情報保護學會論文誌
第18卷 第1號, 2008. 2

3차원 기하학적 해싱을 이용한 퍼지볼트에서의 지문 정합*

이 성 주[1†], 문 대 성[2], 김 학 재[1], 정 용 화[1‡], 이 옥 연[3]
[1]고려대학교, [2]한국전자통신연구원, [3]국민대학교

A Fingerprint Alignment with a 3D Geometric Hashing Table based on the Fuzzy Fingerprint Vault*

Sung-ju Lee[1†], Dae-sung Moon[2], Hak-jae Kim[1], Yong-wha Chung[1‡], Ok-yeon Yi[3]
[1]Korea University, [2]ETRI, [3]Kookmin University

요 약

바이오정보를 이용한 사용자 인증은 일반적인 패스워드 기반 시스템에 비해 많은 장점을 가지고 있다. 또한, 바이오정보를 이용한 인증 시스템은 높은 보안성과 사용자의 편리성을 제공하기 위하여 암호학과 바이오정보 분야를 암호-바이오(crypto-biometric)시스템으로 통합하여 연구되고 있다. 최근 퍼지볼트라 불리는 암호-바이오 시스템이 보고되고 있다. 이것은 사용자의 중요한 비밀키와 바이오정보를 통합하여 정당한 사용자만이 비밀키를 획득 할 수 있도록 안전하게 보관하는 방법이다. 하지만 기존 연구들에서는 바이오정보를 안전하게 보호하기 위해 추가되는 거짓 특징점의 개수가 제한되어 높은 보안성을 제공하지 못하는 문제가 있다. 본 논문에서는 3차원 기하학적 해쉬 테이블을 이용하여 보안성을 개선하고 추가적인 정보 없이 보호된 지문 템플릿에서 자동으로 지문 정렬을 수행하는 방법을 제안한다. 실험을 통하여 제안한 3차원 지문 퍼지볼트 기법이 추가적인 정보 없이 역변환이 불가능한 변환된 영역상에서 자동으로 지문 정렬을 수행가능하다는 것을 확인하였다.

ABSTRACT

Biometrics-based user authentication has several advantages over traditional password-based systems for standalone authentication applications. This is also true for new authentication architectures known as crypto-biometric systems, where cryptography and biometrics are merged to achieve high security and user convenience at the same time. Recently, a cryptographic construct, called fuzzy vault, has been proposed for crypto-biometric systems. This construct aims to secure critical data(e.g., secret key) with the fingerprint data in a way that only the authorized user can access the secret by providing the valid fingerprint, and some implementations results for fingerprint have been reported. However, the previous results had some limitation of the provided security due to the limited numbers of chaff data for hiding real fingerprint data. In this paper, we propose an approach to provide both the automatic alignment of fingerprint data and higher security by using a 3D geometric hash table. Based on the experimental results, we confirm that the proposed approach of using the 3D geometric hash table with the idea of the fuzzy vault can perform the fingerprint verification securely even with more chaff data included.

Keywords : 지문인식, 퍼지볼트, 기하학적 해싱
접수일: 2007년 5월 17일; 채택일: 2007년 11월 29일
* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 (홈네트워크연구센터) 육성지원사업의 연구결과로 수행되었음
† 주저자, peacfeel@korea.ac.kr
‡ 교신저자, ychungy@korea.ac.kr

Ⅰ. 서 론

공개키 암호 알고리즘은 비밀키를 안전하게 보호한다는 가정 하에서 높은 보안성을 제공할 수 있다. 하지만, 공개키 암호 알고리즘의 비밀키가 도난당하거나 손실되면, 암호 알고리즘의 비도에 상관없이 보안성은 급격하게 떨어진다. 실제로 많은 공개키 암호 알고리즘은 비밀키를 안전하게 보관하고 있다고 가정하지만, 간단한 패스워드를 기반으로 비밀키를 보호하는 인증기법(authentication mechanism)을 사용하고 있다. 즉, 사용자는 패스워드를 이용하여 사용자 인증에 성공할 경우 스마트카드 등으로부터 비밀키를 획득하게 된다. 그러나 대부분의 응용에서 지원하는 패스워드의 길이가 충분히 크지 않으며, 지원되는 패스워드 크기 내에서 조차도 사용자들은 충분히 랜덤한 패스워드를 사용하는 대신 사용자가 기억하기 쉬운 문자 등으로 여러 응용에 동일한 패스워드를 사용하고 있는 실정이다. 따라서, 패스워드는 쉽게 추측 또는 유출이 될 수 있으며 부인방지 기능을 제공하기 어렵다는 취약점을 갖는다[1].

패스워드 기반의 사용자 인증 방식이 갖고 있는 이러한 문제점들은 사용자의 바이오정보를 이용함으로써 많은 부분 해결 될 수 있다. 바이오정보를 이용한 사용자 인증 방법이란 사용자의 지문, 얼굴 등의 신체 정보를 이용하여 사용자를 인증하는 것으로, 패스워드 방식에 비하여 많은 정보를 이용함으로써 타인에 의해 도용되거나 사용자가 암기해야 하는 문제가 발생하지 않는 장점이 있다. 본 논문에서는 다양한 바이오정보 중 가용성, 정확도, 경제성면에서 현재까지 가장 현실적인 대안으로 평가받고 있는 지문을 선택하였다[2].

그러나, 이러한 지문을 이용한 사용자인증 시스템에서도 몇 가지 문제가 있다. 먼저, 패스워드와 달리 동일인의 동일 손가락으로부터 입력되는 지문정보가 매번 유사하지만 완전히 일치되지 않는다는 특징 때문에 기존의 암호학적 일방향 해쉬 함수를 적용할 수 없으며, 유출시 대체할 수 있는 지문의 개수가 한정되며 여러 응용에 상이한 지문을 사용하는데 어려움이 있어, 지문정보 보호 문제가 심각한 이슈로 등장하고 있다[3-5].

암호 시스템과 바이오인식 시스템을 통합하여 바이오정보를 보호하는 방법에는 다음 두 가지가 있다[6-20]. (i) 소결합(loosely-coupled) 방식 : 소결합 방식에서 바이오인식은 암호 시스템과 분리되어 있으며, 바이오인식을 통하여 비밀키를 획득하는 과정은 다음과 같다. 두 바이오정보를 비교하여 정당한 사용자라고 인증되면, 사용자는 스마트카드 등에 안전하게 보관되었던 비밀키를 획득할 수 있다. 이러한 방법이 대부분의 바이오인식 시스템에서 사용되지만, 저장된 바이오정보 템플릿을 또 다른 암호키나 패스워드로 보호해야하는 "닭-달걀"의 문제점이 있다. (ii) 밀결합(tightly-coupled) 방식: 밀결합 방식에서는 암호 시스템과 바이오인식 시스템을 하나로 통합하여 사용자 인증을 수행한다. 가령, 바이오정보가 변환된 영역에서 인식 과정이 이루어진다면, 공격자는 변환된 바이오정보를 획득하더라도 사용자의 원정보를 복원 할 수 없다. 일례로 Juels와 Sudan는 퍼지 개념을 적용한 "퍼지볼트"라는 암호이론을 통하여 바이오정보와 비밀키를 동시에 보호할 수 있는 방법을 제안하였다[14].

본 논문에서는 퍼지볼트 이론을 기반으로 하여 지문인식을 수행할 수 있는 방법을 제안한다. 퍼지볼트 이론을 이용하여 지문정보를 보호하기 위해서는 거짓 특징점(chaff minutiae)을 생성하여 사용자의 지문 특징점(real minutiae)과 함께 데이터베이스에 등록하는데, 이러한 사용자의 지문 특징점과 거짓 특징점으로 구성된 정보를 "지문 템플릿"이라고 지칭한다. 최근 퍼지볼트를 이용하여 사용자의 지문정보와 그 사용자의 비밀키를 동시에 보호하려는 연구가 다수 보고되고 있으나[17,18], 지문의 기준점 부재에 따른 정렬 과정을 생략하였기 때문에 실용화 할 수 없는 문제점이 있었다. 또한, 퍼지볼트를 이용한 지문정보 보호시스템의 안전도는 사용자의 지문 특징점과 거짓 특징점으로 구성된 지문 템플릿으로부터 사용자의 지문 특징점만을 찾아내는 복잡도에 근거하기 때문에, 추가된 거짓 특징점의 수에 따라 시스템의 안전도가 증가한다는 특징이 있다. 그러나, 고정된 크기의 지문센서에서 획득된 지문정보에 거짓 특징점의 개수를 증가시키는데 한계가 있으며 거짓 특징점의 수가 증가되면 지문 인식률이 저하된다는 문제 때문에, 기존 연구에서는 추가되는 거짓 특징점의 수를 200개로 한정하였다.

본 연구에서는 기하학적 해싱(Geometric Hashing)[21] 기법을 이용하여 지문정보가 역변환이 불가능한 영역으로 변형된 상태에서 자동으로 정렬을 수행하는 방법을 제안하고, 3차원 해쉬 테이블을 이용하여 기존의 방법들에서 사용한 거짓 특징점 보다 많은 수의 거짓 특징점을 추가하여 높은 보안성을 확보하는 방법을 설명한다. 먼저, 1:N 지문인식(fingerprint identification)

에서 활용되는 기하학적 해싱 기법을 1:1 지문검증(fingerprint verification) 과정에서 수행되도록 수정한다. 그리고 해쉬 테이블은 역변환이 불가능하도록 지문 템플릿을 생성하고, 복원과정 없이 정렬이 가능하도록 수정한다. 실험을 통하여 본 논문에서 제안한 3차원 해쉬 테이블을 이용하여 1,000개의 거짓 특징점을 삽입하더라도 기존의 연구방법에 따른 200개 거짓 특징점을 추가한 경우와 비교하여 정합되는 특징점의 평균 개수를 확인함으로써 추가적인 정보 없이 역변환이 불가능한 변환된 영역상에서 자동으로 지문 정렬을 수행가능하다는 것을 확인하였다.

본 논문의 구성은 다음과 같다. 2장에서는 퍼지볼트를 사용하여 비밀키를 보호하는 기존 연구들을 설명하고, 3장에서는 본 논문에서 제안한 3차원 해쉬 테이블을 사용하여 지문 템플릿을 생성하는 과정과 보호된 템플릿으로 지문인식을 수행하는 방법을 설명한다. 4장에서는 구현 및 성능을 평가하고, 마지막으로 5장에서는 결론을 맺는다.

Ⅱ. 연구배경

2.1. 패스워드 기반 비밀키 보호 방법

일반적으로 암호 시스템에 사용되는 비밀키는 사용자의 패스워드나 패스프레이즈(passphrase) 기반의 인증기법에 의해 보호된다. 하지만, 많은 사람들은 패스워드를 가족이나 본인의 이름이나 생일, 좋아하는 영화나 음악, 사전의 단어 등 기억하기 쉬운 문자나 숫자로 설정하며, 이러한 패스워드는 간단한 사전공격(dictionary attack) 등으로 쉽게 공격당할 수 있다. 즉, 높은 보안성을 위해 매우 길고 랜덤한 패스워드를 사용한다면 사용자가 기억하기 어렵고 사용하기 불편하여, 대부분의 응용에서는 크지 않은 길이의 패스워드를 지원하며 사용자들은 그나마 지원되는 패스워드의 엔트로피를 최대한 활용하지 못하고 있는 실정이다. 또한, 다른 응용에서 각각 다른 패스워드를 생성/사용/관리하도록 권고하고 있지만 대부분의 사용자들은 동일한 패스워드를 여러 응용에서 사용하기 때문에, 하나의 응용에서 패스워드가 노출되면 전체 응용의 패스워드가 노출된다는 문제가 있다.

패스프레이즈를 사용한 방법 역시 유사한 문제점이 있다. 패스프레이즈 방식은 사용자가 비밀키를 획득하기 위해서 매우 긴 문자나 숫자 등을 기억해야하지만, 이 과정에서도 사용자는 긴 문자나 숫자 등을 망각할 수 있다. 이러한 문제점을 해결하기 위해 사용자만이 알 수 있는 일련의 질문에 대해 응답을 확인하는 방법이 있다[22]. 그러나, 이러한 방법은 실시간 인증이 어려워 많이 사용되지 않고 있으며, 비밀키를 보호하는 편리한 방법으로 바이오정보를 이용하는 방법들이 연구되고 있다.

2.2. 지문정보를 이용한 비밀키 보호

1장에서 언급한대로, 소결합 방식에서는 암호 시스템과 바이오인식은 독립적으로 동작하며 입력 바이오정보와 데이터베이스에 저장된 바이오정보 템플릿을 비교하여 같은 사용자라고 확인되면, 사용자는 스마트카드 등에 안전하게 보관되었던 비밀키를 획득한다. 이러한 방법이 현재 바이오인식 시스템에서 많이 사용되지만, 저장된 바이오정보 템플릿을 또 다른 암호키나 패스워드로 보호해야하는 "닭-달걀"의 문제가 있어, 본 논문에서는 밀결합 방식으로 바이오정보와 비밀키를 동시에 보호하는 방법에 국한하여 설명한다.

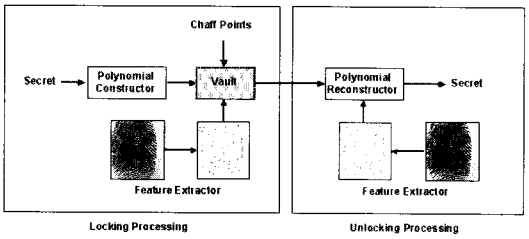
밀결합 방식의 초기 연구로는 biometric encryption [12], fuzzy commitment[13] 등을 들 수 있다. 예를 들어, bit commitment[1]에 퍼지 개념을 도입한 fuzzy commitment에서는 에러 정정 코드인 Hamming 또는 Reed-Solomon 코드를 이용하여 인코딩한 후, 바이오정보와 XOR 연산을 통해 템플릿을 생성한다. 비밀키를 획득하기 위해서는 새로 입력된 바이오정보와 저장되어 있는 템플릿의 XOR 연산을 하고, 그 결과를 에러 정정 코드를 통해 디코딩한다. 이때 새로 입력된 바이오정보와 저장된 템플릿의 차이는 복호화 과정을 통해 복원될 수 있다. 그러나, 이 방법은 지문의 특징점과 같이 바이오정보의 순서가 일부 변경되거나 삭제될 경우 동작하지 않는다는 문제가 있다.

최근 Juels와 Sudan[14]는 자신들이 제안한 fuzzy commitment의 문제점을 해결할 수 있는 fuzzy vault라는 암호학적 방법을 제안하였다. 즉, 입력되는 지문 특징점의 순서 변경이나 삭제를 해결하기 위해 지문 특징점들을 집합으로 취급하는 secret sharing 개념[1]을 도입하였고, 시스템의 안전도는 polynomial reconstruction 문제[1]에 기인한다. 동작 과정을 자세히 설명하면, 먼저 Alice가 비밀키 k를 이용하여 다항식 p를 생성한

후, 자신의 A 집합(지문 특징점으로부터 생성될 수 있는)을 이용하여 p(A)를 계산한다. 그리고, 랜덤하게 거짓 정보(지문의 경우 거짓 특징점)를 A 집합에 추가함으로써 R 집합(지문의 경우, 저장되는 지문 템플릿)을 생성한다. Bob(설명의 편의를 위해 R 집합 생성 후 비밀키 k를 획득하기 위해 자신의 집합을 이용하는 사람을 Bob으로 명명)은 비밀키 k를 획득하기 위하여 자신의 B 집합을 사용하는데, B 집합과 A 집합이 상당히 유사하다면(지문의 경우, 저장된 지문 특징점 집합 A와 입력된 지문 특징점 집합 B가 상당히 유사하다면), Bob은 R 집합에서 충분한 수의 p(A) 정보를 획득할 수 있고, 에러 정정 과정을 거쳐 거짓 정보를 제거한 후 다항식 p를 풀어 비밀키를 획득 할 수 있다. 특히, fuzzy vault는 두 집합의 유사도 따라 다항식을 풀 수 있도록 동작하기 때문에, 동일 손가락으로부터 입력되는 지문정보의 특징점들이 순서가 변경되거나 누락 또는 추가되는 지문인식에 적합하다는 특징이 있다.

이러한 퍼지볼트의 특징을 지문인식에 적용시킨 연구가 최근 발표[17,18]되고 있는데, 모두 지문인식의 중간단계인 정렬과정이 수행되었다고 가정하거나 수동으로 정렬하였기 때문에, 현실적이지 못하며 자동화된 시스템을 구현할 수 없다는 문제점이 있다. 설명을 돕기 위해서 지문 퍼지볼트에 대한 전체적인 시나리오를 [그림 1]에서 보여주는데, locking processing 단계에서는 비밀키 k, 수십개의 지문 특징점으로 구성된 집합 A, 수백개의 거짓 특징점(그림에서 chaff point로 표현)으로 locking set R(그림에서는 vault로 표현)을 생성하고, unlocking processing 단계에서 A와 유사한 지문 특징점 집합 B와 locking set R로부터 비밀키 k를 복원한다. 지문 퍼지볼트에 대한 보다 자세한 내용은 [16-18]에서 확인할 수 있다.

또한, 지문 정렬 문제(저장된 지문정보와 입력된 지문정보의 유사도를 측정하기 위해서는 먼저 두 지문을



[그림 1]. 지문 퍼지볼트 시나리오

정렬해야 하는데, locking set R에 수백개의 거짓 정보가 추가되었고 이중 어느 것이 사용자의 지문특징점 정보이고 어느 것이 추가된 거짓 정보인지 구분할 수 없는 상태에서 두 지문을 정렬하는 문제)를 해결하기 위해 기하학적 해싱 기법[21]을 적용하여 자동으로 정렬을 수행하였다[20]. 그러나 지문 센서에서 입력받은 지문영상의 크기가 제한적이기 때문에 거짓 특징점을 삽입 할 수 있는 최대 개수가 제한되는 상황(이전 연구들은 거짓 특징점의 개수를 200개라고 가정)에서 컴퓨터 성능의 발전 속도를 감안할 때 보안성을 높이기 위해 더 많은 거짓 특징점을 추가할 수 있는 새로운 방법이 필요하다. 본 논문에서는 3차원 해쉬 테이블을 이용하여 더 많은 거짓 특징점을 삽입할 수 있고 추가적인 정보 없이 역변환이 불가능한 변환된 영역상에서 자동으로 지문 정렬을 수행하는 방법을 제안한다.

2.3. 기하학적 해싱(Geometric Hashing)

모델기반 객체인식 시스템(model-based object recognition)은 객체의 집합으로부터 어떤 한 객체와 동일한 객체를 포함하는 장면을 찾아내는 시스템이다. 이때, 객체는 점이나 선으로 표현될 수 있고 기하학적 방법으로 변형된 객체의 정보를 특징(feature)이라하며 특징을 최소화하여 객체정보를 구성한다. 이렇게 만들어진 객체정보와 비교할 객체정보는 AND 또는 OR 연산을 하고 그 결과에 따라 유사도를 확인 할 수 있다.

많은 객체 인식 시스템은 장면과 객체 사이의 특징을 예측 또는 검증을 하는 방법으로 이용한다. 기하학적 해싱 방법은 기존의 방법과는 다르게 전처리 단계와 인식 단계를 병렬적으로 처리하기 때문에 인식과정의 수행시간이 효율적이며 많은 응용에서 활용되고 있다. 특히, 일부 가려진(occluded) 객체를 인식할 수 있는 기하학적 해싱의 특징은 수많은 거짓 정보를 포함한 볼트로부터 실제 사용자의 지문 특징점을 구분해내야 하는 지문 퍼지볼트의 상황에 적합하다. 기하학적 해싱 알고리즘은 다음과 같이 전처리 과정(등록과정)과 인식 과정(식별과정)으로 구분된다.

전처리 과정:

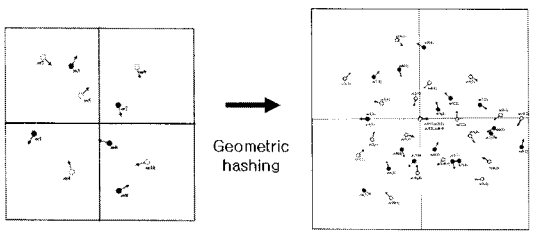
전처리 과정은 오프라인으로 단 한번만 수행된다. 우선, 객체의 점으로 표현된 특징점은 데이터베이스에 저장되기 위하여 기하학적 변환에 의해 많은 정

보를 포함한 해쉬 테이블 형태로 저장된다. 가령, 객체의 특징점 n개가 데이터베이스에 저장되었다면, 한 개의 특징점을 기준점으로 선택하여 쌍이 되는 다른 특징점과 함께 회전이나 이동의 기하학적 변환을 이용하여 좌표축의 중심으로 이동시킨다. 그리고 나머지 특징점들을 같은 방법으로 변환시키고, 변형된 좌표에는 해쉬 테이블의 bin이라고 정의된 방법으로 표시한다. 이렇게 n개의 모든 특징점에 대하여 기준점을 선택하고 나머지 특징점들을 같은 방법으로 변환시킨다.

인식 과정:

온라인으로 수행되는 인식 과정은, 인식하려는 객체의 특징점 S를 추출하고, 임의의 특징점을 기준점으로 선택하여 쌍이 되는 다른 특징점과 함께 회전이나 이동의 기하학적 변환을 이용하여 좌표축의 중심으로 이동시킨다. 그리고 나머지 특징점들을 같은 방법으로 변환시킨다. 이렇게 생성된 테이블은 전처리 과정에서 생성된 해쉬 테이블과 비교되어 bin의 수를 올려준다. 다시 n개의 모든 특징에 대하여 기준점을 선택하고 위와 같이 나머지 특징점들을 같은 방법으로 변환시킨 후, 최종적으로 가장 높은 점수를 가진 bin이 가장 유사한 객체라고 인식한다. 특히, 기준점을 선택하는 것은 상황에 따라 한 개의 특징점을 선택 할 수 있고, 두 개 이상이 될 수도 있다. 또한, 객체의 특징은 점이 아닌 선으로도 표현될 수 있다.

[그림 2]는 기하학적 해싱 기법을 적용한 지문 퍼지 볼트의 전처리 과정(등록과정)을 보여준다. [그림 2(a)]는 총 10개의 진짜(흰색)와 가짜(검정색)로 구성된 특징점이며, 진짜와 가짜를 포함한 모든 특징점에 대하여 기준점을 선택하고 나머지 특징점들이 이동 및 회전 변환을 거쳐 생성된 모습을 [그림 2(b)]에서 보여주고 있다. 2차원 해쉬 테이블을 이용한 지문퍼지볼트의 자세한 내용은 [20,21]에서 잘 설명되어 있다.



(a) (b)

[그림 2]. 2차원 등록 해쉬 테이블 생성 과정 (a) x, y좌표에 존재하는 진짜(흰색)와 거짓(검정색) 특징점, (b) 기하학적 해싱 기법이 적용되어 변환된 진짜와 거짓 특징점

Ⅲ. 3차원 해쉬 테이블을 이용한 퍼지 볼트

본 장에서는 3차원 해쉬 테이블을 사용하여 높은 보안성을 제공하면서, 기존의 2차원 해쉬 테이블을 사용한 방법[20]과 비슷한 방법으로 보호된 지문 템플릿으로부터 지문인식을 수행하는 방법을 설명한다. 먼저, 제안된 방법의 설명을 위해 퍼지볼트에 대해 좀 더 자세히 설명한다. Alice는 비밀키 k를 사용하여 다항식 p를 생성하고 라킹셋(locking set) L을 구성한다. Bob은 언라킹셋(unlocking set) U와 라킹셋 L의 유사도를 비교하여 매칭되는 특징점 쌍을 선택한다. U와 L에서 매칭되는 특징점들을 이용하여 다항식 p를 복원한 후 Alice의 비밀키 k를 획득한다. 라킹셋을 구성하는 과정은 다음과 같다. 보안성 등을 고려하여 차수를 정하고 비밀키 k를 계수로 사용하여 d차 다항식 p를 생성한다. 특히, 특징점의 $0 \leq$ x, y ≤ 255좌표를 다항식의 변수로 사용하며, z좌표는 대량의 거짓 특징점을 삽입하기 위해, θ와 *type*은 오직 지문 템플릿의 정렬을 위해서 사용하고, 다항식의 변수 범위를 넘지 않기 위해 유한체 $GF(q=2^{16})$에서의 연산을 사용한다. 다음, $l_i \in GF(q)$인 l_i을 이용하여, d+1개 이상이 되는 $(l_i, f(l_i))$를 구성한다. 그리고 랜덤한 임의의 값 $c_j \in GF(q)$인 c_i을 생성하고, $f(c_j) \neq \beta_j$인 (c_j, β_j)를 구성한다. 생성된 두 셋(set)을 합하여 라킹셋 L을 구성한다. 만약, 유사도가 높은 후보자의 특징점셋 U의 계수가 d+1 이상이라면, 라킹셋 L로부터 d차 다항식을 복원 할 수 있기 때문에, 비밀키 k를 획득 할 수 있다. 더 자세한 내용은 [14,17,18]에 설명되어 있다.

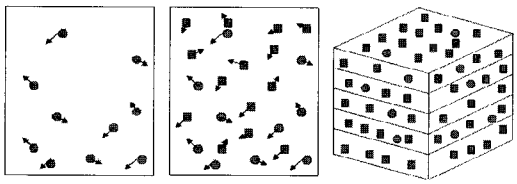
3.1. 3차원 해쉬 테이블 기반 지문 템플릿 생성

본 절에서는 퍼지볼트 이론을 이용하여 안전하게 지문 템플릿을 보호하는 방법을 설명한다. 지문에서 획득한 특징점은 특징점의 위치 및 각도, 그리고 특징점의 종류로 표시할 수 있다. 지문 인식 시스템은 일반적으로 사용자의 지문에서 특징점을 추출한 후 지문 템플릿을

생성하여 파일형태로 저장한다. 하지만 지문 템플릿 공격에 성공할 경우 공격자는 사용자의 지문정보를 도용할 수 있다.

이러한 문제를 해결하기 위해 제안된 지문 퍼지볼트 시스템에서는 지문 템플릿을 보호하기 위하여 퍼지볼트 이론을 이용하여 지문 템플릿에 다수의 거짓 특징점을 생성하여 삽입한다[17,18]. 이렇게 생성된 지문 템플릿은 복원되지 않은 상태에서 사용자의 지문인식 과정을 수행할 수 있고, 보호된 지문 템플릿은 도난당하여도 공격자가 거짓 특징점이 추가된 지문 템플릿으로부터 사용자의 지문 특징점을 분리할 수 없다.

그러나, 지문 퍼지볼트의 기존 연구에서는 추가되는 거짓 특징점의 개수를 최대 200개로 고정하여 지문 템플릿을 보호하였다. 일반적으로 지문 템플릿에 삽입되는 거짓 특징점의 개수가 많아질수록 지문 인식률이 떨어지기 때문에 거짓 특징점을 추가하는데 한계가 있다. 하지만, 날로 향상되는 컴퓨터의 성능으로 공격자의 공격능력이 높아질 수 있다는 점을 고려하면, 더 높은 보안성을 위해 더 많은 거짓 특징점을 추가할 수 있는 방법이 필요하다. 기존 연구와 같이 지문센서에서 획득되는 지문영상에 단순히 거짓 특징점의 개수를 늘려 삽입한다면, 지문영상은 2차원에서 고정되어 있기 때문에 사용자의 특징점과 거짓 특징점들이 매우 밀집하게 되므로 지문 인식률의 성능이 크게 떨어진다는 문제가 있다. 따라서, 본 논문에서는 3차원 해쉬 테이블을 사용하여 거짓 특징점을 지문 템플릿에 삽입함으로써 추가되는 거짓 특징점의 수를 획기적으로 개선하는 방법을 제안한다. [그림 3]에서는 일반적인 지문 템플릿, 2차원 해쉬 테이블을 이용한 퍼지볼트에 의해 보호된 지문 템플릿, 그리고 본 논문에서 제안한 3차원 해쉬 테이블을 이용한 퍼지볼트에 의해 보호된 지문 템플릿의 모습을 보여준다.



(a) (b) (c)

[그림 3]. 지문 템플릿을 보호하는 예 (a) 일반적인 지문 템플릿, (b) 2차원 해쉬 테이블을 이용한 퍼지볼트에 의해 보호된 지문 템플릿(원: 지문으로부터 추출된 사용자의 특징점, 사각형: 거짓 특징점), (c) 3차원 해쉬 테이블을 이용한 퍼지볼트에 의해 보호된 지문 템플릿(원: 사용자의 특징점, 사각형: 거짓 특징점).

3.1.1. 이동 및 회전 변환과 거짓 특징점 선택

우선, 센서로부터 획득한 지문영상으로부터 거짓 특징점을 삽입하기 위해 3차원 테이블을 생성한다. 일반적으로 지문에서 획득한 특징점은 위치, 각도, 그리고 종류로 나타내며 $m_i = (x_i, y_i, \theta_i, t_i)$로 표현되고, 두 개의 좌표(x축과 y축)를 갖는다. 본 논문에서는 3차원 테이블을 생성하기 위하여 $0 < \theta \leq 360°$인 θ를 활용하여 z축 좌표를 생성한다.

특히, 지문 템플릿은 거짓 특징점이 삽입된 상태에서 지문 인식 과정을 수행하여야 한다. 지문 인식률을 고려하여 본 논문에서는 거짓 특징점을 삽입하는 위치와 각도를 고려한다. 가령, 사용자의 특징점과 거짓 특징점의 거리가 너무 가까우면 정당한 사용자 역시 특징점과 거짓 특징점을 구별할 수 없어 인식률이 떨어진다. 따라서 거짓 특징점이 삽입되는 최소한의 거리를 $\triangle d$로 정의하고, 두 특징점 f와 f'로부터 x좌표와 x'좌표의 거리를 $\triangle d1$, y좌표와 y'좌표의 거리를 $\triangle d2$, z좌표와 z좌표의 거리를$\triangle d3$로 거짓 특징점이 삽입 가능한 거리를 정의한다. 거짓 특징점이 삽입 될 때는 사용자의 특징점과의 정의된 거리를 고려하고, 또한 처음 삽입된 거짓 특징점과 나중 삽입되는 거짓 특징점과의 거리 역시 고려한다.

3.1.2. 등록 과정

등록 과정은 지문의 특징점 정보를 획득하고 특징점의 기하학적인 특성을 고려하여 3차원 테이블을 생성한다. 이를 "등록 3차원 테이블"이라 하고 이렇게 생성된 등록 3차원 테이블은 지문 데이터베이스에 저장된다.

특징점 정보 획득 단계는 등록 사용자의 지문영상으로부터 추출된 특징점들을 획득하는 단계이다. 특징점은 위치, 각도, 그리고 종류로 나타내진다. 본 논문에서는 3차원 테이블을 생성하기 위해 θ를 사용하여 z축을 생성하고($z_i = \theta_i/\alpha$), 특징점 정보는 $m_i = (x_i, y_i, z_i, \theta_i, t_i)$로 표현된다. α는 z축의 범위를 결정하는 변수이며 z축의 범위와 메모리 사용량은 정비례 관계를 갖는다. 또한, 특징점의 x좌표와 y표는 라킹셋을 구성하는 정보로 활용되며, z좌표는 대량의 거짓 특

징점을 삽입하기 위해 생성되고, θ와 $type$은 지문정합을 위한 정보로 활용된다.

등록 사용자는 특징점들의 집합으로 표현된다. 거짓 특징점과 사용자의 특징점이 포함된 라킹셋을 $L=\{m_i | 1 \le i \le n\}$으로 표현 할 수 있다. L에서 사용자의 특징점과 거짓 특징점들의 집합은 각각 $G=\{m_i | 1 \le i \le n\}$, $C=\{m_i | n+1 \le i \le r\}$으로 표현한다. 등록 특징점은 L에서 만들어 지며, 3차원 등록 테이블 생성 단계의 각 과정은 다음과 같다.

1) 기준점 선정 단계

등록 사용자의 특징점들의 집합으로부터 첫 번째 특징점인 m_1을 선정한다.

2) 특징점 변환 단계

특징점 변환 단계는 선정된 특징점 m_1을 제외한 다른 특징점 m_2, m_3, …, m_n에 대하여 변환 특징점을 구하는 과정이다. T_1은 m_1을 기준으로 변환된 특징점인 $m_{j(1)}$의 집합이며, $m_{j(1)}$는 m_1을 기준으로 변환된 j번째 특징점을 나타낸다. 이를 "m_1-변환 특징점 집합"이라고 하며, $T_1=\{ m_{j(1)}=(x_{j(1)},y_{j(1)},z_{j(1)},\theta_{j(1)},t_{j(1)} | 1<j \le r)\}$ 으로 표현된다. (식 1)은 기준점 m_1의$(x_1,y_1,z_1,\theta_1,t_1)$에 의하여$(x_{j(1)},y_{j(1)},z_{j(1)},\theta_{j(1)},t_{j(1)})$로 변환되도록 모든 특징점들에 대하여 이동 및 회전 변환를 수행한 결과를 나타낸다. 여기서 ${}_{TR}m_j(1)$은 m_1과 관련된 j번째 특징점을 나타낸다.

$$ {}_{TR}m_j(1)=\begin{pmatrix} {}_{TR}x_j(1) \\ {}_{TR}y_j(1) \\ {}_{TR}z_j(1) \\ {}_{TR}\theta_j(1) \\ {}_{TR}t_j(1) \end{pmatrix}=\begin{pmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 & 0 & 0 \\ -\sin(\theta_1) & \cos(\theta_1) & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} x_j-x_1 \\ y_j-y_1 \\ \theta_j/\alpha \\ \theta_j \\ t_j \end{pmatrix}, where\ 1<j \le r \quad (1) $$

3) 반복 단계

위의 기준점 선정 단계와 특징점 변환 단계는 첫 번째 특징점인 m_1에 대하여 수행한 것이며, 동일한 과정을 m_2, m_3, …, m_n의 모든 특징점들에 대해서도 반복 수행하여 3차원 등록 해쉬 테이블을 생성한다.

3.2. 인식 과정

인식 과정은 특징점 정보 획득 단계, 테이블 생성 단계, 정합 단계, 그리고 후보자 목록 생성 단계를 순차적으로 수행한다. 특징점 정보 획득 단계, 테이블 생성 단계는 등록 과정과 동일한 순서 및 방법으로 수행한다.

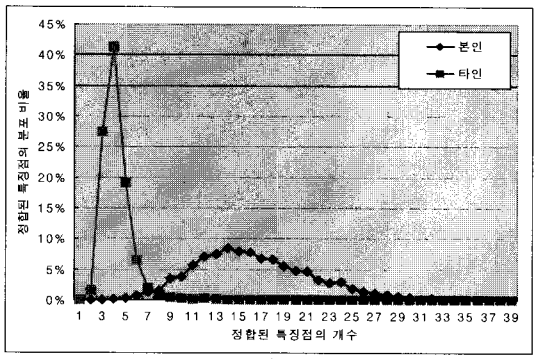
정합 단계는 변환 특징점 집합을 기본 단위로 하여 위치와 각도가 일치하는 특징점 쌍을 구하는 단계이다. 인식 테이블의 변환 특징점들 집합과 지문 데이터베이스에 저장된 등록 테이블의 변환 특징점들 집합 사이에 일치하는 변환 특징점들의 쌍의 수를 구함으로써 측정이 가능하다. 이렇게 만들어진 특징점 셋은 언라킹셋 U가 된다.

후보자 목록은 인식 테이블의 모든 변환 특징점들 집합에 대하여 지문 데이터베이스에 저장된 등록 테이블의 모든 변환 특징점들 집합과 비교하여 유사도를 측정하고, 높은 유사도 순으로 후보자 목록을 구한다. 만약, 유사도가 높은 후보자의 특징점셋 U의 계수가 d+1 이상이라면, d차 다항식을 복원할 수 있다. 이것은 라킹셋 L과 언라킹셋 U가 같은 지문이라는 결과를 의미하며, 공격자는 라킹셋으로부터 사용자의 특징점만을 분리 할 수 없기 때문에, 전사공격 없이 올바른 다항식을 복원할 수 없다.

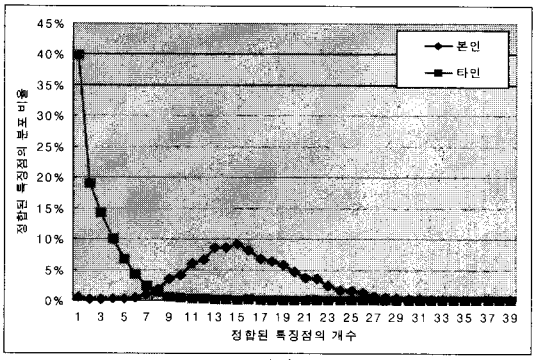
Ⅳ. 구현 및 실험 결과

본 논문에서는 실험을 위해 C언어로 프로그램을 구현하였으며, 광학 센서를 사용하여 100개의 손가락으로부터 한 개의 손가락당 네 개의 지문으로 구성된 400개의 지문 이미지[23]를 사용한다. 센서의 해상도는 500dpi이고 지문 이미지의 크기는 248×292이다. 지문 특징점의 개수는 최대 90개 이고, 최소 16개였으며 평균적으로 39개였다. 또한, 3차원 해쉬 테이블을 생성할 때 α를 6(z=60)으로 설정하였으며, 정합할 때의 특징점 허용 범위는 ±4 pixel이었다.

일반적으로 지문 인식률은 본인과 본인의 정합에서의 FRR(False Reject Rate)과 본인과 타인의 정합에서의 FAR(False Accept Rate)로 나타낼 수 있으며 퍼지 볼트 기반의 지문 인식률은 다항식을 복원할 수 있는 능력에 의존하기 때문에 다항식 차수의 설정이 필요하다. [그림 4]는 2차원 템플릿과 3차원 템플릿으로부터, 기하학적 해싱기법을 적용한 지문의 정합과정을 거쳐 생성된 본인 및 타인의 정합된 특징점의 분포를 보여주고 있다. 다항식 차수를 높게 설정 할수록 FRR은 높아지고 FAR은 낮아지며, 다항식 차수를 낮게 설정하면



(a)



(b)

〔그림 4〕. 본인 및 타인의 정합. (a) 2차원 해쉬 테이블을 이용한 본인 및 타인 정합(거짓 특징점 200개). (b) 3차원 해쉬 테이블을 이용한 본인 및 타인 정합(거짓 특징점 1,000개).

FRR은 낮아지고 FAR은 높아진다. 3차원 해쉬 테이블을 이용한 방법은 1,000개의 거짓 특징점을 삽입하였음에도 불구하고 2차원 해쉬 테이블을 이용한 방법과 에러의 정도가 유사하고 정렬과정을 수행할 수 있음을 확인 할 수 있다.

퍼지볼트의 인식률은 거짓 특징점의 삽입 개수에 따른 정합 결과, 다항식 차수의 설정, RS-code의 복원 능력에 따라 결정된다. 본 논문에서는 정합 과정으로부터 생성된 언라킹셋에 포함된 거짓 특징점을 삭제하기 위한 ECC(Error Correct Code)가 적용된 인식률을 측정하지 못하였다. 이것은 정합 결과로부터 생성된 언라킹셋에 포함되는 거짓 특징점을 삭제하는 RS-code의 복원 능력에 따라 인식률이 결정되기 때문이다. 따라서 본 논문에서는 인식률 대신 지문 템플릿에 삽입되는 거짓 특징점의 개수에 따른 정합 결과를 측정하였다. 이는, 본인정합에서 보호된 템플릿으로부터 정합 결과에 따른 언라킹셋으로 다항식을 풀 수 있기 때문이다. 가령, 등록 시 비밀키를 12차 다항식을 이용하여 저장하였다면 계수가 13개가 되고, 정합을 수행할 때 13개의 사용자 특징점을 획득한다면 12차 다항식을 재생성할 수 있다. 이때, 정합결과로부터 사용자 특징점 13개 이외에 거짓 특징점이 소수 포함되어 있다면 기존의 퍼지볼트[14]와 같이 Peterson-Berlekamp-Massey의 RS-code를 적용하여 거짓 특징점을 삭제함으로써 다항식을 재생성 할 수 있을 것이다.

또한, 일반적으로 지문 템플릿에 삽입되는 거짓 특징점의 개수가 많아질수록 인식률은 떨어진다[19]. 따라서 지문 템플릿에 삽입되는 가짜 특징점의 개수에 따라 정합되는 지문의 진짜와 거짓 특징점의 평균 개수를 측정하였다. 2차원 해쉬 테이블을 이용한 퍼지볼트의 성능을 측정하기 위하여 거짓 특징점의 개수를 100개, 200개, 400개에 따라 측정하였고, 3차원 해쉬 테이블을 이용한 퍼지볼트의 성능을 측정하기 위하여 거짓 특징점의 개수를 100개, 400개, 1,000개에 따라 측정하였다. 특히, 2차원 해쉬 테이블을 이용할 경우 거짓 특징점을 400개까지 삽입 할 수 있었고, 3차원 해쉬 테이블을 이용할 경우 400개 이상까지도 삽입할 수 있었다.

[표 1]에 3차원 해쉬 테이블을 이용하여 거짓 특징점이 1,000개 삽입된 템플릿의 정합 결과와, 2차원 해쉬 테이블을 이용하여 거짓 특징점 200개 삽입된 템플릿의 정합 결과를 나태내었다. 본인 정합은 진짜 특징점이 평균 15개, 거짓 특징점이 1개로 유사한 FRR을 기대할 수 있으며, 타인 정합에서는 3차원 해쉬 테이블을 이용하였을 경우 진짜 특징점은 평균 2개, 거짓 특징점은 평균 6개로 2차원 해쉬 테이블을 이용한 3개와 4개보다 낮은 FAR을 기대할 수 있다. 따라서 3차원 해쉬 테이블을 이용하여 정렬하는 방법은 2차원 해쉬 테이블을 이용하여 정렬하는 방법에 비해 성능이 떨어지지 않고 전체적으로 인식성능이 유사함을 확인 할 수 있다.

또한, 정합 성능은 본 논문에서 제안한 $z=360/\alpha$의 변화에 따라 달라질 수 있다. [표 2]에서는 거짓 특징점이 1,000개가 삽입된 템플릿에서 α를 4, 6, 8로 설정하여 z의 범위가 45, 60, 90인 경우에 대한 정합결과를 보여준다. α를 6(z=60)으로 설정하였을 때의 결과가 α를 8(z=45)이나 4(z=90)으로 설정하였을 경우보다 정합 결과가 좋았음을 확인하였다. 따라서 본 논문에서는 α를 6(z=60)으로 설정하였다. 또한, z의 값이 증가함에 따라 해쉬 테이블 크기가 증가하여 등록시간이 증가하지만, 인식 시간에는 변화가 없음을 확인하였다.

〔표 1〕. 차수와 거짓 특징점 개수에 따른 성능 비교

특징점 평균 : 39	2차원 해쉬 테이블								3차원 해쉬 테이블							
	0개		100개		**200개**		400개		0개		100개		400개		**1,000개**	
	본인	타인	본인	타인	**본인**	**타인**	본인	타인	본인	타인	본인	타인	본인	타인	**본인**	**타인**
정합된 특징점의 평균	15	3	16	5	**16**	**7**	18	11	15	4	15	4	16	4	**16**	**8**
정합된 진짜 특징점의 평균	·	·	15	3	**15**	**3**	14	3	·	·	15	3	15	3	**15**	**2**
정합된 가짜 특징점의 평균	·	·	1	2	**1**	**4**	4	8	·	·	0	1	1	1	**1**	**6**

〔표 2〕. $z=360/\alpha$에 따른 정합 성능(거짓 특징점 1,000개)

특징점 평균 : 39	3차원 해쉬 테이블					
	z=45		z=60		z=90	
	본인	타인	본인	타인	본인	타인
정합된 특징점의 평균	16	9	16	8	16	7
정합된 진짜 특징점의 평균	14	2	15	2	15	2
정합된 가짜 특징점의 평균	2	7	1	6	1	5
등록시간	4.2초		4.7초		5.7초	
정합시간	0.09초					

〔표 3〕. 보안성 비교

	2차원 해쉬 테이블	3차원 해쉬 테이블
거짓 특징점 개수	400	1,000
10차 다항식	3.974×10^{12}	5.833×10^{16}
12차 다항식	1.193×10^{15}	1.02×10^{20}

본 논문에서 제안한 방법의 보안성을 Uludag의 실험[18]에서와 같이 수학적으로 분석하였다. 만약 10차 다항식과 11개의 계수를 사용할 경우, 공격자는 11개 이상의 특징점이 있어야 올바른 다항식을 복원 할 수 있다. 전체 볼트의 수는 1,036개(사용자 특징점 36개, 거짓 특징점 1,000개)이고, 11개의 전체 조합은 C(1036,11) $\approx 3.504\times10^{25}$이다. 또한, 다항식을 풀기 위해 본 논문에서 제안한 방법의 보안성은 C(1036,11)/C(36,11) $\approx 5.8335\times10^{16}$이 된다. 이에 비해, 2차원 해싱 테이블을 이용한 퍼지볼트의 경우, 같은 방법으로 전체 볼트의 수는 436개(사용자 특징점 36개, 거짓 특징점 400개)이고, 10차 다항식을 사용할 경우의 보안성은 C(436,11)/C(36,11) $\approx 3.974\times10^{12}$이다. [표 3]은 2차원 해쉬 테이블을 이용한 퍼지볼트(거짓 특징점 400개)와 3차원 해쉬 테이블을 이용한 퍼지볼트(거짓 특징점 1,000개)의 보안성을 비교하여 보여준다.

마지막으로, 3차원 해쉬 테이블에 1,000개의 거짓 특징점을 추가하는 경우, 등록 시간이 약 4.7초, 정합 시간이 약 0.10초로, 총 4.8초 정도 소요되었다. 그러나, 등록 시간은 오프라인에서 단 한 차례 수행되기 때문에 실질적으로 사용자가 인증하는 시간은 1초 미만의 실시간 처리가 가능하다.

V. 결 론

바이오정보를 이용한 사용자 인증 시스템은 많은 장점을 가지고 있다. 그러나, 바이오정보를 보호하지 않고 사용하였을 경우, 도용당하거나 유출되었을 때 심각한 문제가 제기 될 수 있으며, 바이오정보 획득시마다 상이한 입력 패턴으로 기존의 암호학적 해쉬 함수 적용이 불가능하였다. 본 논문에서는 최근 지문 템플릿을 보호하기 위한 암호학적 방법으로 연구되어지고 있는 퍼지볼트 이론을 단순 적용할 경우 기준점 부재로 인하여 야기되는 지문 정렬 문제점을 해결하기 위하여 기하학적 해싱방법을 사용하였다. 특히, 3차원 해쉬 테이블을 제안하여 거짓 특징점의 개수를 1,000개까지 삽입함으로써 보안성을 향상시켰다.

제안한 방법의 성능을 평가하기 위해서 실제 지문 데이터를 사용하여 실험하였고, 실험을 통하여 제안한 3차원 지문 퍼지볼트 기법이 추가적인 정보 없이 역변환이 불가능한 변환된 영역 상에서 자동으로 지문 정렬을 수행가능하다는 것을 확인하였다.

현재 효율적인 수행 시간과 메모리 사용, 그리고 높은 지문 인식률을 보장하기 위한 연구를 진행 중이다.

참고문헌

[1] W. Stallings, Cryptography and Network Security, Pearson Ed. Inc., 2003.

[2] D. Maltoni, et al., Handbook of Fingerprint Recognition, Springer, 2003.

[3] R. Bolle, J. Connell, and N. Ratha, "Biometric Perils and Patches," Pattern Recognition, Vol. 35, pp. 2727-2738, 2002.

[4] B. Schneier, "The Uses and Abuses of Biometrics," Communications of the ACM, Vol. 42, No, 8, pp. 136, 1999.

[5] S. Prabhakar, S. Pankanti, and A. Jain, "Biometric Recognition: Security and Privacy Concerns," IEEE Security and Privacy, pp. 33-42, 2003.

[6] U. Uludag, et al., "Biometric Cryptosystems: Issues and Challenges," Proc. of IEEE, Vol. 92, No. 6, pp. 948-960, 2004.

[7] V. Matyas and Z. Riha, "Biometric Authentication Systems," TR Ecom- Monitor. Com, 2000.

[8] D. Maio and D. Maltoni, "A Secure Protocol for Electronic Commerce based on Fingerprints and Encryption," Proc. of Conf. on Systems, Cybernetics, and Informatics, pp. 519-525, 1999.

[9] G. Davida, Y. Frankel, and B. Matt, "On Enabling Secure Applications through Off-Line Biometric Identification," Proc. of Symp. on Privacy and Security, pp. 148-157, 1998.

[10] F. Monrose, M. Reiter, and S. Wetzel, "Password Hardening based on Keystroke Dynamics," Proc. of ACM Conf. on Computer and Comm. Security, pp. 73-82, 1999.

[11] N. Ratha, J. Connel, and R. Bolle, "Enhancing Security and Privacy in Biometrics-based Authentication Systems," IBM Systems Journal, Vol. 40, No. 3, pp. 614-634, 2001.

[12] C. Soutar, et al., "Biometric Encryption - Enrollment and Verification Procedures," Proc. SPIE, Vol. 3386, pp. 24-35, 1998.

[13] A. Juels and M. Wattenberg, "A Fuzzy Commitment Scheme," Proc. of ACM Conf. on Computer and Comm. Security, pp. 28-36, 1999.

[14] A. Juels and M. Sudan, "A Fuzzy Vault Scheme," Proc. of Symp. on Information Theory, pp. 408, 2002.

[15] J. Linnartz and P. Tuyls, "New Shielding Functions to Enhance Privacy and Prevent Misuse of Biometric Templates," LNCS 2688 - Proc. of AVBPA, pp. 393-402, 2003.

[16] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data," LNCS 3027 - Proc. of EuroCrypt, pp. 523-540, 2004.

[17] T. Clancy, N. Kiyavash, and D. Lin, "Secure Smartcard-based Fingerprint Authentication," Proc. of ACM SIGMM Multim., Biom. Met. & App., pp. 45-52, 2003.

[18] U. Uludag, S. Pankanti, and A. Jain, "Fuzzy Vault for Fingerprints," LNCS 3546 - Proc. of AVBPA, pp. 310-319, 2005.

[19] S. Yang and I. Verbauwhede, "Secure Fuzzy Vault based fingerprint verification system," IEEE Signals, Systems and Computers, Vol. 1, pp. 557-581, 2004.

[20] Y. Chung, et al., "Automatic Alignment of Fingerprint Features for Fuzzy Fingerprint Vault," LNCS - 3822 Proc. of CISC, pp. 358-369, 2005.

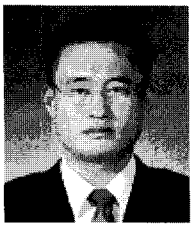
[21] H. Wolfson and I. Rigoutsos, "Geometric Hashing: an Overview," IEEE Computational Science and Engineering, Vol. 4, pp. 10-21, Oct.-Dec. 1997.

[22] C. Ellison, et al., "Protecting Secret Keys with Personal Entropy," Future Generation Computer Systems, Vol. 16, pp. 311-318, 2000.

[23] D. Ahn, et al., "Specification of ETRI Fingerprint Database(in Korean)," Technical Report - ETRI, 2002.

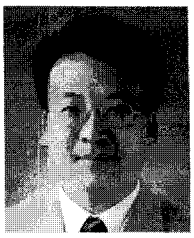
〈著者紹介〉

이 성 주 (Sung-ju Lee) 학생회원
2006년 2월 : 고려대학교 전산학과 학사
2007년 3월~현재 : 고려대학교 전산학과 석사과정
<관심분야> 생체인식, 정보보호



문 대 성 (Dae-sung Moon) 정회원
1999년 2월 : 인제대학교 전산학과 학사
2002년 2월 : 부산대학교 컴퓨터공학과 석사
2007년 2월 : 고려대학교 전산학과 박사
2002년~현재 : 한국전자통신연구원 정보보호연구본부 생체인식기술연구팀 선임연구원
<관심분야> 생체인식, 정보보호, 영상처리

김 학 재 (Hak-jae Kim) 학생회원
2007년 2월 : 고려대학교 전산학과 학사
2007년 3월~현재 : 고려대학교 전산학과 석사과정
<관심분야> 정보보호, 병렬구조



정 용 화 (Yong-wha Chung) 종신회원
1984년 : 한양대학교 전자통신공학과 학사
1986년 : 한양대학교 전자통신공학과 석사
1997년 : 미국 Univ. of Southern California 전기공학과(컴퓨터공학 전공) 박사
1986년~2003년 : 한국전자통신연구원 생체인식기술연구팀장
2003년~현재 : 고려대학교 컴퓨터정보학과 부교수
<관심분야> 생체인식, 정보보호, 생체정보 보호

이 옥 연 (Ok-yeon Yi) 종신회원
1988년 : 고려대학교 수학과 학사
1990년 : 고려대학교 수학과 석사
1996년 : University of Kentucky 수학과 박사
1999년~2001년 : 한국전자통신연구원 선임연구원, 팀장
2001년~현재 : 국민대학교 수학과 부교수
<관심분야> 정보보호, 이동통신, 암호론