KIPS Tr. Comp. and Comm. Sys.
Vol.5, No.2 pp.41~46 pISSN: 2287-5891

http://dx.doi.org/10.3745/KTCCS.2016.5.2.41

Framework for Secure Biometric System Design on Smartphones

Jong-Hyuk Im† · Hee-Yong Kwon†† · Mun-Kyu Lee†††

ABSTRACT

Fast growth of smartphone technology and advent of Fintech enabled smartphones to deal with more sensitive information. Although many devices applying biometric technology are released as a step for protecting sensitive information securely, there can be potential vulnerabilities if security is not considered at the design stage of a biometric system. By analyzing the potential vulnerabilities, we classify threats in biometric system design process on smartphones and we propose the design requirements for solving these problems. In addition, we propose a framework for secure biometric system design on smartphone by synthesizing the design requirements.

Keywords : Biometrics, Smartphone Security, System Design Requirement, Framework

스마트폰 상의 안전한 바이오인식 시스템 설계를 위한 프레임워크

임 종 혁† · 권 희 용†† · 이 문 규†††

요 약

최근 스마트폰 기술의 빠른 발전과 핀테크의 등장으로 스마트폰은 더욱 많은 민감한 정보를 다루게 되었다. 이 같은 민감한 정보를 안전하게 보호하는 수단으로 바이오인식 기술이 적용된 다양한 기기들이 출시되고 있으나, 바이오인식 시스템 설계 시 보안을 고려하지 않을 경우 잠재적인 취약점이 존재할 수 있다. 이에 본 논문에서는 잠재적인 취약점의 분석을 통해 스마트폰 상의 바이오인식 시스템 설계 과정에서 주의할 점을 분류하고, 이를 해결하기 위한 설계 요구사항을 제시한다. 또한, 설계 요구사항을 종합하여 안전한 스마트폰 바이오인식 시스템 설계를 위한 프레임워크를 제시한다.

키워드 : 바이오인식, 스마트폰 보안, 시스템 설계 요구사항, 프레임워크

1. 서 론

스마트폰은 최근 관련 기술들의 발전으로 인스턴트 메시지, 이메일 등의 기능을 사용하는 도구로써의 역할을 넘어, 핀테크(Fintech)와 같이 민감한 개인 정보를 취급하는 서비스를 제공하는 경우가 늘어났다. 이러한 정보들은 유출되었을 경우 심각한 프라이버시 문제를 일으킬 수 있으므로 스마트폰 상에서 이를 안전하게 보호하기 위한 여러 연구가 진행되고 있으며, 이 중 사용자 인증 기술 또한 중요성이 강조되고 있다. 또한, 최근 대형 스마트폰 제조사들은 제품에 지문인식 기능을 추가하는 등 스마트폰 상의 안전한 사용자 인증을 위한 바이오인식 기술을 하나의 중요한 해결방법으로 사용하고 있다[2].

그러나 바이오인식 기술을 스마트폰에 적용하기 위한 시스템의 설계가 잘못되었을 경우에 이들이 취약점이 되어 스마트폰 상의 신용 정보나 바이오인식 원본 정보와 같은 민감한 개인 정보의 유출이라는 치명적인 문제를 일으킬 수 있다. 본 논문에서는 스마트폰 바이오인식 시스템 설계에서 발생할 수 있는 문제점을 실 사례와 함께 분류하고, 스마트폰 바이오인식 시스템을 안전하게 설계하기 위한 설계 요구사항을 제시한다. 또한, 제시한 설계 요구사항을 통합하여 안전한 스마트폰 바이오인식 시스템 설계를 위한 프레임워크를 제시한다.

2. 바이오인식 기술 및 위협 모델

바이오인식 기술은 보편성, 유일성, 영구성, 획득성 등과 같은 보안 측면에서의 다양한 요구조건을 만족시키는 사용자의 신체적 특성이나 행위적 특성에 기반을 둔다[3]. 대표적으로 지문, 홍채, 얼굴, 정맥, 필체, 목소리 등을 주로 이용하는데, 일반적으로 바이오인식은 등록 단계에서 정상 사용자의 인증

※ 이 논문은 2015년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(No.2014R1A1A2058514)의 결과임.
※ 이 논문은 2015년도 한국정보처리학회 추계학술발표대회에서 '스마트폰 상의 안전한 바이오인식 시스템 설계를 위한 요구사항'의 제목으로 발표된 논문을 확장한 것임.
† 준 회 원 : 인하대학교 컴퓨터정보공학과 박사과정
†† 비 회 원 : 인하대학교 컴퓨터정보공학과 석사과정
††† 정 회 원 : 인하대학교 컴퓨터정보공학과 부교수
Manuscript Received : December 23, 2015
First Revision : February 26, 2016
Accepted : February 26, 2016
* Corresponding Author : Mun-Kyu Lee(mklee@inha.ac.kr)

정보인 템플릿을 저장하고, 인증 시도 시에 바이오인식 센서를 통해 취득한 인증 정보를 비교하는 형태로 동작한다. 또한, 바이오인식 기술의 성능 지표로는 FMR (False Match Rate)와 FNMR(False Nonmatch Rate) 및 이들을 결합한 EER (Equal Error Rate)를 주로 사용하는데, FMR은 서로 다른 두 사람의 바이오인식 데이터가 같은 사람으로 잘못 매치될 비율, FNMR은 같은 사람의 두 바이오인식 데이터가 다른 사람으로 잘못 매치될 비율을 뜻하며 EER은 FMR과 FNMR 곡선이 만나는 지점의 FMR 및 FNMR 값을 뜻한다[3].

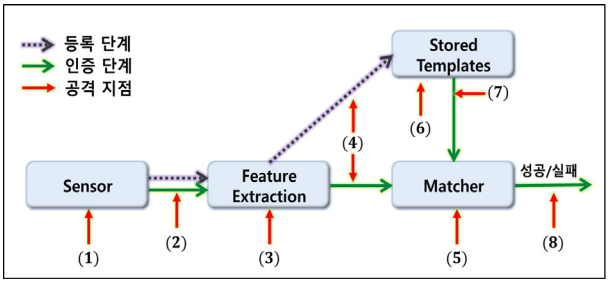

Fig. 1. Biometric process and possible attack points in biometrics system[4]

Fig. 1은 일반적인 바이오인식 시스템의 등록 및 인증 단계를 포함하는 전형적인 바이오인식 절차와, 보안에 대한 고려 없이 설계될 경우 각 단계에서 공격 가능한 지점을 도식화한 위협 모델이다[4]. 먼저, 등록 단계는 바이오인식 센서를 통해 바이오인식 원본 데이터를 획득한 후 특징점을 추출하여 템플릿 형태로 저장하는 단계이다. 다음으로 인증 단계는 인증이 필요한 사용자 어플리케이션으로부터 인증 요청을 받았을 때, 센서로부터 인증하려는 사용자의 바이오인식 데이터를 입력받은 후 특징점을 추출하여 템플릿으로 만든다. 이를 기존 저장된 사용자의 템플릿과 비교하여 인증 성공/실패 여부를 사용자 어플리케이션으로 전달하는 절차로 이루어진다. 다음은 Fig. 1의 각 공격 지점들에서 공격자의 능력에 대한 간략한 설명이다.

(1) 사용자로부터 바이오인식 데이터를 얻는 부분으로, 센서를 기만하기 위해 가짜 지문 등을 이용할 수 있다.
(2) 기기 내에 미리 저장된 바이오인식 원본 데이터를 재사용하여 센서를 우회하고 이를 전송할 수 있다.
(3) 특징점 추출 자체를 조작하는 것으로, 공격자가 임의로 템플릿 생성을 시도할 수 있다.
(4) 특징점 추출 이후 템플릿을 저장하거나 매쳐로 전송하는 채널을 공격하는 경우로, 패킷을 조작하여 (3)과 같이 공격자가 임의로 템플릿을 생성하거나, 정상 템플릿을 탈취할 수 있다.
(5) 매쳐를 직접 공격해 실제 매칭 여부와 관계없이 사전에 결정된 결과가 나오도록 할 수 있다.
(6) 저장된 템플릿을 직접 공격해 공격자가 저장된 템플릿을 획득하거나 조작할 수 있다. 템플릿의 조작의 결과로 FMR이나 FNMR이 높아질 수 있다.
(7) 매쳐로 저장된 템플릿이 전송되는 채널을 공격하는 경우로, 데이터를 가로채 정상 사용자의 템플릿을 획득하거나 이를 조작함으로써 정상적인 매치 결과를 얻을 수 없게 할 수 있다.
(8) 사용자 인증 결과(성공/실패)를 조작할 수 있다.

3. 기존 스마트폰 바이오인식 시스템의 문제점

스마트폰은 개인이 휴대하는 기기이므로 항상 분실 및 도난의 위험성이 있으며, 이때 내부 정보를 쉽게 탈취당할 수 있는 특징을 가지고 있다. 또한, 많은 스마트폰은 바이오인식 단계를 우회할 수 있는 취약점을 가지고 있다. 다음은 스마트폰에서 바이오인식 시스템 설계 시 보안을 고려하지 않았을 때 생길 수 있는 취약점 및 공격 사례들이다.

먼저 스마트폰에서 바이오인식 데이터를 처리하거나 저장하는 단계에서 다양한 문제가 발생할 수 있다. 스마트폰에서 바이오인식 데이터를 등록하거나 인증할 때, 등록이나 인증의 결과에 대한 피드백을 제공하는 경우, 또는 저장된 바이오인식 데이터를 안전하게 관리하지 않는 경우에 사용자의 바이오인식 데이터가 유출될 수 있다. HTC 사의 스마트폰인 One MAX는 사용자의 지문 이미지를 암호화되지 않은 폴더에 저장하여, 권한을 가지지 않는 사용자 또는 앱이 해당 지문 이미지에 접근할 수 있는 취약점을 가진다[5].

그러나 바이오인식 데이터가 유출되었을 경우를 대비하여 바이오인식 데이터를 암호화하더라도, 암호화키의 생성 및 관리가 적절하게 이루어지지 않거나, 암호화키가 안전하게 적용되지 않아 암호화하지 않은 것과 같은 취약한 상태가 되는 경우가 있으며, 이를 이용해 실제로 공격이 성공한 사례가 있다[6].

마지막으로, 바이오인식을 통한 사용자 인증 루틴 자체를 우회하는 경우가 있다. 바이오인식을 통한 사용자 인증 루틴 우회에는 바이오인식 결과를 조작하는 경우, 바이오인식 센서를 기만하는 경우, 또는 바이오인식 매쳐를 기만하는 경우가 있는데, 이러한 경우들에는 바이오인식 시스템이 무력화되기 때문에 치명적인 취약점이 될 수 있다. 예를 들어 iPhone 5S에서 바이오인식을 위한 인증 시 센서에 의해 지문 데이터를 획득하는 과정에서 물리적인 센서 기만이 성공한 사례가 있다[7].

위와 같은 취약점들은 크게 다음과 같이 분류할 수 있다.

(a) 바이오인식 데이터 처리 문제(Fig. 1의 (3), (4), (7))
(b) 바이오인식 데이터 저장 문제(Fig. 1의 (2), (6))
(c) 바이오인식 데이터 암호화 문제(Fig. 1의 (4), (6), (7))
(d) 인증 루틴 우회 및 센서 기만 문제(Fig. 1의 (1), (2), (5), (7), (8))

4. 안전한 스마트폰 바이오인식 설계 요구사항

본 장에서는 3장에서 기술한 시스템 설계의 문제점을 해결하기 위한 안전한 스마트폰 바이오인식 시스템 설계 요구사항을 제시한다.

4.1 바이오인식 데이터 처리 방법 설계 요구사항

3장에서 기술한 바와 같이 바이오인식 데이터의 처리 시에 다양한 문제가 발생할 수 있다. 본 절에서는 바이오인식 데이터 처리 과정에서 발생할 수 있는 문제를 방지할 수 있는 설계 요구사항에 관해 기술한다.

1) 바이오인식 센서 및 데이터 처리 기능 제한

본 요구사항은 Fig. 1의 (3), (4), (7) 지점을 보호하는 방안이다. 먼저 바이오인식 센서 하드웨어에서 단순히 인식 기능 이외에 템플릿 데이터의 저장 기능까지 제공하는 경우에는 센서 외부로부터 내부의 데이터 유출을 방지하기 위해 단순히 등록이나 인증 결과만을 포함하는 한정적인 인터페이스를 제공해야 한다. 그러나 바이오인식 데이터를 인식하는 순간에 사용자 어플리케이션에서 원본 이미지 또는 추출된 템플릿 데이터를 디스플레이하는 등 결과 피드백의 제공이 필요한 경우가 있다. 이러한 피드백은 전혀 제공하지 않는 것이 바람직하지만 불가피한 경우에는 제공 정보를 최소화하고 원본 데이터 복구 공격을 막기 위해 피드백 정보의 해상도를 줄이는 등의 방법을 제공해야 한다. Jain 등은 지문 데이터가 인식되기 위한 최소 해상도가 250-300 PPI(Pixels Per Inch) 이상이 되어야 한다는 사실을 실험적으로 확인한 바 있다[8]. 따라서 지문 인식의 경우에는 피드백 이미지의 해상도가 250 PPI 미만이 되도록 설계함으로써, 사용자에게 필요한 최소한의 피드백 정보를 제공하되, 이 정보가 다른 시스템의 인식 정보로 사용되는 것은 막을 수 있다. 이와 마찬가지로 타 바이오인식 매체에 대해서도 사용자 피드백 이미지는 인증이 불가능한 수준으로 해상도를 줄여야 한다.

4.2 바이오인식 데이터 저장 방법 설계 요구사항

스마트폰에서 동작하는 바이오인식은 스마트폰에 저장된 바이오인식 데이터를 사용하는데, 바이오인식 데이터를 안전하게 저장하지 않을 경우 바이오인식 데이터 유출의 위험과 원본 데이터 해독의 위험이 있다. 본 절에서는 바이오인식 데이터 저장 과정에서 발생할 수 있는 문제를 방지할 수 있는 설계 요구사항에 관해 기술한다.

1) 저장된 바이오인식 데이터 관리

본 요구사항은 Fig. 1의 (2), (6) 지점을 보호하는 방안이다. 외부로부터 바이오인식 데이터 유출을 방지하기 위해 바이오인식 데이터를 저장하기 위한 안전한 저장소를 제공하도록 하여야 한다. 예를 들어 온라인 환경에서 바이오인식 기술을 활용한 인증방식을 다루는 국제 인증기술 표준인 FIDO (Fast Identity Online)는 클라이언트의 비밀키 및 바이오인식 데이터를 신뢰된 실행 환경(Trusted Execution Environment, TEE)에 저장할 것을 강력히 권장한다[9]. 또한, 저장된 바이오인식 데이터가 유출되었을 경우 유출된 데이터로부터 원본 데이터를 복구할 수 없도록 하여야 하고, 해당 데이터를 사용할 수 없도록 할 수 있어야 한다. 이에 따라 데이터베이스에 대한 적절한 암호화를 제공하거나, Ratha 등이 소개한 cancelable 바이오인식 방법을 사용하여 사용자 인증을 위한 바이오인식 데이터 등록 시마다 서로 다른 변형함수를 사용함으로써 유출된 바이오인식 데이터로부터 원본 데이터를 복구할 수 없도록 하여야 한다[10]. 마지막으로 안전하게 삭제되지 않은 바이오인식 데이터가 공격자에 의해 복구되는 경우 공격자가 해당 데이터를 사용하여 인증 과정을 진행할 수 있다는 위험이 있다. 이에 따라 바이오인식 데이터에 대한 삭제 시 추후 이를 복구할 수 없도록 바이오인식 데이터가 기록되어 있는 공간에 다양한 방식으로 재기록 작업을 수행하여 삭제하여야 한다[11].

4.3 바이오인식 데이터 암호화를 위한 설계 요구사항

스마트폰 환경의 특성상 디바이스를 탈취당할 경우 바이오인식 데이터가 Fig. 1의 (4), (6), (7) 지점에서 쉽게 유출될 수 있으므로 암호화는 필수적이다. 본 절에서는 바이오인식 데이터에 안전하게 암호화를 적용하기 위한 설계 요구사항을 기술한다.

1) 암호화키의 적절한 생성 및 관리

스마트폰에서 바이오인식 데이터가 유출되는 경우, 이를 다른 스마트폰에 주입하여 인증을 통과시키는 공격이 쉽게 이루어질 수 있다. 이러한 공격을 방지하기 위해서는 구현단계에서 암호화키의 하드 코딩은 지양해야 하며, 기기마다 다른 암호화키를 생성할 수 있도록 해야 한다. 이를 위해서는 프로세서 ID와 같은 하드웨어 기반의 고유정보나 PUF(Physical Unclonable Function)[12-13] 등을 이용하는 것이 바람직하다.

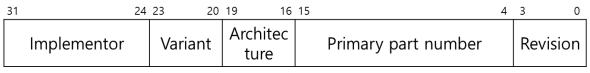

Fig. 2. Main ID Register Format of ARM Cortex-A Series

프로세서 ID를 구하기 위해서는 특정 레지스터 값을 가져오는 방법을 많이 사용하는데, Fig. 2는 ARM의 Cortex-A 시리즈 프로세서의 Main ID 레지스터 포맷이며 이 중 Primary part number 부분에서 부품 고유번호를 확인할 수 있다[14]. Apple의 A8, Qualcomm의 Snapdragon, Samsung의 Exynos 등의 프로세서들도 ARM 아키텍처를 기반으로 구현되었으므로 유사한 방식으로 프로세서 ID를 확인 가능하다[15-17]. 한편, PUF는 같은 설계도 및 공정을 이용하여 칩이 설계, 제조되었다 하더라도 공정상의 미세한 변화에 의해 칩마다 물리적 특성이 미세하게 다르다는 점을 이용하여 칩 고유 값을 쉽게 생성할 수 있는 방법이다. 공정에 따른 차이를 이용하는 PUF의 특성으로 인해 PUF 회로가 공개되어도 같은 출력의 회로 구성은 어려우므로, 위와 같이 생성된 고유 값은 해당 칩의 ID 또는 암호화키로 활용할 수 있다[12-13].

2) 암호화키의 안전한 적용

암호화키가 적절하게 생성된다 해도, 구현상에서의 암호화키 적용이 안전하지 못하다면 암호화의 안전성을 보장할 수 없다. 실제로, Jo 등은 [6]에서 역공학을 통해 지문인식 시스템이 포함된 상용 안드로이드 스마트폰의 지문 인식 어

플리케이션 내에 암호화키가 존재하는 것을 확인하고, 암호화된 지문 템플릿을 추출하여 이를 템플릿으로 복호화하는데 성공하였다. 이와 같은 문제를 해결하기 위해서는 역공학을 더욱 어렵게 만들어야 하는데, 해결법 중 하나로 난독화(Obfuscation) 기술이 있다. 난독화는 어플리케이션을 역컴파일(Decompile)하더라도 원본의 소스 코드와 제어 흐름을 알아보기 힘들게 만드는 것이 목적이다. 주로 소스 코드를 대상으로 하며, 대표적인 난독화 기법들로는 식별자 변환, API 은닉화, 제어 흐름을 복잡하게 만드는 기법 등이 사용된다. 대표적인 난독화 도구로는 ProGuard, DexGuard, Themida 등이 있다[18-21].

4.4 인증 루틴 우회 및 센서 기만 방지 설계 요구사항

본 절에서는 인증 루틴을 우회하거나 센서를 기만하는 공격을 방지하기 위한 바이오인식 시스템 설계 요구사항을 기술한다.

1) 바이오인식 템플릿 조작 및 매쳐 우회 방지

본 요구사항은 Fig. 1의 (5), (7), (8) 지점을 보호하는 방안이다. Fig. 1의 (5)와 (8)은 스마트폰 상의 바이오인식 시스템이 바이오인식 어플리케이션에 인증 요청 후 전달받은 결과를 통해 접근 허용 여부를 결정하는 점을 이용하여 바이오인식 어플리케이션 코드를 수정해 매치 결과를 조작하는 경우다. 이러한 공격으로 결과가 조작되는 것을 방지하기 위해서는 검증된 클라이언트 어플리케이션만이 설치와 실행이 가능한 환경을 제공해야 한다[22]. 삼성의 KNOX는 이러한 환경을 가진 하나의 사례로, KNOX 환경 하에는 KNOX Apps로부터 설치된 어플리케이션만 사용할 수 있으며[23], 루팅된 스마트폰의 진입 자체를 방지하도록 설계되었다[24]. 또한, 이러한 환경을 제공하는 경우 Fig. 1의 (3), (4), (7) 공격도 일정 부분 방지할 수 있다.

Fig. 1의 (7)은 저장된 템플릿이 매쳐로 전송되는 채널을 공격하여 데이터를 가로채 조작하는 경우로, 정상적인 매치 결과를 얻을 수 없게 만든다. 이는 저장된 템플릿을 전송받을 때 확인과정이 없으므로 발생한다. 바이오인식 사용자 등록 과정에서 바이오인식 템플릿에 MAC(Message Authentication Code)[25]을 추가하고 사용자 인증 시에는 MAC의 유효성을 확인하면 해당 기기에서 생성된 템플릿인지 여부를 확인할 수 있다. 단, MAC은 기기별로 다르게 생성하지 않으면 공격자의 기기에서 생성한 템플릿을 피해자 기기에서 구분할 수 없으므로, 앞서 언급한 하드웨어에 따른 특성을 보이는 프로세서 ID나 PUF 등을 이용하여 생성하는 것이 바람직하다. 이와 관련된 연구로, 워터마킹을 통해 바이오인식 데이터의 진위를 확인한 연구도 진행된 바 있다[26-27].

2) 바이오인식 센서 기만 방지

본 요구사항은 Fig. 1의 (1), (2) 지점을 보호하는 방안이다. 바이오인식 과정 수행 시 사용자는 센서를 통해 바이오인식 데이터를 입력한다. 이 경우에 iPhone 5S에 대한 공격 사례[7]와 같이 물리적으로 센서를 기만하는 공격을 방지하기 위해서는 liveness detection 기술을 적용하여 가짜 바이오인식 데이터와 실제 바이오인식 데이터를 구분하는 과정을 추가하여야 한다. 지문 인식의 liveness detection 방법에는 맥박 측정, 체온 측정, 피부 저항 탐지 등이 있으며 안면 인식의 liveness detection에는 눈의 깜빡임 감지, 안면 근육의 움직임 감지, 목소리 감지 등의 방법이 사용된다[28].

5. 안전한 스마트폰 바이오인식을 위한 프레임워크

본 장에서는 스마트폰을 이용한 바이오인식 시스템을 설계할 때 발생할 수 있는 잠재적인 취약점들을 방지하고, 본 논문에서 제시한 설계 요구사항을 만족시켜 안전한 스마트폰 바이오인식 시스템을 설계하기 위한 프레임워크를 제시한다. 다음은 스마트폰 바이오인식 시스템을 부주의하게 설계했을 때 생길 수 있는 잠재적인 취약점과 이를 해결하기 위해 프레임워크에서 고려해야 하는 모듈 또는 기능이다.

(a) 바이오인식 데이터의 안전한 처리를 위한 모듈
- 제한된 인터페이스만을 제공하는 센서
- 피드백 이미지의 해상도 축소 모듈

(b) 바이오인식 데이터의 안전한 저장을 위한 모듈
- 암호화를 위한 키 생성, 암호화/복호화 모듈
- 템플릿 저장을 위한 안전한 저장소
- Cancelable 바이오인식을 위한 템플릿 변형 모듈
- 안전한 바이오인식 데이터 삭제 인터페이스

(c) 바이오인식 데이터의 안전한 암호화를 위한 모듈
- PUF나 프로세서 ID 등의 하드웨어 기반 ID 생성 모듈
- 암호화 키 저장을 위한 안전한 저장소

(d) 인증 루틴 우회 및 센서 기만 문제 해결을 위한 모듈
- 템플릿 확인을 위한 MAC 모듈
- 센서 기만을 방지하기 위한 liveness detection 모듈

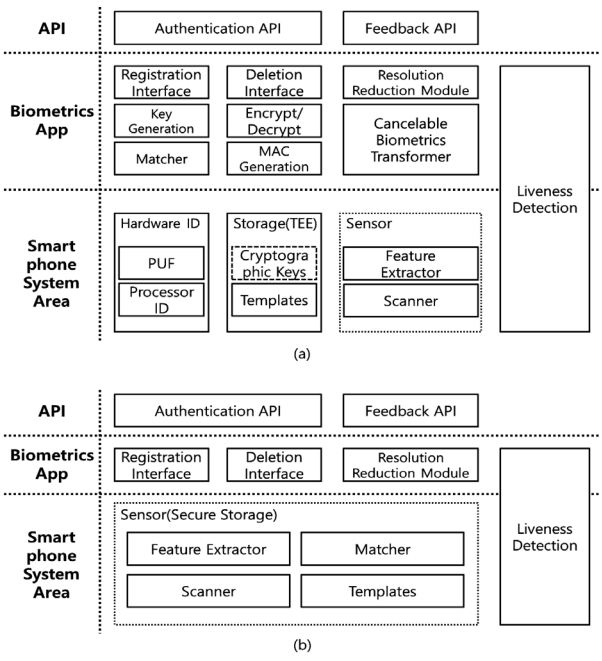

Fig. 3. Secure Biometric System Framework for Smartphone

Fig. 3은 이러한 프레임워크를 도식화한 그림으로 Fig. 3의 (a)는 바이오인식 센서 외부에 템플릿을 저장하고 매칭을 수행하도록 설계하기 위한 프레임워크이고, Fig. 3의 (b)는 바이오인식 센서 내부에 템플릿을 저장하고 매칭도 수행하여 바이오인식 어플리케이션에서는 인증 결과만을 받도록 설계하기 위한 프레임워크다.

Fig. 3의 (a), (b) 모두 바이오인식 이외의 다른 사용자 어플리케이션에 바이오인식 기반의 사용자 인증 기능을 제공하고 사용자에게 피드백을 제공하기 위한 API를 최상위층에 포함하며, 이 API 기능은 하위의 바이오인식 어플리케이션에서 제공한다. 바이오인식 어플리케이션은 바이오인식 센서를 포함하는 하위 시스템 영역의 기능을 활용하도록 구성되어 있다.

좀 더 구체적으로, 바이오인식 어플리케이션에서는 사용자 등록 및 삭제 인터페이스와 해상도 축소 모듈을 포함하고 있다. 또한, 바이오인식 원본 이미지 스캔 과정에서 바이오인식 어플리케이션과 시스템 영역을 아우르는 liveness detection 기법 적용으로 물리적 센서 기만행위를 방지한다.

Fig. 3의 (a)는 바이오인식 센서 외부에 템플릿을 저장하고 매칭을 수행하는 모델이므로, 바이오인식 어플리케이션에서 추가로 키 생성, 암호화/복호화, cancelable 바이오인식을 위한 템플릿 변형 모듈, 바이오인식 템플릿 매쳐 및 MAC 모듈 등 다양한 기능을 제공하여야 하며, 스마트폰 시스템 영역에서는 이에 대응하여 암호학적 키들과 템플릿 등을 안전하게 저장하기 위해 TEE 기능을 제공하는 안전한 저장소, 기기별로 다른 값을 갖는 PUF나 프로세서 ID 등의 하드웨어 ID 생성기, 바이오인식 데이터 스캔과 특징점 추출이 가능한 바이오인식 센서를 제공한다. 이 경우에 바이오인식을 위한 사용자 등록 과정은 센서의 스캐너로부터 입력받은 바이오인식 원본 데이터의 특징점을 추출하여 템플릿으로 만든 후 바이오인식 어플리케이션에 전송하여 cancelable 바이오인식을 위한 변형 모듈을 통해 cancelable 템플릿으로 변환한다. 변환된 템플릿에 기기 고윳값을 이용한 MAC을 붙이고 암호화한 후 TEE 환경을 제공하는 안전한 스토리지에 저장한다. 암호화 및 MAC 생성을 위해서는 바이오인식 어플리케이션 내의 키 생성 모듈이 시스템 영역의 하드웨어 ID 생성기로부터 받은 하드웨어 정보를 활용하여 키를 생성하고, 이를 이용하여 암호화/복호화 및 MAC 생성 모듈에 제공하게 된다. 키는 등록 및 인증 시 매번 동일한 방법으로 생성하면 저장이 불필요하나, 편의상 저장이 필요할 경우에는 안전한 저장소에 저장하여야 한다.

한편, 사용자 인증 요청을 받은 후에는 스캐너로부터 바이오인식 원본 이미지를 입력받아 특징점 추출로 템플릿을 구성한다. 사용자 등록과정과 마찬가지로 cancelable 템플릿으로 변환하고 매쳐로 보내져 안전한 저장소에서 복호화를 통해 가져온 템플릿 데이터와 비교하여 인증 결과를 반환한다. 이때, 저장된 템플릿에 대한 MAC을 확인하여 매쳐로 들어온 저장된 템플릿의 진위를 확인한다. 마지막으로 사용자 데이터 삭제 시에는, 안전한 저장소에 보관된 템플릿을 복구할 수 없도록 삭제한다.

Fig. 3의 (b)는 센서 내부에 템플릿을 저장하고 매칭도 수행하는 경우로 Fig. 3의 (a)에 비해 단순한 형태로 구성되어 있다. 센서는 외부로부터 완전히 차단된 안전한 환경이며, 자체적으로 바이오인식 원본 이미지 스캔, 특징점 추출, 템플릿 저장 및 매쳐의 기능을 수행한다. 그렇기 때문에 바이오인식 어플리케이션에서는 단순히 사용자 등록 요청, 사용자 인증 요청 및 사용자 삭제 요청만이 가능하며 모든 과정은 센서 내부에서 수행되어 바이오인식 어플리케이션은 단순히 등록, 인증, 삭제의 결과만을 반환받을 수 있다.

6. 결 론

본 논문에서는 일반적인 바이오인식 시스템의 잠재적 취약점 모델과 기존 스마트폰 바이오인식 시스템의 공격 사례 분석을 통해 스마트폰에서 바이오인식 시스템을 설계할 경우 발생할 수 있는 문제점을 기술하고 이를 분류하였다. 또한, 분류된 문제점을 바탕으로 안전한 스마트폰 바이오인식 시스템 설계를 위한 6가지 세부 설계 요구사항을 제시하고, 이를 종합하여 안전한 스마트폰 바이오인식을 위한 프레임워크를 제시하였다.

향후에는, 본 논문에서 제시한 설계 요구사항을 반영한 프레임워크를 구현하여 실제 스마트폰에 직접 적용함으로써 제안 프레임워크의 유효성이나 성능을 검증하는 동시에 안전성과 편의성 사이의 trade-off를 찾아볼 수 있다. 즉, UI/UX의 관점에서, 구현된 시스템의 보안 기능 제공 정도에 따른 사용자의 불편함 등을 분석하는 것도 가능하다. 이밖에도, 프레임워크를 스마트폰 전체의 보안에 적용하는 것과 특정 어플리케이션 진입 시의 보안에 적용하는 것에 대해서도 각각 위와 같은 UI/UX 관점의 분석을 수행하여 안전성을 조절하는 기준을 세울 수 있다. 마지막으로, 스마트폰 외에 바이오인식을 이용하는 다른 기기에서 발생할 수 있는 문제점을 분석하고 프레임워크를 제시하는 것도 가능하다.

References

[1] J.-H. Im and M.-K. Lee, “Requirement for Secure Biometric System Design on Smartphones,” *Proceedings of Korea Information Processing Society Fall Conference*, Vol.22, No.2, pp.870-871, 2015.

[2] Korea Internet & Security Agency (KISA), “Ten industrial issue in internet and information security 2015,” *INTERNT & SECURITY FOCUS*, pp.25-16, 2015.

[3] U. Uludag, S. Pankanti, S. Prabhakar and A. K. Jain, “Biometric Cryptosystems: Issues and Challenges,” in *Proc. IEEE*, Vol.92, pp.948-960, 2004.

[4] N. K. Ratha, J. H. Connell, and R. M. Bolle, “Enhancing security and privacy in biometrics-based authentication systems,” *IBM Systems Journal*, Vol.40, No.3, pp.614-634, 2001.

[5] S. Gibbs, HTC stored user fingerprints as image file in unencrypted folder [Internet], http://www.theguardian.com/technology/2015/aug/10/htc-fingerprints-world-readable-unencrypted-folder.

[6] Y.-H. Jo, S.-Y. Jeon, J.-H. Im, and M.-K. Lee, "Vulnerability Analysis on Smartphone Fingerprint Templates," *Futuretech 2015*, p.9, 2015.
[7] R. X. Cringely, Show of hands: Who hasn't hacked Apples's Touch ID? [Internet], http://www.infoworld.com/article/2612275/cringely/show-of-hands-who-hasn-t-hacked-apple-s-touch-id-.html.
[8] A. K. Jain, Y. Chen, and M. Demirkus, "Pores and Ridges: High-Resolution Fingerprint Matching Using Level 3 Features," *IEEE Tranactions on Pattern Analysis and Machine Intelligence*, Vol.21, No.1, pp.15-27, 2007.
[9] FIDO alliance, FIDO UAF Authenticator Commands v1.0 [Internet], https://fidoalliance.org/specs/fido-uaf-v1.0-ps-20141208/fido-uaf-authnr-cmds-v1.0-ps-20141208.html#bib-UAFProtocol.
[10] N. K. Ratha, S. Chikkerur, J. H. Connell, and R. M. Bolle, "Generating Cancelable Fingerprint Templates," *IEEE Tranactions on Pattern Analysis and Machine Intelligence*, Vol.29, No.4, pp.561-572, 2007.
[11] United States Department of Defense, DoD 5220.22-M, Operating Manual [Internet], https://www.fas.org/sgp/library/nispom/nispom2006.pdf.
[12] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, Vol.297, pp.2026-2030, 2002.
[13] G. E. Suh and S. Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation," *Design Automation Conference 2007*. 44th ACM/IEEE, pp. 9-14, 2007.
[14] ARM, ARM Cortex-A8 Technical Reference Manual [Internet], http://infocenter.arm.com/help/topic/com.arm.doc.ddi0344k/DDI0344K_cortex_a8_r3p2_trm.pdf.
[15] J. Ho, B Chester, C. Heinonen, and R. Smith, A8: Apple's First 20nm SoC [Internet], http://www.anandtech.com/show/8554/the-iphone-6-review/2.
[16] Qualcomm, Snapdragon 810 Processor Specification [Internet], https://www.qualcomm.com/products/snapdragon/processors/810.
[17] Samsung Exynos, Solution Overview [Internet], http://www.samsung.com/semiconductor/minisite/Exynos/w/solution.html#?v=overview.
[18] Y. Piao, J. Jung, and J. Yi, "Structural and functional analysis of ProGuard obfuscation tool," *The Journal of Korean Institute of Communications and Information Sciences*, Vol.38, No.08, pp.654-662, 2013.
[19] Guardsquare, ProGuard [Internet], http://proguard.sourceforge.net.
[20] Guardsquare, DexGuard [Internet], http://www.guardsquare.com/ dexguard.
[21] OREANS, Themida [Internet], http://www.oreans.com/.
[22] S.-Y. Jeon, J.-H. Im, Y.-H. Jo, and M.-K. Lee, "Potential Vulnerabilities and Solutions of Biometric Authentication on Smartphones," *The 25th Joint Conference on Communications and Information*, D1, 2015.
[23] Samsung, KNOX Apps [Internet], https://www.samsungknox.com/en/products/knoxworkspace/features/apps.
[24] P. Ning, "About rooting Samsung KNOX-enabled devices and the KNOX warranty void bit," Samsung KNOX, https://www.samsungknox.com/ko/blog/aboutrooting-samsung-knox-enabled-devices-and-knox-warranty-void-bit.
[25] ISO/IEC 9797-1 Std., "Information technology - Security techniques - Message Authentication Codes (MACs) - Part 1: Mechanisms using a block cipher," *ISO*, 2011.
[26] D. F. Smith, A. Wiliem and B. C. Lovell, "Face Recognition on Consumer Devices: Reflections on Replay Attacks," *IEEE Transaction on Information Forensics and Security*, Vol.10, No.4, pp.736-745, 2015.
[27] M. Vatsa, R. Singh, A. Noore, M. M. Houck, and K. Morris, "Robust biometric image watermarking for fingerprint and face template protection," *IEICE Electonic Express*, Vol.3, No.2, pp.23-28, 2006.
[28] M. Krieg and N. Rogmann, "Liveness Detection in Biometrics," *Biometrics Special Interest Group (BIOSIG), 2015 International Conference of the*, pp.1-14, 2015.

임 종 혁
e-mail : imjhyuk@gmail.com
2013년 인하대학교 컴퓨터정보공학부(학사)
2015년 인하대학교 컴퓨터정보공학과(석사)
2015년~현 재 인하대학교 컴퓨터정보공학과 박사과정
관심분야 : 정보보호, 암호학, 시스템 보안 등

권 희 용
e-mail : heeyong.kr@gmail.com
2015년 인하대학교 컴퓨터정보공학부(학사)
2015년~현 재 인하대학교 컴퓨터정보공학과 석사과정
관심분야 : 정보보호, 암호학, 네트워크 보안 등

이 문 규
e-mail : mklee@inha.ac.kr
1996년 서울대학교 컴퓨터공학과(학사)
1998년 서울대학교 컴퓨터공학과(석사)
2003년 서울대학교 전기컴퓨터공학부(박사)
2003년~2005년 한국전자통신연구원 선임연구원
2005년~현 재 인하대학교 컴퓨터정보공학과 부교수
관심분야 : 정보보호, 암호학, 컴퓨터이론