INAE Letters manuscript No.
(will be inserted by the editor)

Biometrics Data Security Techniques for Portable Mobile Devices: A Case Study

S. Roy · B. Matloob · A. Seetharam · A. Rameshbabu · W. C. O'Dell · W. I. Davis

Received: date / Accepted: date

Abstract In this work, a specific case-study of the development of a biometric authentication system for mobile clients (iPads) has been reported. The case-study addresses specific questions involving the technologies used, and provides a comparison of various products offered or developed by different vendors to implement such a system. We believe this work will help both researchers and developers who want to develop and experiment with similar biometric authentication systems - a form of multi-factor authentication gradually gaining usage.

Keywords Biometrics · multi-factor authentication · mobile clients

1 Introduction

Authentication in a cyber environment is the process of enabling a device to validate the identity of a specific user so as to allow or deny the user's request to access a system/network or a service. In this paper, we focus on biometrics (and specifically fingerprinting) as a means of authentication, particularly for mobile devices. We present a case study using the Fulcrum fingerprinting scanner and iPads, enlisting the steps needed to design a biometric authentication system for mobile devices. The goal is to provide the reader a recipe for setting up a biometric authentication system for mobile devices with limited effort.

A generic biometric authentication system is shown in Fig. 1. Setting up this authentication system requires a server and a centralized biometric

S. Roy, B. Matloob
University of North Florida

A. Seetharam
SUNY Binghamton

A. Rameshbabu, W. C. O'Dell, W. I. Davis
Johnson & Johnson Vision Care Inc.

database as shown in Fig. 1. In general there are two stages: enrollment and verification and/or identification. The enrollment phase creates an association between the user and the user's biometric characteristics. For ease of discussion, we assume that certain unique and identifying features extracted from a user's fingerprint are stored at the biometric database. For example, these fingerprint features could have been extracted and inserted into the database when the employee first joined the organization.

Verification involves validating if a claimed user is the actual user (e.g., authentication in an organization to gain access to a system). Identification is matching an unknown user from a list of potential candidates (e.g., authentication done during immigration in several countries). In this paper, we focus mainly on verification. Mobile clients (e.g. iPads, iPhones) establish a secure connection (e.g., by using SSL) to the server to authenticate themselves. The clients are also connected locally to a fingerprint scanner for obtaining the user's fingerprint. The image of the fingerprint is then transferred from the scanner to the mobile client, which then extracts features from the fingerprint. These features are sent to the server over the secure connection. These features are compared to the one available in the database to verify the authenticity of the user.

2 Authentication: An Overview

In this section, we provide an overview of how authentication is performed in a cyber environment. As mentioned before, authentication is the process of validating a user's identity. Authentication can either be one-way or mutual. In one-way authentication, only one of the parties involved in the communication authenticates itself to the other, while in mutual authentication both parties involved in the communication authenticate each other. The use of authentication builds a sense of trust for the end users, prevents access to a the network or service to intruders (i.e., external attack), and is necessary for holding a person accountable, in case the threats were a result of her actions (i.e., internal attack). Authentication along with the *CIA Triad* (confidentiality, integrity and availability) is one of the most important aspects of any cyber-physical system.

One of the simplest ways to implement user authentication is to use single-factor authentication. The widely adopted form of single-factor is to an username and password to log on to a website. Single-factor authentication provides limited protection against attackers; if the attacker obtains access to the user's username and password, then the user's account can be compromised. To overcome the drawbacks of single-factor authentication, multi-factor authentication has been proposed that involves identifying the user using more than one form of authentication. Multi-factor therefore takes a combination of several factors of authentication; three major factors include verification by:

1. Something a user knows (such as a password)
2. Something the user has (such as a smart card or a security token)

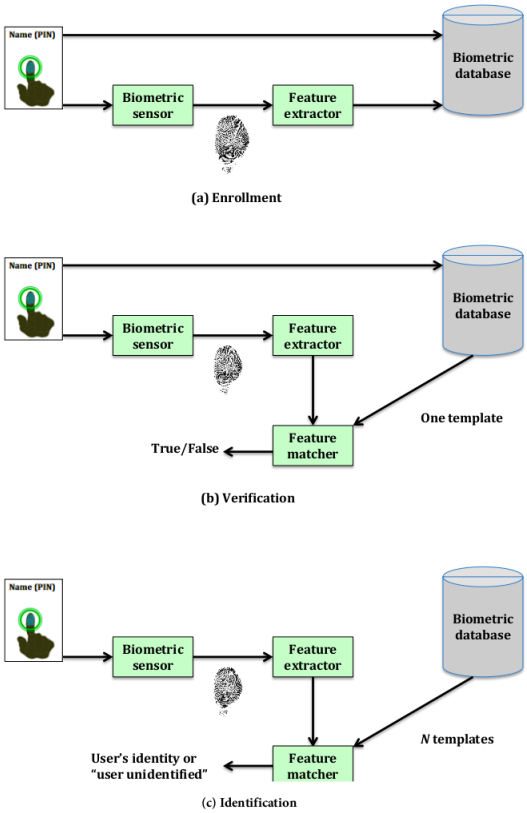

Fig. 1 A Generic Biometric System (based on [17])

3. Something the user is (such as the use of static biometrics)

A scenario where multi-factor authentication that has been widely adopted by taking the first two factors into account is in securing email services. Popular email services such as Gmail, Yahoo Mail and others have incorporated multi-factor authentication by asking users to (*i*) enter their password, followed by (*ii*) entering a code sent to their registered phone numbers via SMS. Multi-factor authentication using the first two factors definitely improves the quality of verification, but can still be vulnerable to attacks. For example, if a malicious user steals both the password and smart card of a legitimate user, the attacker can gain access to the user's account.

In scenarios where the level of security provided by the first two factors is insufficient, a third factor (e.g., biometrics) is also employed. Biometrics are essentially of two types - static and dynamic. Static biometrics are associated with an individual's physical characteristics. They include:

1. Facial characteristics

2. Fingerprints
3. Hand geometry
4. Retinal pattern
5. Iris pattern
6. Signature
7. Voice

Dynamic biometrics deal with something the user does or the user's behavioral characteristics (e.g., handwriting, voice pattern). The different forms of biometrics used for authentication vary in both cost and accuracy. Accuracy is measured by taking into account false positives (a legitimate person is denied access), and false negatives (an illegitimate person is provided access). Biometric features such as voice, facial characteristics, signature, and hand geometry are the least accurate followed by fingerprints and retinal patterns, and then iris patterns [17]. Authentication based on iris patterns is expensive and so most biometric systems today rely on retinal patterns and fingerprints.

As the chance of two users having the same biometric features is small, using biometrics improves authentication by making it extremely difficult for an attacker to compromise a system. Biometrics can definitely improve authentication, but they are expensive to implement and increase the complexity of the authentication systems[10, 9, 16, 8, 17].

2.1 Biometric Authentication in Mobile Devices

Use of biometrics as means of authentication is gradually becoming more frequent. This is triggered due to the increase in usage of mobile devices such as smartphones, laptops and tablets. The objective is to achieve the same level of security including error-rate and throughput through biometrics in these portable mobile devices when compared to fixed devices. Additionally, constraints such as storage of the biometric features, energy consumption of these devices are also important. For example, the feature (e.g. fingerprint) has to be stored locally in the device in some secured manner. Some services also offer to store these features in their own cloud depending on the application. These constraints necessitate a holistic modification of the biometric subsystem, including algorithmic optimization, consideration of new scenarios and use cases, and the adaptation of biometric sensors.

The primary challenge with incorporating biometrics in mobile devices is related to their vulnerability to sensitive data attacks, that accounts for a big challenge to biometric systems deployment. The use of biometrics in mobile devices in conjunction with other factors of authentication such as smart cards shows to be a viable solution. Therefore, the research for assuring security, reliability, and acceptable performance in this area is ongoing [3].

2.2 Market and Trends

The two industries that mainly use biometrics for security in mobile devices are finance and government. In the finance industry, the online banking systems play a major driver. The government applications consist of identifying people (often against their will) from nationalized fingerprint databases (e.g. during immigration). Biometric techniques have also been employed in voter registration and national identification projects. Many of those projects have been completed in developing economies, where biometrics can overcome low literacy rates and poor manual record keeping. As an example India's *Aadhar* is considered the world's largest national biometric identification system [5]. Developing economies are considered hottest market by several vendors.

Vendor	***Main Product(s)***	***Key Clients***
3M Cogent	Civil ID Systems, Readers and Scanners, Fingerprint Services	Various
Booz Allen Hamilton	Forensics	U.S. Army, Homeland Security
EyeVerify	Eye vein verification	RSA Security, Mountain America Credit Union
FIDO Alliance	Biometrics with Public Key Encryption	Alibaba Group, Bank of America, AMEX
Fujitsu	Fingerprint reader	Alliance with Microsoft, Oracle, etc.
Fulcrum Biometrics	Fingerprint reader, Mobile biometrics, Biometrics servers	Military, Financial, Nonprofit, etc.
Leidos	Mobile collection of fingerprints, iris, face	Military
Honeywell	Various	Various
RSA	Fingerprints with OTP	Various
Qualcomm	3D Fingerprints	DOCOMO

Table 1 Summary of the vendors providing biometric feature extraction

3 iPad Security

The problem addressed in this paper is to develop a biometric authentication system for iPad clients. This authentication system would enable users to have access to their respective accounts using iPads. Fingerprints were to be used as the authentication medium. Another requirement for this project was the fingerprints of the users should be stored in a centralized server database of the organization. The verification phase would happen in the server's end. In this section, we discuss a feature of an iPad security system called *Touch ID*, which is relevant to the research performed.

3.1 Touch ID

Touch ID is iPad's fingerprint sensing system that makes secure access to the device faster and easier. This technology reads fingerprint data from any angle and learns more about a user's fingerprint over time, with the sensor continuing to expand the fingerprint map as additional overlapping nodes are identified with each use [7]. This is a biometric fingerprint technology combined with some learning capabilities.

In this project we first explored if Touch ID could be used to capture the fingerprint for the biometric authentication (instead of an external fingerprint reader). We found third-party apps can use system-provided APIs to ask the user to authenticate using Touch ID or a passcode. The app is only notified as to whether the authentication was successful; it can't access Touch ID or the data associated with the enrolled fingerprint [7].

Again, one of the primary requirements of this project was the fingerprints would be securely stored in the organization's central server (and not locally on the iPad clients). Also the verification and authentication was to happen in the server (and again not on the iPad client). Since Touch ID did not serve both these requirements, the need to use an external fingerprint reader came up. We chose Fulcrum Biometrics as the external fingerprint reader for this project. The details are given in the subsequent sections.

4 Security of Biometric Sensors on Other Mobile Platforms

Majority of the mobile devices have been moving towards adopting fingerprint scanning as a means of authentication to replace passwords. While biometrics using fingerprint scanning has its advantages over passwords, the biggest disadvantage is the difficulty to replace the stored fingerprints if they are stolen or hacked. A password or even a smart card (other means of multi-factor authentication), can be easily replaced or reset if discovered to be stolen or hacked. But this is extremely difficult with biometrics, since the fingerprints represent a feature of a person, and cannot be changed. In this section, we report certain security breaches that have happened with different mobile devices that use fingerprints as a means of authentication.

One of the security preaches in mobile phones was reported here. Security contractors discovered preinstalled software in some Android phones that monitors where users go, whom they talk to and what they write in text messages [1]. Another incident reported here [18,15] talks about researchers at Michigan State University being able to bypass fingerprint authentications of Samsung Galaxy S6 and Huawei Honor 7, both of which run on Android. They took a photograph of a person's fingerprint and then used printers in their office to recreate the photograph in high-resolution on special paper. From there, to unlock the phone, it was as simple as placing the paper over the sensor. As a last report, a backdoor called *CoolReaper* that exposes users to potential malicious activity and appears to have been installed and maintained by Cool-

pad despite objections from customers [4]. This backdoor might be contained in millions of Android-based mobile devices sold by Coolpad, one of the worlds largest smartphone manufacturers based in China. To sum it up, issues with the security features of various mobile devices have been well-documented. However, the security of Apple products have been proved to be quite sturdy till date.

5 A Specific Case: Fulcrum Biometrics for iPads

In this section, our goal is to design and implement a biometric authentication system for iPad clients using the Fulcrum fingerprinting device. The iPads authenticate themselves to a centralized server that allows or denies access. The problem involved the following steps (illustrated in Fig. 2):

1. The fingerprint should be extracted at the iPad, and sent to the server (once for registration, and subsequently for verification).
2. The server then connects to its database to either register the fingerprint, or verify the fingerprint.
3. In case of a verification, the server allows/disallows the iPad client access to the resources.

The problem involving iPad is it has its own fingerprint scanning security system called *Touch ID security* [12,6]. But as with most of Apple's hardware and software, the fingerprint (image or vector) that Touch ID security extracts, cannot be obtained and used for any other purpose. We needed to find an external fingerprint reader for this purpose (as discussed in the previous section).

We chose the reader Fulcrum Biometrics [2] provides for this purpose. Fulcrum biometrics fingerprint reader comes with a Software Development Kit (SDK). The only drawback with this product is the incapability of extracting fingerprint information from the scanned fingerprint. It only provides a raw image of the fingerprint. That drawback added another step in the application development, which interfaces with the product.

5.1 Development Process

Since the Fulcrum Biometrics extracts and store the fingerprint in an image form, we had to find a way to extract the information (minutiae features) from the raw image. National Institute of Standards and Technology (NIST) has built and made available a project with many tools to extract minutiae features of a fingerprint. The project name is (NBIS) and it stands for NIST Biometric Image Software. However, the tools provided use C language and therefore it is easier to use these tools in the centralized servers rather than in the mobile devices themselves. Hence, we divided our research project into the following modules:

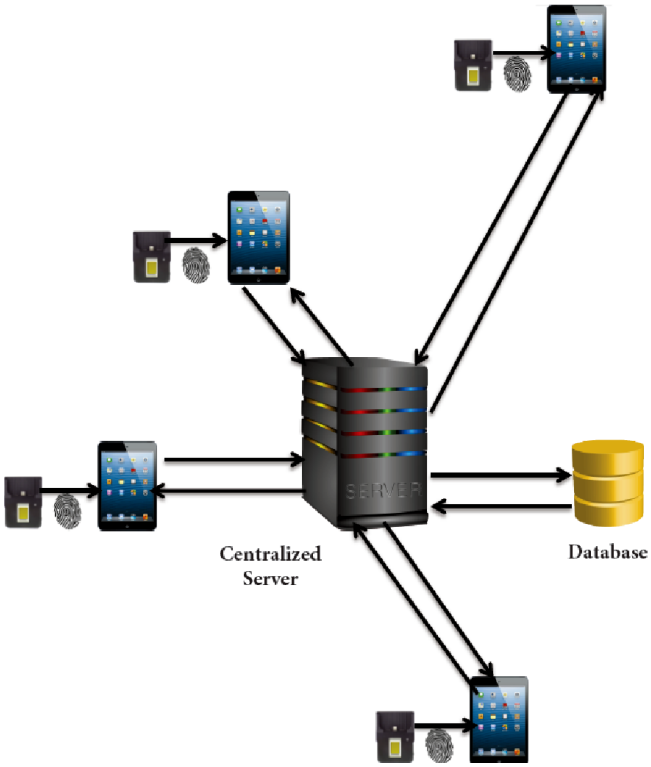

Fig. 2 The system for which solution was provided.

- Building the mobile application
- Building the Server

5.1.1 Building the mobile application module

There were two choices for programming language to implement the mobile application for the user interface on the iPads: (1) Objective C or (2) Swift language. However, since Fulcrums Biometrics SDKs are written in Objective C, it was logical also to use Objective C to implement the user interface on the iPads. As far as the mobile application development itself, we actually divided the mobile application development into a few stages.

1. The first stage is to implement the classes necessary to communicate with the centralized server through wireless network. These classes are responsible for opening the network socket, send and receive data, and close the socket when finished.
2. The second stage was to incorporate the Fulcrum Application Programming Interface (API) library provided in their SDK to the main program and establish the connection. Once the device receives data from the scan-

ner, it provides the data in a form of image and that allowed us to move to the next stage.
3. The third stage was to send the image obtained from the previous stage over to the server. This stage wasn't that long and it involved simply converting the image data into a string and sending it over to the server through the wireless network.

5.1.2 Building the Server

The centralized server that we used for our research is a Ubuntu 14.04 64-bit version server.

1. We developed a Java project that takes care of creating a network socket and wait and listen for any connection and receive data. The program basically receives the fingerprint image string from the mobile device through the wireless network and re-create the image file.
2. The next stage in the program development was to extract the information from the image received and match it up with either another image or information that are stored either locally or in the database to achieve identity verification. We used NIST's minutiae detector called, MINDTCT that are a part of their Biometric Image Software [13].

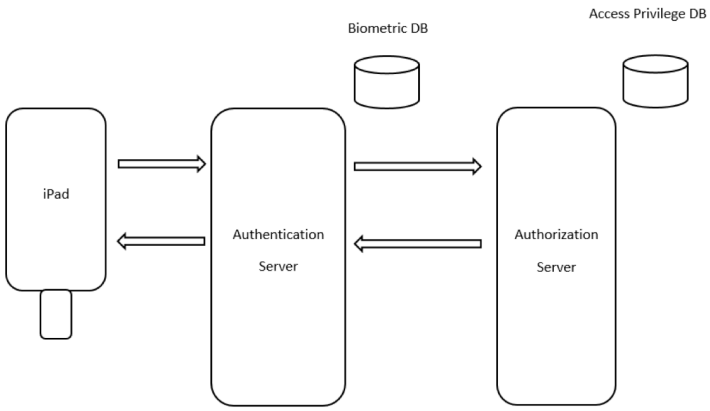

Fig. 3 The Servers and the Client

5.1.3 Integrating the Client(s) and Server

The integration of the client(s) was performed in two levels. The server we developed with the biometrics database is called the *Authentication Server* in Fig. 4. This Authentication Server interfaces between the Clients(s) (the

iPads), and another *Authorization Server*. The process of authentication and authorization proceeds as follows:

1. The Authorization Server is a preexisting server that contains the access privileges in its database of each employee of the organization.
2. The Authentication Server (developed by us) contain the biometric features in its (biometric) database of each employee, that is stored when an employee registers himself/herself using a client (iPad).
3. Both the servers also store the employee identity (employee-id), a unique identifier for any employee, which servers as the primary key (and foreign key) for the relationship of the schemas.
4. When an employee wants to login to access resources of the organization from an iPad, the employee uses the fingerprint reader (Fulcrum Biometrics) attached to the iPad to send his/her fingerprints along with his/her employee-id to the Authentication Server.
5. The Authentication Server then matches the received fingerprint with the one stored against the received employee-id (it rejects the request of the employee-id is not found in its database). The matching algorithm used here was NIST's fingerprint matching algorithm, BOZORTH3, which is a minutiae based fingerprint matching algorithm. It does both one-to-one and one-to-many matching operations. It accepts minutiae generated by the MINDTCT algorithm [13].
6. The employee gets authenticated by the Authentication Server depending on the outcome of the match.
7. If the employee is authentication, the Authentication Server sends a token that contains the timestamp and employee-id to the Authorization Server.
8. The Authorization Server then verifies the access privileges of the employee based on the received employee-id, and allows or disallows the employee's access request accordingly.

6 Evaluation of the System

The standard Detection Error Tradeoff (DET) curve, False Accept Rate (FAR), and False Reject Rate (FRR) have been used to measure the matching accuracy of the system [11]. FAR corresponds to impostor attempts that are falsely accepted, FRR corresponds to genuine attempts that are falsely rejected, while the DET curve plots FRR (Yaxis) vs. FAR (Xaxis). Since our tests were automated, we ignored the Failure to Acquire Rate (FTA) which is the proportion of the attempts for which the system fails to produce a sample of sufficient quality.

We obtained a dataset of 8-Bit Gray Scale Images of Fingerprint Image from NIST's repository of datasets for biometric system testing [14]. The dataset contains 2000 8-bit gray scale fingerprint image pairs. Each image is 512-by-512 pixels with 32 rows of white space at the bottom and classified using one of the five following classes: A=Arch, L=Left Loop, R=Right Loop, T=Tented

Arch, W=Whorl. The database is evenly distributed over each of the five classifications with 400 fingerprint pairs from each class.
Additionally the dataset has the following features:

- 2000 8-bit gray scale fingerprint image pairs including classifications.
- 400 fingerprint pairs from each of the five classifications - Arch, Left and Right Loops, Tented Arch, Whorl.
- each of the fingerprint pairs are two completely different rollings of the same fingerprint.
- 19.7 pixels per millimeter resolution.

To test our system:

1. We inserted one image from each pair into the server's database (a total of 2000 images inserted). Each image was chosen at random from its pair.
2. We then tested our system's FAR and FRR over the remaining 2000 (valid) images and another 2000 invalid images again obtained from NIST's dataset.
3. The FAR and FRR were calculated over four different values for *percentage matching* parameters. Each experiment was performed 100 times, and the average value was calculated over the 100 iterations.

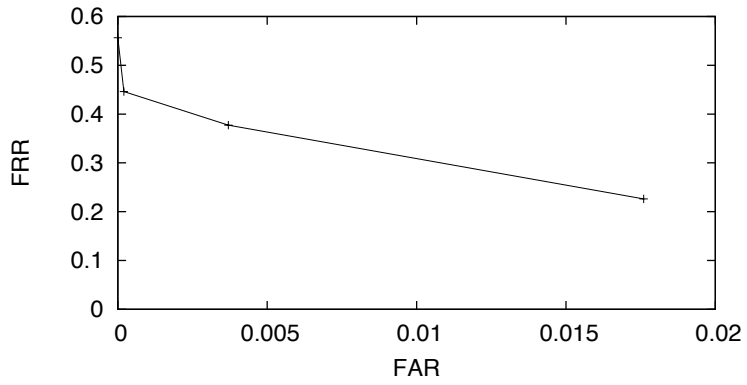

Fig. 4 DET curve for the experiment

The percentage matching parameter defines what percentage of match between two different images would be determined as an authentication pass. The matching performance of our system can be seen in Fig. 4 and Table 2. The percentage match parameter is shown alongside FAR and FRR to illustrate the tradeoff between security and matching. For more security (a higher value for percentage match), it reduces the FAR while increasing FRR. In other words, the chances of an impostor getting authenticated reduces, while the number of false positives, i.e. a legitimate person not getting authenticated increases by raising the value of percentage match.

Percentage Match	FRR	FAR
25%	0.2261	0.0176
35%	0.3773	0.0037
40%	0.4463	0.0002
50%	0.5564	0.0000

Table 2 FAR and FRR Performance Results

As shown in Fig. 4, with a percentage match setting of 50 and above, the false acceptance rate went down to zero. But that also generated a large number of false rejections (false positives). Therefore, we recommended a value of around 35 for the parameter.

7 Conclusion

In this paper, we provided an overview of authentication, with particular focus on authentication using biometrics in mobile devices. We then considered a particular case study of biometric authentication using iPads and the Fulcrum fingerprinting scanner and outlined the specifics steps needed to set up a biometric authentication system. The system was implemented over a secure internal network of the organization. Of further research would be the challenges faced when the same system is to be implemented over an insecure network. We did not use any indexing to improve the search in the database for a possible fingerprint during authentication. This is an area where the efficiency of the system can be further improved upon. The fingerprints were stored in the database in the bitstream form generated by MINDTCT. A further improvement of the security could be to use AES to store the bitstreams in an encrypted format. That would have the additional overhead of decryption during each authentication. Finally, the details of the implementation including our source code will be shared with any interested researcher upon request.

Acknowledgement

We thank the anonymous reviewers to help greatly improve the presentation and contents of this article.

References

1. Apuzzo, M., Schmidt, M.S.: Secret Back Door in Some U.S. Phones Sent Data to China, Analysts Say (2016 (accessed July 10, 2017)). https://www.nytimes.com/2016/11/16/us/politics/china-phones-software-security.html?rref=collection%2Fbyline%2Fmatt-apuzz
2. Biometrics, F.: Biometric software and security products— fulcrum biometrics (2011)
3. Blanco-Gonzalo, R., Sanchez-Reillo, R.: Biometrics on mobile devices. Encyclopedia of Biometrics pp. 282–289 (2015)

4. Burns, E.: CoolReaper backdoor uncovered in CoolPad Android devices (2014 (accessed July 10, 2017)). http://www.cbronline.com/news/security/coolreaper-backdoor-uncovered-in-coolpad-android-devices-4471988
5. Dass, R., et al.: Unique identity project in india: A divine dream or a miscalculated heroism? Indian Institute of Management (2011)
6. Hoog, A., Strzempka, K.: iPhone and iOS forensics: Investigation, analysis and mobile security for Apple iPhone, iPad and iOS devices. Elsevier (2011)
7. IOS: iOS Security (2017 (accessed July 10, 2017)). https://www.apple.com/business/docs/iOS_Security_Guide.pdf
8. Jain, A., Bolle, R., Pankanti, S.: Biometrics: personal identification in networked society, vol. 479. Springer Science & Business Media (2006)
9. Laï, L., Ho, S.W., Poor, H.V.: Privacy–security trade-offs in biometric security systemspart ii: Multiple use case. Information Forensics and Security, IEEE Transactions on **6**(1), 140–151 (2011)
10. Lai, L., Ho, S.W., Poor, H.V.: Privacy–security trade-offs in biometric security systemspart i: Single use case. Information Forensics and Security, IEEE Transactions on **6**(1), 122–139 (2011)
11. Mansfield, A.J., Wayman, J.L.: Best practices in testing and reporting performance of biometric devices (2002)
12. Morrissey, S., Campbell, T.: iOS forensic analysis for iPhone, iPad, and iPod touch, vol. 23. Springer (2010)
13. NIST: NIST Biometric Image Software (NBIS) (2010 (accessed February 3, 2017)). https://www.nist.gov/services-resources/software/nist-biometric-image-software-nbis
14. NIST: NIST Special Database 4 (2010 (accessed February 3, 2017)). https://www.nist.gov/srd/nist-special-database-4
15. Page, C.: Galaxy S6 and Honor 7 fingerprint sensors 'hacked' using an inkjet printer (2016 (accessed July 10, 2017)). https://www.theinquirer.net/inquirer/news/2450100/galaxy-s6-and-honor-7-fingerprint-sensors-hacked-using-an-inkjet-prin
16. Prabhakar, S., Pankanti, S., Jain, A.K.: Biometric recognition: Security and privacy concerns. IEEE Security & Privacy (2), 33–42 (2003)
17. Stallings, W., Brown, L.: Computer security. Principles and Practice (2008)
18. Szoldra, P.: It looks like even your fingerprint can be hacked (2016 (accessed July 10, 2017)). http://www.businessinsider.com/fingerprint-hacked-smartphones-2016-3