$$P=\{(p_x^1, p_y^1, \alpha^1), \ldots, (p_x^p, p_y^p, \alpha^p)\} \quad (1)$$

$$Q=\{(q_x^1, q_y^1, \beta^1), \ldots, (q_x^q, q_y^q, \beta^q)\} \quad (2)$$

여기서 (p_x^i, p_y^i, α^i), (q_x^i, q_y^i, β^i)는 x, y 좌표와 방향정보로 구성된 P와 Q 각각의 i번째 특징점이다. 또한, 동일인의 두 지문에 대해서 두 번째 지문 Q를 정렬 파라미터(회전과 천이)를 이용하여 변환하면 첫 번째 지문 P를 획득할 수 있다고 가정한다.

일반적인 지문 인증 시스템에서는 두 지문 사이의 회전과 위치 변화에 대한 허용오차를 미리 결정하고 동작한다. 즉, 회전과 위치 변화 각각의 정렬 파라미터가 취할 수 있는 범위를 $\theta \in \{\theta_1, \ldots, \theta_L\}$, $\Delta x \in \{\Delta x_1, \ldots, \Delta x_M\}$, $\Delta y \in \{\Delta y_1, \ldots, \Delta y_N\}$ 이라고 정의한다. 두 지문의 좌표계를 일치시키기 위한 정렬 파라미터는 누적 배열(accumulator array) A를 이용하여 구할 수 있다. 여기서 누적 배열 A는 각도의 차이(θ), x축의 차이(Δx), y축의 차이(Δy)를 나타내는 세 개의 축을 가진 3차원 배열이다.

등록 지문의 특징점 집합 P중에서 임의의 한 원소 p와 인증 지문 Q중에서 임의의 한 원소 q, 즉 P와 Q의 모든 쌍 $(p_i, q_j)^k$을 생성한다.

$$(p_i, q_j)^k \in \{(p_1, q_1)^1, (p_1, q_2)^2, \ldots, (p_p, q_q)^{p*q}\}$$

k번째 쌍 $(p_i, q_j)^k$의 x, y 각도에 대한 차이가 θ_k, Δx_k, Δy_k 일 경우, 누적 배열 A에서 해당 $A(\theta_k, \Delta x_k, \Delta y_k)$를 증가시킨다. 모든 쌍 $(p_i, q_j)^k$에 대하여 차이를 계산하고 누적 배열을 증가시킨 후, 최종적으로 최대 누적 값을 갖는 누적 배열 A의 원소 $A(\theta, \Delta x, \Delta y)$를 찾는다. 이 때의 인덱스 $\theta, \Delta x, \Delta y$을 회전과 천이에 관한 정렬 파라미터라 하고, 식 (3)에 적용하면 두 지문의 좌표계를 일치시킬 수 있다.

$$F_{\theta, \Delta x, \Delta y}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \quad (3)$$

위의 정렬 파라미터를 구하는 과정을 예를 들어 설명하면 아래와 같다. 비교를 위해 입력된 인증 지문이 보안토큰 내부에 미리 저장된 등록 지문에 대해서 *x*축으로 +10pixel, *y*축으로 +10pixel, 각도가 +10도 만큼 이동하고 회전되었다고 가정한다. 이는 인증 지문을 x축, y축, 각도에 대해서 각각 -10,-10,-10 만큼 이동하고 회전시키면 등록 지문과 좌표계가 일치되어 두 지문을 중첩시켰을 때 같은 위치에서 특징점의 쌍이 존재하게 된다는 의미이다. 이때, -10,-10, -10이 구하고자 하는 정렬파라미터가 된다. 두 지문의 특징점이 각각 30개라고 가정하고 두 지문 특징점들에서 모든 가능한 쌍은 900개의 쌍이 나오게 된다. 각 특징점 쌍의 차이를 계산한 후, 해당 누적 배열에 누적시키게 되면 (10,10,10)에서 30번의 누적이 발생하게 되며 정렬파라미터가 된다.

정렬 과정에서 이와 같은 일반적인 방법을 사용할 경우, 누적 배열에 요구되는 메모리 공간은 *O(LMN)*이 된다. 예를 들어, *L, M, N*이 각각 64, 128, 128일 경우 누적 배열에 요구되는 메모리 공간은 1,048,576 Bytes이다. 이것은 현재 본 논문에서 구현하고 있는 보안토큰의 하드웨어 명세에 맞지 않아서 동작이 불가능하다. 따라서, 요구되는 메모리 공간을 적게 사용할 수 있는 누적 배열의 구현이 필요하다.

3.2 Match-on-Token 시스템에 적합한 지문 정합 알고리즘

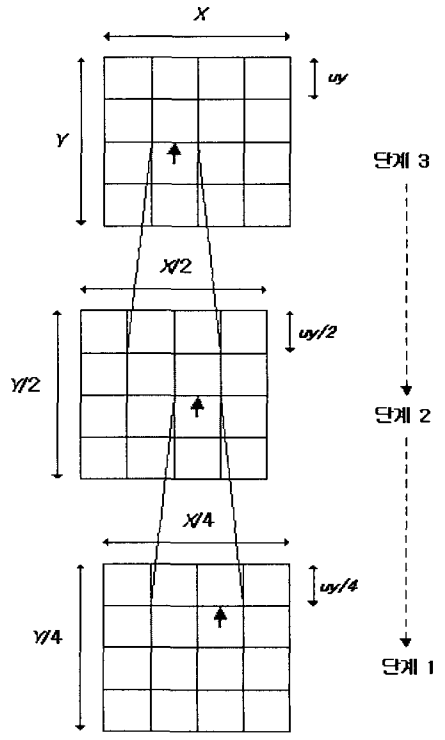
본 논문에서 제안한 방법은 그림 5에서처럼 누적 배열을 사용할 때 피라미드 기법을 이용하여 단계적으로 정확한 정렬 파라미터를 찾아가는 방법을 사용하였다. 제안한 알고리즘의 설명을 위해 아래와 같이 몇 가지 변수를 정의한다.

depth(*d*) : 피라미드의 단계 수
unit angle(*ua*) : 각 단계의 누적 배열에서 각도에 대한 단위
unit x(*ux*) : 각 단계의 누적 배열에서 x 축에 대한 단위
unit y(*uy*) : 각 단계의 누적 배열에서 y 축에 대한 단위
maximum bin([그림 5]에서 ↑) : 각 단계의 누적 배열에서 최대 누적 빈의 위치

제안한 알고리즘 설명의 용이를 위해서 depth *d*는 3으로 가정한다. 또한, 누적 배열은 각도와 *x*값, *y*값을 축으로 하여 3차원으로 이루어지는데 단지 좌표상의 파라미터 중 y축(*uy*)만을 고려해서 설명한다.

depth *d*가 3이기 때문에 세 단계의 반복 과정을 거치게 된다. 먼저 정밀하지 않은 해상도를 가진 단계 3에서 찾은 최대 빈(↑)은 대략적인 정렬 파라미터이다. 단계 3에서 *y*축에 관한 정렬 파라미터의 범위는 *Y*이며 *y*축에 대한 단위 사이즈는 *uy* 이다. 단계 2에서는 단계 3에서 사용한 것과 동일한 메모리 공간을 사용하면서 좀더 정확한 정렬파라미터를 구하기 위해, 단계 3에서 찾은 최대 빈을 중심으로 y축에 관한 정렬 파라미터의 범위는 *Y/2*로 y축에 대한 단위 사이즈는 *uy/2* 로 조정한 후, 단계 3의 과정을 반복한다. 최종적으로, 가장 세밀한 해상도를 가지는 단계 1(*Y/4, uy/4*)에서 정확한 정렬 파라미터를 찾게 된다.

[그림 5]에서처럼 본 논문에서 제안한 방법은 각 단계에서 필요로 하는 메모리 량이 같다는 것을 알 수 있다. 예를 들어, *L, M, N*이 각각 64, 128, 128이라고 가정했을 때, 누적 배열에 사용되는 메모리 공간은 $(64/2^{3-1})*(128/2^{3-1})*(128/2^{3-1})$, 즉 16,384 Bytes이다. 또한, 단계 2와 1에서 필요한 메모리 공간은 각각 $(32/2^{2-1})*(64/2^{2-1})*(64/2^{2-1})$Bytes, $(16/2^{1-1})*(32/2^{1-1})*(32/2^{1-1})$Bytes로 동일하다. 따라서, 제안된 알고리즘이 필요로 하는 메



〔그림 5〕 피라미드 기법을 이용한 지문 정렬 방법

단계 1. 구현할 시스템을 고려하여 누적 배열을 초기화
예) 구현할 시스템에서
가용한 메모리 공간 : 10Kbytes
*L, M, N*은 각각 64, 128, 128
depth $d = 3$
단계 2. *L, M, N*의 unit size 설정
$\theta = \{\theta_l | 1 \le l \le \lfloor 64/2^{d-1} \rfloor \times 2^{d-1}\}$
$\Delta x = \{\Delta x_m | 1 \le m \le \lfloor 128/2^{d-1} \rfloor \times 2^{d-1}\}$
$\Delta y = \{\Delta y_n | 1 \le n \le \lfloor 128/2^{d-1} \rfloor \times 2^{d-1}\}$
각 단계에서 누적 배열 *A*을 누적 배열 *A(i, j, k)*의 최대 누적 위치를 찾고, 이 때의 *i, j, k*를 정렬 파라미터로 한다. 구해진 정렬 파라미터에 대해 두 지문 특징점을 정렬한 후 유사도를 계산
단계 3. $d = d - 1$로 한 후 $d = 1$ 일 때까지 단계 2를 반복

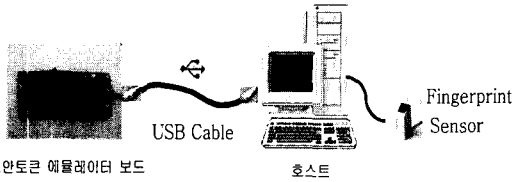
〔그림 6〕 피라미드 누적 배열을 이용한 정렬 절차

모리 공간은 $O(LMN/2^{3(d-1)})$이다. 여기서, d를 $2^{3(d-1)} = L$이 되게 할 수 있으므로, 필요 메모리 량은 $O(MN)$이 된다. 이는 앞서 설명한 일반적인 지문 인증 알고리즘이 필요로 하는 메모리 공간 $O(LMN)$에 비해 적은 메모리 공간을 필요로 함을 알 수 있다. 또한, 일반적인 알고리즘의 명령어(instruction)는 $O(|P||Q|)$만큼 필요한 반면, 제안된 알고리즘의 명령어 수는 약 $O(d'P||Q|)$로 다소 증가 한다. [그림 6]에서 특징점 정렬 단계의 대략적인 절차를 설명한다.

Ⅳ 보안 토큰 시스템 성능 측정 및 비교

이 장에서는 본 논문에서 제안한 지문 인증 알고리즘의 성능을 필요 메모리 량, 명령어 수, 에러율의 세 가지 관점에서 분석한다.

[그림 7]은 개발한 지문 인증 USB 토큰 시스템의 구성이다. 호스트 PC는 지문 센서로부터 지문 영상을 입력받아 전처리 및 특징점 추출 과정을 수행한다. 호스트 PC로부터 추출된 지문의 특징점 정보는 USB 케이블을 통해서 보안 토큰으로 전달되고, 보안 토큰에서는 미리 저장되어 있는 등록 지문의 특징점



〔그림 7〕 보안 토큰 시스템 구성

〔표 2〕 실험 결과

	명령어 수 (Mega Ins.)	수행 시간 (206MHz)	메모리 량 (KBytes)
일반적인 알고리즘	20	약 0.2초	300
제안한 알고리즘	46	약 0.5초	6.8

정보와 비교한 후 최종적으로 인증 여부를 호스트 PC에 전송한다.

본 논문에서 제안한 지문 인증 알고리즘은 Nit-Gen[11]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.

본 논문에서 제안한 지문 인증 알고리즘은 파라미터(ux, uy, ua, d 등)들의 초기화값에 따라 메모리 사용량 및 에러율이 차이를 보인다. 실험을 통하여 $ux = 4$, $uy = 4$, $ua = 2$, $d = 3$ 일 때 보안 토큰의 메모리 규격을 만족하면서 인식 성능이 가장 우수함을 확인하였다. [표 2]는 제안한 알고리즘의 계산시간과 필요 메모리량을 제시하였다.

즉, 제안된 피라미드 기법을 이용한 지문 인증 알고리즘은 약 6.8KBytes의 메모리 공간과 4천6백만 명령어 수를 필요로 한다. 반면에, 피라미드 기법을 사용하지 않고 누적 배열만을 이용한 일반적인 지문 인증 알고리즘은 약 300KBytes의 메모리 공간과 2천만개의 명령어를 필요로 한다. 2장에서 설명한 하드웨어 규격 하에서 구동하는 지문 인증 알고리즘을 구현하기 위해 사용 메모리 공간을 줄이는 것에 초점을 두었다. 그러나, [표 2]에서와 같이 수행시간에서는 크게 차이가 나지 않았다. 따라서, 피라미드 기법을 이용한 지문 인증 알고리즘은 구현한 보안 토큰상에서 실시간으로 동작이 가능하다는 것을 알 수 있다. 또한, 제안한 지문 인증 방법과 일반적인 방법의 EER(Equal Error Rate)은 약 1.7%로 유사한 결과를 보였다.

Ⅴ. 결 론

전자상거래 등의 분야에 안전한 사용자 인증을 제공하기 위해 생체 정보와 보안 토큰과 같은 개인 기

기의 결합이 이루어지고 있는 추세이다. 본 논문에서는 생체 정보의 저장 뿐만 아니라 인증 과정도 보안 토큰 내부에서 이루어지는 Match-on-Token 시스템을 구현하였다. 특히, 하드웨어 자원이 제한적인 보안 토큰에서 실시간으로 동작 가능한 새로운 지문 인증 알고리즘을 제안하였다.

제안한 알고리즘은 지문 인증 과정 중에서 정합 단계에 초점을 맞추었는데, 피라미드 기법이 적용된 누적 배열을 사용하여 정렬 파라미터를 구함으로써 메모리 사용을 줄일 수 있었다. 실험결과, 제안한 지문 정합은 6.8KBytes의 메모리와 약 0.5초의 시간이 소요됨을 확인하였다.

참 고 문 헌

[1] A. Jain, R. Bole, and S. Panakanti, Biometrics: *Personal Identification in Networked Society*, Kluwer Academic Publishers, 1999.

[2] L. Jain, *et al.*, *Intelligent Biometric Techniques in Fingerprint and Face Recognition*, CRC Press, 1999.

[3] F. Gamble, L. Frye, and D. Grieser, "Real-time Fingerprint Verification System", *Applied Optics*, Vol. 31, No.5, pp.652~655, 1992.

[4] 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.

[5] 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.

[6] Kingpin, "Attacks on and Countermeasures for USB Hardware Token Devices", in *Proc. of the Fifth Nordic Workshop on Secure IT Systems Encouraging Co-operation*, pp.35~57, 2000.

[7] M. Janke, FingerCard Project Presentation, http://www.finger-card.org, 2001.

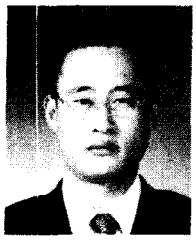
[8] Anil K. Jain and Lin Hong, "Filterbank-Based Fingerprint Matching", *IEEE Trans. on Image Processing*, Vol.9, No.5, 2000.

[9] Y. Gil, Y. Chung, D. Ahn, J. Moon, and H. Kim, "Performance Analysis of Smart Card-based Fingerprint Recognition for Secure User Authentication", in *Proc. of IFIP on E-commerce, E-business, E-government*, pp.87~96, 2001.

[10] Intel, http://www.intel.com.

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

〈著者紹介〉



문 대 성 (Dae-Sung Moon) 정회원
1999년 : 인제대학교 전산학과 졸업
2002년 : 부산대학교 컴퓨터공학과 석사
2002년~현재 : 한국전자통신연구원 정보보호연구본부 생체인식기술연구팀 연구원
<관심분야> 생체인식, 영상처리, 정보보호

길 연 희 (Youn-Hee Gil) 정회원
1999년 : 부산대학교 컴퓨터공학과 졸업
2002년 : 부산대학교 컴퓨터공학과 석사
2002년~현재 : 한국전자통신연구원 정보보호연구본부 생체인식기술연구팀 연구원
<관심분야> 생체인식, 패턴인식, 영상처리

안 도 성 (Do-Sung Ahn)
1992년 : 인하대학교 자동화공학과 졸업
1994년 : 인하대학교 기계공학과 석사
2001년 : 인하대학교 자동화공학과 박사
2001년~현재 : 한국전자통신연구원 정보보호연구본부 생체인식기술연구팀 선임연구원
<관심분야> 지문인식, 생체인식, 정보보호

반 성 범 (Sung-Bum Pan) 정회원
1991년 : 서강대학교 전자공학과 졸업
1995년 : 서강대학교 전자공학과 석사
1999년 : 서강대학교 전자공학과 박사
1999년~현재 : 한국전자통신연구원 정보보호연구본부 생체인식기술연구팀 선임연구원
<관심분야> 생체인식, 영상처리, VLSI 신호처리

정 용 화 (Yong-Wha Chung) 정회원
1984년 : 한양대학교 전자통신공학과 졸업
1986년 : 한양대학교 전자통신공학과 석사
1997년 : 미국 Univ. of Southern California 컴퓨터공학과 박사
1986년~2003년 : 한국전자통신연구원 정보보호연구본부 생체인식기술연구팀장
2003년~현재 : 고려대학교 컴퓨터정보학과 교수
<관심분야> 생체인식, 암호알고리즘, 병렬처리 등

정 교 일 (Kyo-Il Chung) 종신회원
1981년 : 한양대학교 전자공학과 졸업
1983년 : 한양대학교 산업대학원 전자계산학과 석사
1997년 : 한양대학교 전자공학과 박사
1981년~현재 : 한국전자통신연구원 정보보호연구본부 정보보호기반연구부 부장
<관심분야> IC카드, 정보보호, 생체인식, 신호처리