情報保護學會論文誌
第19卷 第1號, 2009. 2

비밀분산 기법을 이용한 보안토큰 기반 지문 퍼지볼트의 보안성 향상 방법*

최 한 나,[1†] 이 성 주,[1] 문 대 성,[2] 최 우 용,[2] 정 용 화,[1‡] 반 성 범[3]

[1]고려대학교 컴퓨터정보학과, [2]한국전자통신연구원 바이오인식연구팀,
[3]조선대학교 제어계측로봇공학과

Improved Security for Fuzzy Fingerprint Vault Using Secret Sharing over a Security Token and a Server*

Hanna Choi,[1†] Sungju Lee,[1] Daesung Moon,[2] Woo-Yong Choi,[2] Yongwha Chung,[1‡] Sung Bum Pan[3]

[1]Department of Computer and Information Science, Korea University, [2]Biometrics Technology Research Team, ETRI,
[3]Department of Control, Instrument and Robot Engineering, Chosun University

요 약

보안토큰 기반의 사용자 인증 시스템에서 사용자의 지문정보를 이용하는 방법이 대두되고 있다. 그러나 지문정보가 타인에게 도용된다면 패스워드와 달리 변경이 불가능하거나 제한적이기 때문에 사용자의 지문정보는 더욱 안전하게 보관되어야 한다. 이러한 문제를 해결하기 위한 방법 중 하나로 지문 퍼지볼트(Fuzzy Fingerprint Vault)가 보고되었다. 본 논문에서는 비밀분산 기법을 이용하여 지문 인식률의 성능저하 없이 보안토큰 기반 지문 퍼지볼트 시스템의 보안성을 향상시키는 방법을 제안한다. 즉, 퍼지볼트 이론이 적용된 사용자의 지문 템플릿을 지문 인식률과 보안성을 모두 고려하여 두 부분으로 나누어 각각 보안토큰과 서버에 저장한다. 또한, 퍼지볼트 이론을 지문에 적용하였을 때 발생하는 자동 정렬(Auto-Alignment) 문제는 기하학적 해싱 방법을 분산 적용하여 해결한다. 실험을 통하여 지문 인식률의 성능저하는 무시할 수준이고 보안성은 향상됨을 확인하였다.

ABSTRACT

Recently, in the security token based authentication system, there is an increasing trend of using fingerprint for the token holder verification, instead of passwords. However, the security of the fingerprint data is particularly important as the possible compromise of the data will be permanent. In this paper, we propose an approach for secure fingerprint verification by distributing both the secret and the computation based on the fuzzy vault(a cryptographic construct which has been proposed for crypto-biometric systems). That is, a user fingerprint template which is applied to the fuzzy vault is divided into two parts, and each part is stored into a security token and a server, respectively. At distributing the fingerprint template, we consider both the security level and the verification accuracy. Then, the geometric hashing technique is applied to solve the fingerprint alignment problem, and this computation is also distributed over the combination of the security token and the server in the form of the challenge-response. Finally, the polynomial can be reconstructed from the accumulated real points from both the security token and the server. Based on the experimental results, we confirm that our proposed approach can perform the fuzzy vault-based fingerprint verification more securely on a combination of a security token and a server without significant degradation of the verification accuracy.

Keywords : Security Token, Fingerprint Recognition, Fuzzy Vault, Secret Sharing

Ⅰ. 서 론

최근, 사용자 인증 시스템에서는 사용자의 지문정보를 이용하는 방법이 대두되고 있다. 그러나 지문정보가 타인에게 도용된다면 패스워드와 달리 변경이 불가능하거나 제한적이기 때문에 사용자의 지문정보는 더욱 안전하게 보관되어야 한다[1]. Jules와 Sudan[2]은 퍼지 개념을 적용한 퍼지볼트(Fuzzy Vault)라는 암호이론을 통하여 지문정보를 보호할 수 있는 방법을 제안하였다. 즉, 랜덤함수를 통하여 거짓 특징점(Chaff Minutiae)을 생성한 후, 사용자의 지문 특징점(Real Minutiae)과 함께 지문 템플릿을 구성한다. 또한, 지문 퍼지볼트 시스템의 보안성은 지문 템플릿으로부터 지문 특징점과 거짓 특징점을 구분하는 어려움에 기반을 둔다.

이러한 퍼지볼트 이론을 지문에 적용한 연구 결과가 다수 발표되고 있으나[3-7], 지문 센서에서 입력받은 지문영상의 크기가 제한적이기 때문에 거짓 특징점을 삽입할 수 있는 최대 개수가 제한된다는 문제가 있다. 즉, 사용자의 지문으로부터 추출할 수 있는 고정된 범위의 특징점 수에 비하여 상대적으로 과도한 수의 거짓 특징점을 지문 템플릿에 삽입하면 지문 인식률이 급격하게 저하되는 문제가 있다.

본 논문에서는 지문정보를 보호하기 위하여 퍼지볼트 이론을 적용하고, 생성된 지문 템플릿을 비밀 분산(Secret sharing)[8] 방식으로 보안토큰(Security Token)과 서버에 분산 저장하는 방법을 제안한다. 즉, 보안토큰과 서버 중 어느 한 부분의 공격이라도 전체 지문정보를 알아낼 수 없도록 함으로써 더욱 안전하게 지문정보를 보호할 수 있다. 특히, 지문 템플릿을 분산 저장하는 방법에서 지문 특징점의 개수나 신뢰성에 따라 보안성뿐만 아니라 인식률도 영향을 받는다는 점을 고려해야 한다.

또한, 퍼지볼트 이론을 지문에 적용할 경우, 기준점 부재에 따른 자동 정렬 문제가 발생한다. 본 논문에서는 이러한 자동 정렬 문제를 해결하기 위해 기하학적 해싱 기법[5]을 이용한다. 특히, 보안토큰은 서버에 비하여 자원 제약적이기 때문에 보안토큰에서 수행되는 계산량의 일부를 서버에서 수행해야 하는데, 이때에도 보안성과 인식률이 저하되지 않도록 고려해야한다. 본 논문에서는 다자간 안전한 계산(Secure Multi-Party Computation) 프로토콜[9] 개념을 적용하여 이 문제를 해결한다.

본 논문의 구성은 다음과 같다. 2장에서는 보안토큰 기반으로 지문정보를 이용하여 사용자 인증을 하는 방법과 기하학적 해싱 기법에 대하여 설명하고, 3장에서는 본 논문에서 제안한 지문 템플릿의 분산 저장 방법과 서버와 보안토큰에서의 자동 정렬 문제를 해결하기 위한 방법을 설명한다. 4장에서는 보안성 및 성능을 평가하고, 5장에서는 결론을 맺는다.

Ⅱ. 연구 배경

2.1 보안토큰 기반의 지문 인증 방법

최근, 보안토큰 기반의 인증 시스템에서는 패스워드 대신에 사용자의 지문정보를 이용하는 인증 방법이 많이 사용되어진다. 그러나 보안토큰의 분실이나 위조로 인해 지문정보가 유출되어 악용될 수 있다. 따라서 보안토큰의 지문정보가 유출되는 것을 막고, 보안토큰의 지문정보만으로는 사용자 인증을 할 수 없도록 하는 지문정보 보호방법이 필요하다. 즉, 지문정보를 보안토큰과 서버에 나누어 저장함으로써 전체 지문정보의 유출을 막고, 사용자 인증단계에서는 보안토큰의 지문정보와 사용자의 입력지문, 그리고 서버의 지문정보 모두를 이용하는 방법이 필요하다. 본 논문에서는 퍼지볼트 이론의 바탕이 되는 비밀 분산[8] 개념을 확대 적용하여 보안토큰의 지문정보를 보호한다. 즉, 비밀 정보인 지문정보를 보안토큰과 서버에 분산 저장하는 경우, 공격자가 사용자의 지문정보를 알기 위해서는 보안토큰과 서버에 저장된 각각의 지문정보를 모두 획득해야한다.

2.2 자동 정렬 문제를 해결하기 위한 기하학적 해싱 기법

지문정보를 보호하기 위하여 퍼지볼트 이론을 지문에 적용할 경우, 기준점 부재로 인하여 자동 정렬 문제가 발생한다. 기하학적 해싱 기법은 지문 템플릿을 구성하는 사용자의 특징점, 거짓 특징점 모두를 기준점으로 선택하여 자동 정렬하는 방법이다. 그러나, 기하학적 해

접수일(2008년 9월 4일), 수정일(2008년 12월 16일), 게재확정일(2009년 1월 22일)
* 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT 연구센터 (홈네트워크연구센터) 육성・지원사업의 연구결과로 수행되었음
† 주저자, seracle7@korea.ac.kr
‡ 교신저자, ychungy@korea.ac.kr

싱 기법은 높은 인식률을 갖지만 모든 특징점에 대한 변환된 특징점 정보를 오프라인(Off-Line)에서 해쉬 테이블로 생성하기 때문에 많은 메모리를 요구한다[5]. 따라서 지문 템플릿을 분산 저장하는 방법에서는 보안토큰과 서버의 메모리, CPU 등의 성능이 서로 다른 점을 고려하여 기하학적 해싱 기법을 적용해야한다. 또한, 다자간 안전한 계산 프로토콜[9] 개념을 적용하여 보안토큰과 서버간 전송되는 정보를 보호한다.

Ⅲ. 제안 알고리즘

본 논문에서는 지문정보를 보호하기 위한 방법으로 보안토큰과 서버를 사용하여 지문정보를 더욱 안전하게 보관할 수 있는 방법을 제안한다. 또한, 자동 정렬 문제를 해결하기 위하여 보안토큰과 서버 각각에 대하여 기하학적 해싱 기법을 적용한다. 특히, 보안토큰은 서버에 비하여 자원이 제약적이기 때문에 보안토큰에서 수행되는 계산량 일부를 서버에서 수행하도록 한다.

3.1 지문 템플릿의 분산 방법

지문정보를 보호하기 위해 퍼지볼트가 적용된 지문 템플릿을 나누어 저장할 경우, 나누어지는 지문 특징점의 개수에 따라 지문 인식률과 보안성이 달라진다. 따라서 지문 인식률과 보안성을 모두 고려하여 지문 템플릿을 나누어 저장하는 방법이 필요하다.

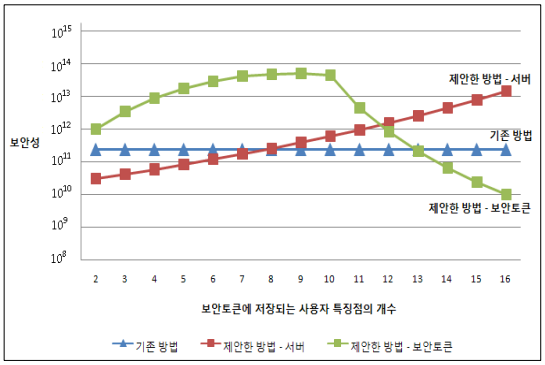
3.1.1 서버와 보안토큰에 저장되는 특징점의 개수

먼저, 지문 템플릿을 분산 저장하기 위해서는 보안토큰에 저장할 특징점의 개수를 결정한다. 사용자 특징점은 퍼지볼트 이론에서 사용된 다항식을 복원하기 위하여 사용되며, 다항식의 차수가 d 이면 다항식을 복원하기 위해서는 $d+1$개의 사용자 특징점이 필요하다. 따라서 보안토큰의 경우, 보안토큰의 사용자 특징점만을 이용하여 다항식을 복원할 수 없도록 하기 위하여 $d+1$개보다 작은 수의 사용자 특징점을 보안토큰에 저장한다.

반면 서버의 경우, 보안토큰에 저장되는 특징점을 제외한 모든 특징점을 저장하기 때문에 평균적으로 $d+1$개보다 많은 수의 사용자 특징점이 저장되어진다. 또한, 기존의 방법과는 다르게 생성된 지문 템플릿을 보안토큰과 서버에 나누어 저장하기 때문에 서버에 저장되는 지문 템플릿을 구성하는 거짓 특징점의 개수가 감소하게 된다. 따라서, 서버에 분산 저장된 지문 템플릿은 분산 전의 보안성을 유지하여야 한다.

이 때, 지문 템플릿의 보안성은 Uludag의 실험[4]을 이용하여 분석할 수 있다. 만약 d차 다항식, a 개의 사용자 특징점, 그리고 b 개의 거짓 특징점을 이용한다면, d 차 다항식을 복원하기 위해서는 $d+1$개의 사용자 특징점이 필요하다. 따라서, 전체 볼트의 수와 $d+1$개의 조합은 $C(a+b, d+1)$이고, 다항식을 풀기 위한 조합은 $C(a, d+1)$이기 때문에 공격자가 다항식을 풀기 위해서는 평균적으로 $C(a+b, d+1)/C(a, d+1)$의 계산이 필요하다.

Uludag의 실험[4]은 본 논문에서 제안한 방법을 서버와 보안토큰에 적용할 수 있다. 예를 들어, [그림 1]과 같이 9차 다항식과 삽입되는 거짓 특징점의 개수가 서버와 보안토큰에 각각 300개, 100개, 총 400개일 때를 살펴보면 보안토큰에 저장되는 특징점의 개수가 8개 이하일 경우, 서버에 분산 저장된 지문 템플릿의 보안성이 분산 전 지문 템플릿의 보안성 보다 낮게 나타난다. 또한 보안토큰에 저장되는 특징점의 개수가 10개 이상일 경우에는 보안토큰의 사용자 특징점으로 다항식을 복원할 수 있기 때문에 보안토큰의 보안성이 저하된다. 기존 방법의 경우, 지문 템플릿을 분산 저장하지 않기 때문에 보안성은 특징점 개수에 의해 고정적으로 결정된다[4]. 따라서, 9차 다항식과 총 400개의 거짓 특징점을 이용할 경우, 보안토큰과 서버 모두의 보안성을 고려하여 사용자 특징점 중 8~10개만을 보안토큰에 저장하면 기존 방법에 비해 높은 보안성을 제공할 수 있다.

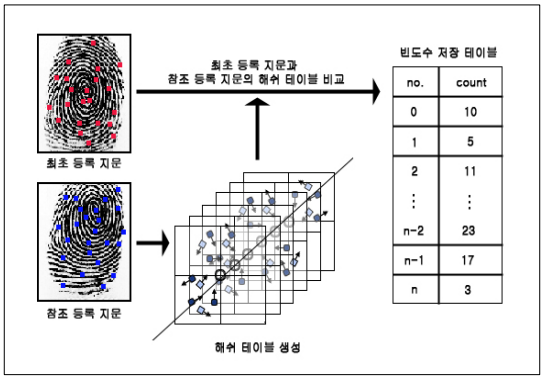


[그림 1] 9차 다항식을 이용한 지문 템플릿의 분산 전과 분산 후의 보안성

3.1.2 서버와 보안토큰에 저장되는 특징점의 신뢰성

사용자 인식률은 저장되는 특징점의 개수와 저장되는 특징점의 신뢰성에 영향을 받는다. 즉, 서버에서의 부분정렬 정보를 결정하는 특징점의 정보가 신뢰성이 낮으면 보안토큰의 특징점 정렬을 올바르게 수행하지 못한다. 특징점의 신뢰성은 사용자가 지문 정보를 입력하였을 때, 추출 가능성이 높은 특징점으로써, 사용자가 지문 정보를 등록하는 단계에서 각 특징점의 추출 빈도수를 이용하여 나타낼 수 있다. 따라서 특징점을 분산 저장하는 방법에서 특징점의 신뢰성을 고려하는 방법이 필요하다.

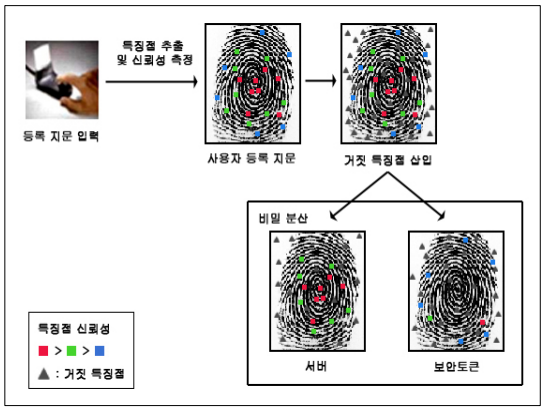
본 논문에서는 특징점의 신뢰성 측정에 필요한 특징점의 추출 빈도수를 계산하기 위하여 사용자 지문 등록 단계에서 등록 지문을 여러번 입력하여 비교하는 방법을 이용한다. 즉, 첫 번째로 입력한 지문은 최초 등록 지문, 이후에 입력한 지문은 참조 등록 지문이라 하면, [그림 2]와 같이 참조 등록 지문의 해쉬 테이블을 생성한 후, 최초 등록 지문과 비교한다. 이 때, 각 특징점의 추출 빈도수를 계산하기 위하여 빈도수 저장 테이블을 생성한 후, 각 특징점이 정합된 횟수를 기록한다.



[그림 2] 특징점의 신뢰성 측정을 위한 빈도수 저장 테이블 생성 과정

이와 같은 과정을 참조 등록 지문의 입력 횟수 만큼 수행하여 사용자 특징점의 빈도수를 계산한 후, 최초 등록 지문과 각 참조 등록 지문의 비교 횟수 및 각 빈도수 저장 테이블에 저장된 값을 이용하여 특징점의 신뢰성을 계산한다.

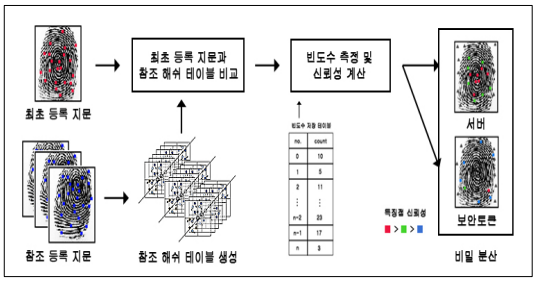
대부분 신뢰성이 높은 특징점은 지문의 중심에 집중되어 있고 추출 빈도수가 상대적으로 높게 나타난다. 따라서 지문 영상의 중심에 위치하는 특징점을 기준점으로 선택하면 올바른 특징점 정렬을 하게 될 가능성이 높아진다. 즉, 지문 중심점 부근에 위치하는 특징점은 다른 위치에 있는 특징점보다 신뢰성이 높으며, 지문 중심점 부근에서 멀어질수록 특징점의 신뢰성이 낮게 나타난다. 따라서 지문 템플릿을 분산 저장할 때 서버가 정확한 부분정렬 정보를 선정할 수 있도록 신뢰성이 높은 특징점 중 지문의 중심에 위치하는 특징점을 우선적으로 서버에 저장한다. 또한, 보안토큰에서는 전체 사용자 특징점과 보안토큰에 저장하는 특징점의 개수를 고려하여 적합한 신뢰성을 갖는 특징점을 저장한다.



[그림 3] 특징점의 신뢰성을 고려한 사용자 지문 특징점의 분산 방법

[그림 3]은 각 특징점의 위치에 따른 신뢰성을 나타내며, 이를 고려한 분산 저장과정이다. 센서를 통해 입력된 등록지문은 특징점 추출과 거짓 특징점의 삽입단계를 통하여 지문 템플릿을 생성한다. 생성된 지문 템플릿은 특징점의 신뢰성과 보안토큰의 보안성을 고려하여 보안토큰과 서버에 분산 저장되어진다.

[그림 4]는 사용자 지문 등록에 대한 순서를 나타낸다. 이 과정을 자세히 살펴보면, 사용자가 입력한 최초 등록 지문과 참조 지문 정보를 입력받은 후, 참조 해쉬 테이블을 생성하여 최초 등록지문과 비교한다. 또한, 특징점 빈도수 테이블을 생성하여 각 특징점의 빈도수를 측정하여 특징점의 신뢰성을 계산한다. 계산된 특징점의 신뢰성은 특징점의 위치와 저장되는 개수 등을 고려하여 서버와 보안토큰에 분산 저장되어진다.



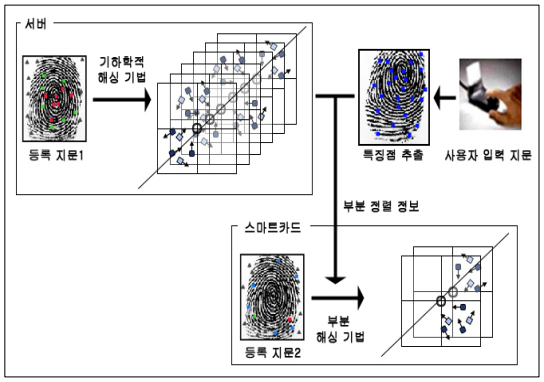
[그림 4] 사용자 지문 등록단계

3.2 보안토큰과 서버에서의 지문 자동 정렬 방법

서버의 특징점 자동 정렬은 사용자 인증 단계에서 이루어진다. 본 논문에서는 기하학적 해싱 기법을 서버와 보안토큰에 각각 적용하여 자동 정렬 문제를 해결하였다. 그러나 기하학적 해싱 기법은 모든 특징점을 기준으로 하여 변환된 특징점 정보를 해쉬 테이블로 생성하기 때문에 많은 메모리를 필요로 한다. 더욱이 보안토큰은 자원 제약적이기 때문에 효율적으로 기하학적 해싱 기법을 적용할 필요가 있다. 따라서 보안토큰에서 자동 정렬 할 때에는 서버에서 이루어진 자동 정렬 정보의 일부인 부분정렬 정보를 이용하여 정렬한다.

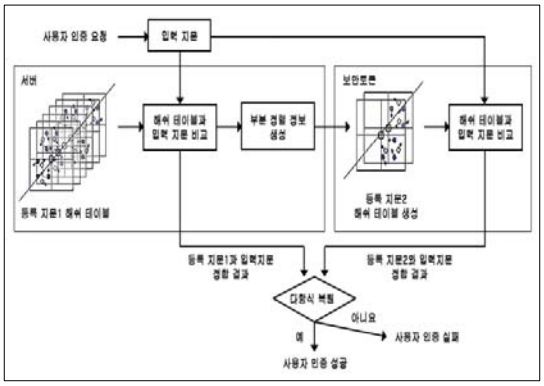
사용자 인증 단계에서의 첫 번째 단계는 서버의 특징점과 입력지문을 비교한다. 즉, 서버의 특징점을 기하학적 해싱 기법을 이용하여 생성된 해시 테이블과 입력지문을 비교한다. 두 번째 단계는 서버에서 부분정렬 정보를 선정한 후, 보안토큰에게 전송하는 단계이다. 부분정렬 정보는 보안토큰의 지문과 입력지문의 올바른 자동 정렬을 위하여 필요하며, 분산 전 지문 정보를 자동 정렬 할 때 기준점으로 선정될 가능성이 높은 지문 특징점 정보이다. 사용자 인증 단계에서 마지막은 보안토큰의 특징점을 자동 정렬 한 후, 입력지문과 비교하는 단계이다. 보안토큰은 자원 제약적인 특성을 갖기 때문에 서버에서 특징점 자동 정렬 방법으로 사용된 기하학적 해싱 기법과는 다른 부분정렬 정보를 이용한 기하학적 해싱 기법을 적용한다. 즉, 보안토큰은 저장된 모든 특징점을 기준으로 사용하여 특징점 정보를 해쉬 테이블로 생성하는 것이 아니라 서버에서 전송받은 올바른 자동 정렬 정보의 가능성이 높은 일부 기준점 정보를 이용하여 해쉬 테이블을 생성하는 것이다.

[그림 5]는 보안토큰과 서버의 지문을 자동 정렬하는 방법을 설명하고 있다. 서버는 서버의 특징점을 자동 정렬하기 위하여 모든 특징점을 기준점으로 하였을 때 변환된 특징점 정보를 해쉬 테이블로 생성한다. 해쉬 테이블로 생성된 정보 중 입력지문과 가장 높은 정합률을 보이는 일부의 기준점 정보를 부분정렬 정보로 생성하여 보안토큰에게 전송한다. 보안토큰은 서버에서의 특징점 자동 정렬 방법과는 다르게 전송받은 부분정렬 정보만을 이용하여 해쉬 테이블을 생성함으로써 보안토큰의 계산량을 줄이고 보안토큰의 특징점 자동 정렬을 수행한다. 여기서, 등록 지문1은 서버에 저장된 사용자의 지문 특징점을, 등록 지문2는 보안토큰에 저장된 사용자의 특징점으로 구성된 지문 정보를 말한다.



[그림 5] 서버와 보안토큰의 자동 정렬 방법

지문 특징점의 자동 정렬은 사용자 인증부에서 수행되어진다. 먼저, 사용자 인증 요청이 들어오면 서버는 저장되어 있는 등록 지문1의 해쉬 테이블과 입력 지문을 비교하고, 부분 정렬 정보를 생성한다. 생성된 부분정렬 정보는 보안토큰으로 전송되어 등록 지문2의 해쉬



[그림 6] 사용자 인증단계

테이블 생성과정에서 사용되어진다. 보안토큰은 등록 지문2의 해쉬 테이블이 생성되면 입력 지문과 비교한다. 사용자 인증 여부는 서버에서의 입력 지문 비교 결과와 보안토큰에서의 입력 지문 비교 결과를 이용하여 다항식 복원 여부에 따라 결정한다. 즉, 다항식 복원을 시도하여 복원이 된다면 사용자 인증에 성공하게 되고, 복원이 되지 않을 경우, 사용자 인증에 실패하게 된다. [그림 6]은 사용자 인증 과정을 나타낸다.

Ⅳ. 실험 결과 및 보안성 분석

본 논문에서는 실험을 위해 FVC2002[10]를 사용하여 40개의 손가락으로부터 한 개의 손가락 당 세 개의 지문으로 구성된 120개의 지문 이미지를 사용한다. 센서의 해상도는 500dpi이고 지문 이미지의 크기는 388×374이다. 거짓 특징점은 랜덤 함수를 이용하여 서버와 보안토큰에 각각 300개, 100개로 총 400개를 생성한다. 평균 특징점 개수는 총 36개이고, 등록 해쉬 테이블의 크기는 256×256이다. 실험 환경으로는 서버로 PC(Pentium4 CPU 2.8GHz, 2GB)를, 보안토큰으로 임베디드 시스템(Xscale CPU 520MHz, 128MB)을 사용하였다.

[표 1]은 본 논문에서 제안한 방법에 대한 보안성을 알아보기 위해서 서버와 보안토큰의 경우로 나누어 계산한 결과이다. 서버의 경우, Uludag의 실험[4]과 같은 방법으로 보안성을 구할 수 있다. 만약 9차 다항식과 10개의 계수를 사용할 경우, 공격자는 10개 이상의 특징점이 있어야 올바른 다항식을 복원할 수 있다. 서버에서 전체 볼트의 수는 326개이고, 10개의 전체 조합은 $C(326,10)\approx 3.35\times10^{18}$이다. 또한, 다항식을 풀기 위한 조합은 $C(26,10)\approx 8.43\times10^{6}$이다. 그러므로 공격자가 다항식을 풀기 위해 평균적으로 3.97×10^{11} ($\approx C(327,10)/C(27,10)$)계산이 필요하다. 이 때, 보안토큰에 저장된 사용자의 특징점 개수는 9개이다.

그러나 보안토큰의 경우, 사용자의 특징점 정보가 다항식을 복원할 때 필요한 특징점의 정보보다 부족하므로 서버에서 하나 이상의 사용자 특징점 정보를 추가적으로 알아내야한다. 따라서 보안토큰의 보안성은 보안토큰에서의 진짜 특징점을 알아내기 위한 평균 계산량($\approx 4.26\times10^{12}$)과 추가적으로 서버에서 특징점 1개를 알아내기 위한 평균 계산량($\approx$12.11)의 곱인 $5.16\times10^{13}(\approx 4.26\times10^{12}\times12.11)$으로 나타낼 수 있다.

[표 2]는 보안성 분석 결과를 이용하여 9차 다항식과 보안토큰에 저장되는 특징점이 9개 일 때, 등록지문과 입력지문의 정합 결과이다. 먼저, 제안 방법의 본인과 타인의 정합결과를 살펴보면 본인의 정합된 사용자 특징점의 평균 개수는 15개이고 거짓 특징점의 평균 개수는 2개이다. 이에 비해, 타인의 실험에서 정합된 사용자 특징점의 평균 개수는 2개이고 거짓 특징점의 개수는 5개이다.

또한, 기존 방법[5]과 제안 방법의 성능을 살펴보기 위하여 FRR(False Rejection Rate)과 FAR(False Acceptance Rate) 성능을 측정하였다. 기존 방법[5]에서는 FRR이 12.1%, FAR 이 0.1%로 측정되고, 제안 방법에서의 FRR은 17.9%, FAR은 0%로 측정되어진다. 따라서 제안 방법이 기존 방법과 비교하였을 때 성능의 큰 차이를 보이지 않는 것을 확인하였다.

[표 1] 다항식 차수에 따른 보안성 비교

	Uludag, et al.[4]	제안 방법	
		서버	보안토큰
9차	2.43×10^{11}	3.97×10^{11}	5.16×10^{13}
10차	3.97×10^{12}	1.21×10^{13}	5.88×10^{14}
11차	6.76×10^{13}	4.54×10^{14}	6.15×10^{15}

[표 2] 기존 방법[5]과 제안 방법의 성능 차이

(a) 기존 방법[5]의 특징점 평균 정합 개수 (단위:개)

	본인	타인
정합된 총 특징점 개수	18	12
정합된 사용자 특징점 개수	14	2
정합된 거짓 특징점 개수	4	10

(b) 제안 방법의 특징점 평균 정합 개수 (단위:개)

	서버		보안토큰	
	본인	타인	본인	타인
정합된 총 특징점 개수	13	5	4	2
정합된 사용자 특징점 개수	12	1	3	1
정합된 거짓 특징점 개수	1	4	1	1

(c) 기존 방법과 제안 방법의 FRR, FAR

	기존 방법[5]	제안한 방법
FRR	0.121	0.179
FAR	0.001	0

마지막으로 기존 방법[5]과 제안 방법의 수행시간을 측정하였다. [표 3]과 같이 지문 특징점의 등록시간은 0.093초(보안토큰 0.028초, 서버 0.065초), 인증시간은 1.806초(보안토큰 0.240초, 서버 1.566초)이다. 따라서 본 논문에서 제안한 방법은 실시간 처리도 가능함을 확인하였다.

[표 3] 지문 등록 및 인증 시간 비교 (단위:초)

	기존 방법[5] (거짓특징점 400개)	제안 방법 (거짓특징점 400개)	
		서버 (거짓특징점300개)	보안토큰 (거짓특징점100개)
등록시간	0.390	0.028	0.065
인증시간	0.101	0.240	1.566

Ⅴ. 결 론

본 논문에서는 퍼지볼트가 적용된 지문 템플릿을 서버와 보안토큰에 분산 저장함으로써 지문정보를 보호하였다. 먼저, 지문정보의 보안성과 인식률을 고려하여 지문 특징점을 나누는 방법을 결정한 후, 나누어진 지문 특징점의 자동 정렬 문제를 해결하기 위하여 기하학적 해싱 기법을 적용하였다. 또한, 보안토큰의 자원 제약적인 특징을 고려하여 대부분의 계산량을 서버에게 할당하였다. 즉, 서버는 먼저 부분정렬을 수행한 후 신뢰성이 높은 기준점 정보인 부분정렬 정보를 보안토큰에게 전송함으로써 보안토큰의 계산량을 줄이면서 지문 자동정렬을 정확하게 수행할 수 있다. 제안 방법에서는 기존의 방법보다 보안성은 평균 서버는 4배, 보안토큰은 150배 향상되었고 인식 성능은 5.8% 하락하였다. 또한, 수행시간은 등록에서 0.093초, 인증에서 1.806초로 실시간적으로 수행할 수 있음을 확인하였다. 본 논문에서는 인식률과 보안성을 분석하기 위하여 실제 지문 이미지를 사용하여 실험하였으며, 실험을 통하여 기존 방법의 인식률을 유지하면서 비밀 분산을 통하여 보안성을 개선할 수 있음을 확인하였다.

참고문헌

[1] U. Uludag, S. Pankanti, S. Prabhakar, and A.K. Jain, “Biometric Cryptosystems: Issues and Challenges,” IEEE, vol. 92, no. 6, pp. 948-960, June 2004.

[2] A. Juels and M. Sudan, “A Fuzzy Vault Scheme,” IEEE International Symposium on Information Theory, pp. 408, Nov. 2002.

[3] T. Clancy, N. Kiyayash, and D. Lin, “Secure Smartcard-based Fingerprint Authentication,” ACM SIGMM workshop on Biometrics methods and applications 2003, pp. 45-52, Nov. 2003.

[4] U. Uludag, S. Pankanti, and A. Jain, “Fuzzy Vault for Fingerprints,” International Conference on Audio- and Video-Based Biometric Person Authentication 2005, LNCS 3546, pp. 310-319, June 2005.

[5] Y. Chung, D. Moon, S. Lee, S. Jung, T. Kim, and D. Ahn, “Automatic Alignment of Fingerprint Features for Fuzzy Fingerprint Vault,” Conference on Information Security and Cryptology 2005, LNCS 3822, pp. 358-369, Dec. 2005.

[6] J. Jeffers and A. Arakala, “Minutiae-based Structures for a Fuzzy Vault,” Biometrics Symposium, pp. 1-6, Aug. 2006.

[7] H. Kikuchi, Y. Onuki, and K. Nagai, “Evaluation and Implement of Fuzzy Vault Scheme using Indexed Minutiae,” IEEE International Conference on Systems, Man and Cybernetics, pp. 3709-3712, Oct. 2007.

[8] A. Shamir, “How to Share a Secret,” Communications of the ACM, vol. 22, no. 11, pp. 612-613, Nov. 1979.

[9] R. Canetti, “Studies in Secure Multi-Party Computation and Applications,” Ph.D. Thesis, The Weizmann Institute of Science, June 1995.

[10] Fingerprint Verification Competition(FVC 2002), http://bias.csr.unibo.it/fvc2002.

<著者紹介>

최 한 나 (Hanna Choi) 학생회원
2008년 2월: 고려대학교 컴퓨터정보학과 학사
2008년 3월~현재: 고려대학교 컴퓨터정보학과 석사 과정
<관심분야> 바이오인식, 정보보호

이 성 주 (Sungju Lee) 학생회원
2006년 2월: 고려대학교 전산학과 학사
2008년 2월: 고려대학교 전산학과 석사
2008년 3월~현재: 고려대학교 컴퓨터정보학과 박사 과정
<관심분야> 바이오인식, 정보보호

문 대 성 (Dae-sung Moon) 정회원
1999년 2월: 인제대학교 전산학과 학사
2001년 2월: 부산대학교 컴퓨터공학과 석사
2007년 2월: 고려대학교 전산학과 박사
2001년~현재 : 한국전자통신연구원 선임연구원
<관심분야> 바이오인식, 정보보호, 영상처리

최 우 용 (Woo Yong Choi) 정회원
1998년 2월: 부산대학교 통계학과 학사
2000년 2월: 부산대학교 전자공학과 석사
2002년 2월~2001년 1월: L&H Korea 연구원
2001년~현재: 한국전자통신연구원 선임연구원
<관심분야> 바이오인식, 정보보호, 영상처리

정 용 화 (Yongwha Chung) 종신회원
1984년: 한양대학교 전자통신공학과 학사
1986년: 한양대학교 전자통신공학과 석사
1997년: 미국 Univ. of Southern California 전기공학과(컴퓨터공학 전공) 박사
1986년~2003년: 한국전자통신연구원 생체인식기술연구팀 팀장
2003년~현재: 고려대학교 컴퓨터정보학과 교수
<관심분야> 바이오인식, 정보보호, 바이오정보보호

반 성 범 (Sung Bum Pan) 종신회원
1991년: 서강대학교 전자공학과 학사
1995년: 서강대학교 전자공학과 석사
1999년: 서강대학교 전자공학과 박사
1999년~2005년: 한국전자통신연구원 정보보호연구본부 생체인식기술연구팀 팀장
2005년~현재: 조선대학교 제어계측로봇공학과 조교수
<관심분야> 바이오인식, 영상처리, VLSI 신호처리