BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

DISSERTATION

BIOMETRIC SECURITY SYSTEMS
FINGERPRINT RECOGNITION TECHNOLOGY

2005 **ING. MARTIN DRAHANSKÝ**

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

DISSERTATION

submitted to the Faculty of Information Technology in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the field of study:

INFORMATION TECHNOLOGY

by

Ing. Martin Drahanský

BIOMETRIC SECURITY SYSTEMS
FINGERPRINT RECOGNITION TECHNOLOGY

Supervisor:	doc. Ing. František Zbořil, CSc.
Tutor:	Dr. rer. nat. Luděk Smolík
State doctoral exam on:	June 16, 2003
Submitted on:	March 30, 2005

This dissertation is available in the library of the Faculty of Information Technology of the Brno University of Technology, Czech Republic.

Affirmation

This dissertation is the result of my own work. The work has not been submitted either in whole or in part for a degree at any other university. Certain parts of the work have already been published in conference proceedings [23, 24, 25, 26, 29, 30, 33, 34], workshop proceedings [27, 31, 32], project reports [28, 3], as an article in the journal [35], and in the book [12].

Dedication

I dedicate this work to my parents.

Martin Drahanský

Abstract

This dissertation describes certain special art of biometric systems. General biometric systems are well known in public and systems based on the fingerprint recognition belong, without question, to the most familiar ones. Fingerprints have been used for identification and authentication for a long time because their uniqueness and reliability have been proven in everyday life. Nowadays, there are a great number of such biometric systems based on fingerprint recognition on the market. One group of them is used for forensic purposes (these are called dactyloscopic systems and are used in tasks of person identification). Another group of biometric systems represents the topic of interest of this dissertation – access or verification systems.

Both such systems and related basic biometric terms and processes are described in the first and second chapters.

If we try to combine a biometric (fingerprint) system with some cryptographic system, we are confronted with the question, if there is enough information entropy in the fingerprint. Some computations of the similarity among fingerprints have already been published but they have considered the matching of fingerprints. For cryptographic tasks, it is more important to exploit the information strength hidden in fingerprint papillary line structures. The answer to this question can be found in the third chapter.

Finally, if the information strength is adequate to the cryptographic requirements, we can design a system, which uses fingerprint technology as a biometric information input and offers biometric keys to the cryptographic subsystem. The detailed description of such Biometric Security System can be found in the fourth chapter, where all processes needed for the computations and processing are described. The Biometric Security System was implemented and appropriate modules were tested. The test analyses and reports are presented in the last, fifth, chapter.

Keywords: biometrics, cryptography, verification, authentication, identification, fingerprint, enrollment, matching, classification, orientation field, ridge count, biometric template, biometric key, certificate, entropy, error rate, PIN, password

Acknowledgements

First of all, I would like to thank my parents and friends for their tolerant attitude to my working effort during the elaboration of this dissertation, especially in the last months, when the intensity of work rapidly increased. I would also like to thank my parents for their incessant support during all the years of my studies.

From the scientific point of view, I would like to express my thanks to my supervisor doc. Ing. František Zbořil, CSc. for his support and help in my work, my tutor Dr. rer. nat. Luděk Smolík for his willingness to consult technical details of my work at any time and Prof. Dr. rer. nat. Christoph Ruland for his consultation to cryptographic problems. Further, I would also like to thank the team of the Fraunhofer Gesellschaft (IGD in Darmstadt, Germany), in particular Dr.-Ing. Christoph Busch and Dr.-Ing. Michael Arnold, for the possibility to work on very interesting projects focused on fingerprint recognition technologies, where I gained a great deal of knowledge and practical experiences. For the correction of my English text, I would like to thank Ing. Jiří Fojtek.

At last, I would like to thank the Brno University of Technology, Faculty of Information Technology for technical support, further the Grant Agency of the Czech Republic and the Foundation of the FRVŠ for financial support. The research was done under the support of the FRVŠ project No. FR0835/2003/G1, the GA GAČR project No. 102/01/1485 and the Research Intention No. CEZ:J22/98:262200012. Finally, I would like to express my gratitude to my colleagues of the University Siegen in Germany who helped me to create the database of fingerprints which was crucial for the completion of practical tests within this work.

Contents

List of Tables

List of Figures

Introductory Word

Fingers are not typical only for people, because e.g. the apes have fingers too. But one attribute is specific for human fingers – we have a unique art of skin corrugation which forms different patterns. Such patterns can be found not only on the fingers, but also on the whole palm and on the sole of our feet, including toes. This unique attribute cannot be found anywhere else on human body.

Such interesting finger patterns were discovered by human beings in very early epochs of their life on Earth. Some very old archaeological artefacts contain paintings of papillary lines and some prehistoric engravings represent fingers with their papillary lines. In those ancient times, these fingerprint interpretations were only amazing subjects and represented at most models for art artefacts.

The basics of modern and scientific method (called *dactyloscopy*) for fingerprint examination and comparison were established at first in the late 16th century. In 1864, the first scientific article about the study of papillary lines and skin structures [22] was published by *Nehemiah Grew*. Detailed investigations and research in this field followed. Later the classes of fingerprints were defined. The following three names are associated, in particular, with such classification: *Edward Henry*, *Francis Galton* and *J.E. Purkyne*. The present dactyloscopic systems use almost the same classification method (of course optimized for our actual conditions). The uniqueness of fingerprints has been accepted as an axiom and this axiom is still valid. The unique structures in fingerprints were used for other purposes too, not only for identification tasks as we know them today. The fingerprints have been used e.g. as the substitution of hand-written signature or as the trademark of a company (e.g. *Thomas Bewick* Factory). In the later years, this standalone scientific discipline was included into a wider discipline, the biometry. The biometry was (and still is) known as the science on comparative measurement of various human body parts, applicable for the purposes of identification of any particular person. Such measurements were done manually, results were stored in written form (on paper) and kept in a special filing cabinet.

Due to rapid progress in electronics and computer science, biometric data can be transformed into digital form and processed using computer technology so that an automatic comparison of such data can be made. This accelerated all the processes from the acquirement of biometric information to its evaluation. This progress was essential. Due to increase of Earth population, there was no other way which could be applied to practical processing of biometric data. At the moment, huge fingerprint databases exist, which contain millions of fingerprints.

With the invention of letters and writing, the demand on information hiding started to grow constantly. The first methods were simple (e.g. invisible ink), but better methods were developed with the improvement of theoretical knowledge, suitable not only for information hiding. These new methods included general algorithms for information encryption and decryption, using permutations and simple substitutions at the beginning, but later using keys as an important secret element to pro-

tect the information. The algorithm itself needs not to be kept secret – in fact actual algorithm structures are publicly known.

The scientists discovered that biometric features could be used not only for criminal investigation but also for other purposes. That was the beginning of the era of access systems. Such access systems can control the access to physical or logical objects. Users do not need to remember passwords or personal identification numbers; they could simply use their body attributes to get a granted access. With the increasing computational power and better theoretical basics, it has been possible to attack or deceive single cryptographic or biometric systems. This is therefore the reason for increasing use of the combination of both biometrics and cryptography.

The biometric data was (and often still is) only transformed to a digital form needed for the comparison with saved templates. Scientists wanted to transform biometric data in a convenient way, so that the information extracted each time from true biometric feature would be always the same. In general, the implementation of this idea seems to be quite difficult as the impression of our body parts vary a little bit every time. The variations of impressions can result from certain environmental influences or specific internal emotion of the person, or the system itself. Some new algorithms were developed and implemented what eliminates the influence of unwanted effects and transforms the input biometric impression of respective attribute to nearly identical representation. Such transformation could be used for the generation of biometric key and such biometric key then for cryptographic purposes (data encryption and decryption).

This thesis tries to describe new methods suitable for the Biometric Security Systems; such new methods are based on the known biometric systems but use cryptographic algorithms to enhance their functionality. The initial, rather theoretical part is followed by the description of biometric key generation; this all is related to the fingerprint technology.

In Brno, March 29, 2005 Ing. Martin Drahanský

1. Introduction

Authentication is a fundamental component of human interaction with computers. Traditional means of authentication, primarily passwords and personal identification numbers (PINs), have until recently dominated computing, and are likely to remain essential for the years to come. However, stronger authentication technologies, capable of providing higher degrees of certainty that a user really is who he or she claims to be, are becoming common. Biometrics is one of such strong authentication technologies.

Biometric technologies, as we know them today, have been made possible by explosive advances in computing power and have been made necessary by the near universal interconnection of computers around the world. The increased perception of data and information as near equivalents of money, in conjunction with the opportunities for access provided by the Internet, is a paradigm shift with significant repercussions on authentication. If data is money, then server-based or local hard drives are our new vaults, and information-rich companies will be held responsible for their security. Because of this, passwords and PINs are nearing the end of their life cycle for many applications. In the Fig. 1.1 you can see the increase of security and comfort for three different authentication methods [33].

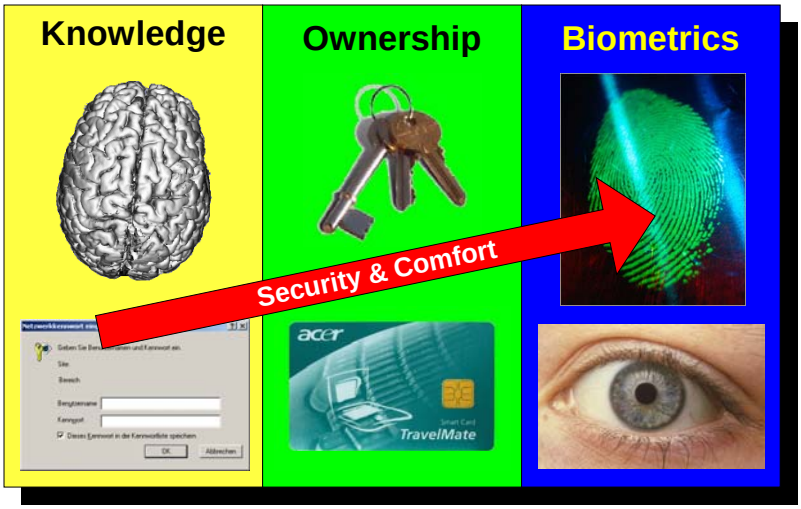


Fig. 1.1: Increase of security and comfort

Since early 1999, four factors (reduced cost, reduced size, increased accuracy, and increased ease of use) have combined to make biometrics an increasingly feasible solution for securing access to computers and networks. But biometrics is much more than a replacement for passwords or PINs. Millions of people around the world use biometric technology in applications as varied as time and attendance, voter registration, international travel, and benefit distribution. Depending on the application, biometrics can be used for security, for convenience, for fraud reduction, even as an empowering technology.

A number of biometric attributes are in use in various applications (Fig. 1.2 – the gray-grades correspond to the uniqueness of each biometric attribute; black = the most unique one, white = the least unique one [33]). Each biometric attribute has its strengths and weaknesses and the choice typically depends on the application [71, 33]. No single biometric attribute is expected to meet requirements of all applications effectively. The match between a biometric attribute and an application is determined depending on the characteristics of the application and the properties of the biometric attribute.

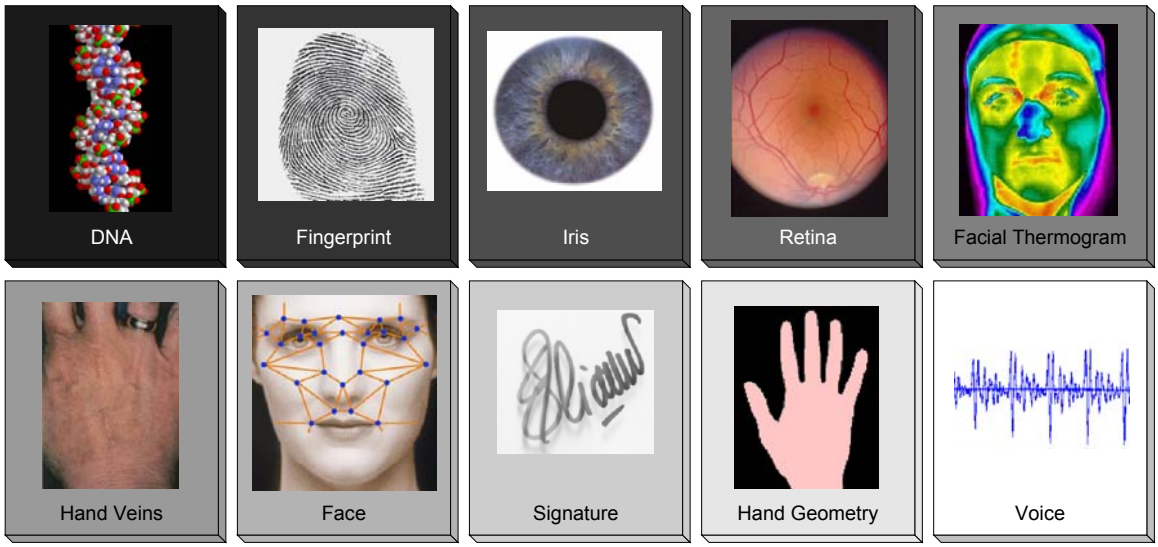


Fig. 1.2: Different biometric attributes (ordered in accord. with their uniqueness)

Biometric technology is used for dozens of types of applications, ranging from modest (providing time and attendance functionality for a small company) to expansive (ensuring the integrity of a 10 million-person voter registration database). Depending on the application, the benefits of using or deploying biometrics may include the increased security, increased convenience, reduced fraud, or delivery of enhanced services. In some applications, the biometric serves only as a deterrent; in others, it is central to system operation. Regardless of the rationale for deploying biometrics, there are two common elements:

1. The benefits of biometrics usage and deployment are derived from having a high degree of certainty regarding an individual's identity.
2. The benefits lead directly or indirectly to cost savings or to reduced risk of financial losses for an individual or institution.

1.1 Benefits of Biometrics

Three fundamental techniques are used in authentication mechanisms (Fig. 1.1):

- Something you know, which usually refers to passwords and PINs.
- Something you have, which usually refers to smart cards or tokens.
- Something you are, which refers to biometrics – the measurement of physical characteristics or personal traits.

The most frequently used authentication methods are passwords and PINs. They secure access to personal computers (PCs), networks, and applications; control entry to secure areas of a building; and authorize automatic teller machine (ATM) and debit transactions. Handheld tokens (such as cards and key fobs) have replaced passwords in some higher-security applications. Physical authentication devices, such as smart cards and password tokens, were developed in order to eliminate certain weaknesses associated with passwords. However, passwords, PINs, and tokens or cards have a number of problems that call into question their suitability for modern applications, particularly high-security applications such as access to online financial accounts or medical data. What benefits does the biometrics provide compared to these authentication methods?

a) *Increased Security*. Biometrics can provide a greater degree of security than traditional authentication methods, meaning that resources are accessible only to authorized users and are kept protected from unauthorized users. In theory, a password is memorized by a single person, it's hard to guess, it's never written down, and it's never shared. In practice, however, people constantly violate these expectations. Passwords and PINs are easily guessed or compromised; tokens can be stolen. Many users select obvious words or numbers for password or PIN authentication, so that an unauthorized user may be able to break into an account with little effort. In addition, many users write passwords in conspicuous places, especially as the number of passwords users must manage continually increases. "Good passwords", i.e. long passwords with numbers and symbols, are too difficult to remember for most users and are rarely enforced. In practice, password-based systems incorporate various cryptographic techniques to resist attacks, notably password hashing [98, 99].

   By contrast, biometric data cannot be guessed or stolen in the same fashion as a password or token. Although some biometric systems can be broken under certain conditions, today's biometric systems are highly unlikely to be fooled by a picture of a face, an impression of a fingerprint, or a recording of a voice. This assumes, of course, that the impostor has been able to gather these physiological characteristics – what is unlikely in most cases.

   In systems where the biometric authentication releases passwords (leveraging the existing username-password infrastructure), the user or administrator can create longer and more complex passwords than would be feasible without biometrics.

   Passwords, PINs, and tokens can also be shared, which increases the likelihood of malicious or unaccountable use. In many enterprises, a common password is shared among administrators to facilitate system administration. Unfortunately, because there is no certainty as to who is using a shared password or token – or whether the user is even authorized – security and accountability are greatly reduced. Being based on distinctive characteristics, biometric data cannot be shared in this fashion, although in some systems two users can choose to share a joint bank account by each enrolling a fingerprint.

   Although there are a number of security issues involved in biometric system usage that must be addressed through intelligent system design, the level of

security provided by most biometric systems far exceeds the security provided by passwords, PINs, and tokens.

b) *Increased Convenience*. One of the reasons passwords are kept simple (and are then subject to compromise) is that they are easily forgotten. As computer users are forced to manage more and more passwords, the likelihood of passwords being forgotten increases, unless users choose a universal password, reducing security further. Tokens and cards can be forgotten as well; though keeping them attached to keychains reduces this risk.

   Because biometrics are difficult if not impossible to forget, they can offer much greater convenience than systems based on remembering multiple passwords or on keeping possession of an authentication token. For PC applications in which a user must access multiple resources, biometrics can greatly simplify the authentication process – the biometrics replaces multiple passwords, in theory reducing the burden on both the user and the system administrator. Applications such as point-of-sale transactions have also begun to see the use of biometrics to authorize purchases from prefunded accounts, eliminating the need for cards.

   Biometric authentication also allows for association of higher levels of rights and privileges with a successful authentication. Highly sensitive information can more readily be made available on a biometrically protected network than on one protected by passwords. This can increase user and enterprise convenience, as users can access otherwise protected information without the need for human intervention.

c) *Increased Accountability*. Given the increased awareness of security issues in the enterprise and in customer-facing applications, the need for strong auditing and reporting capabilities has grown more pronounced. Using biometrics to secure computers and facilities eliminates phenomena such as buddy punching and provides a high degree of certainty as to what user accessed what computer at what time. Even if the auditing and reporting capabilities of a system are rarely used, the fact that they exist often serves as an effective deterrent.

The benefits of security, convenience, and accountability apply primarily to enterprises, corporations, and home users; in addition, they describe the rationale for biometric verification. The benefits of biometric identification, especially on a large scale, differ substantially.

The attacker usually has a grander goal in mind, such as the embezzlement of a certain amount of money or the capture of certain goods or services. But for the authentication system itself, the attacker's goal is usually limited to one of the three below described possibilities [98]:

- *Masquerade*. This is the classic risk to an authentication system. If attacker's goal is masquerade, he is simply trying to convince the system that he is in fact someone else. An attacker proceeds by trying to trick the system into accepting him as being the other person.
- *Multiple Identities*. Some systems, particularly those that dispense a government's social services program, are obligated to provide service to qualifying

individuals within their jurisdiction. These individuals generally show up in person and request services. For many reasons, however, some people have found it profitable to register two or more times for the same benefit.

- *Identity Theft*. This is the extreme case of authentication risks – when an attacker establishes new accounts that are attributed to a particular victim but authenticated by the attacker. In a simple masquerade, the attacker may assume the victim's identity temporarily in the context of system the victim already uses. In an identity theft, the attacker collects personal identification information for a victim (name, social security number, date of birth, mother's maiden name, etc.) and uses it to assume the victim's identity in a broad range of transactions.

In identification systems, biometrics can still be used for security, convenience, and accountability, especially when they are deployed to a modest number of users. However, identification systems are more often deployed in large-scale environments, anywhere from tens of thousands to tens of millions of users. In these applications, biometric identification is not replacing passwords or PINs – it is providing new types of fraud-reducing functionality.

a) *Fraud Detection*. Identification systems are deployed to determine whether a person's biometric information exists more than once in a database. By locating and identifying individuals who have already registered for a program or service, biometrics can reduce fraud. In a public benefits program, for example, a person may be able to register under multiple identities using fraudulent documentation. A person can also obtain fraudulent identification such as a driver's license. Without biometrics, there is no way to be certain that a person is not electronically registered under a different identity.

b) *Fraud Deterrence*. Perhaps even more than fraud detection, fraud deterrence is a primary benefit in large-scale identification systems. It can be difficult to return a highly certain match against millions of existing biometric records: In some cases, the error rates in large-scale identification systems can run into the single digits (much higher than would be acceptable in verification applications). However, the very presence of the biometrics provides a benefit, as it dissuades people who might otherwise be prone to attempt multiple registrations. If the presence of biometric identification technology can deter individuals from attempting to enroll multiple times in a public benefit or driver's license system, then the public agency has saved money and ensured the integrity of its records. In the absence of biometrics, there is no efficient way of identifying duplicate applicants or registrants, and it is therefore difficult to deter such applications.

Important is the *level of robustness*. It is the characterization of the strength of a security function, mechanism, service or solution, and the assurance (of confidence) that it is implemented and functioning correctly to support the level of concern assigned to a particular information system.

Notwithstanding the benefits of biometric technology, biometrics are not suitable for every application and user, and in some cases biometric authentication is simply the wrong solution. One of the major challenges facing the biometric industry is defining those environments in which biometrics provide the strongest benefit to

individuals and institutions, and then demonstrating that the benefits of deployment outweigh the risks and costs. Over time, the increased effectiveness and affordability of biometric technologies has continually broadened the range of applications in which biometrics operate effectively.

1.2 Key Biometric Terms

Because biometrics refers to such a broad range of technologies, systems, and applications, it is essential to discuss the terminology, classifications, and unique processes that define biometrics.

Biometrics is the automated use of physiological or behavioral characteristics to determine or verify identity.

Biometric System is essentially a pattern recognition system that recognizes a person by determining the authenticity of a specific physiological and/or behavioral characteristic possessed by the person. An important issue in designing a practical biometric system is to determine how an individual is recognized. Depending on the application context, a biometric system may be called either a verification or identification system. The scheme of the biometric system is shown in Fig. 1.3.

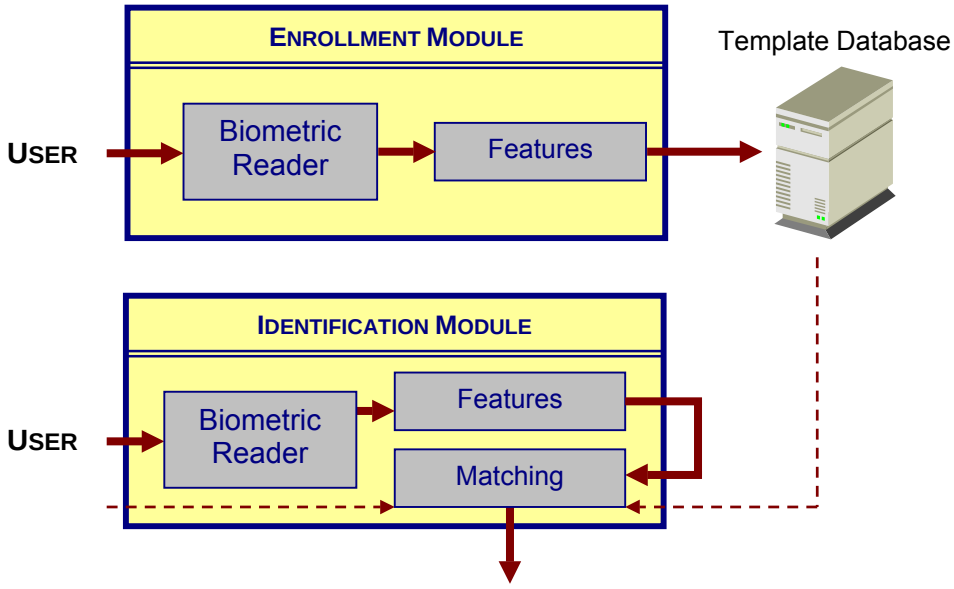


Fig. 1.3: General Biometric System

Automated use. Behavioral and physiological characteristics are regularly used to manually verify or determine identity – this is something that humans do every day when we greet friends or check an ID card. Biometric technologies, by contrast, are automated – computers or machines used to verify or determine identity through behavioral or physiological characteristics. Because the process is automated, biometric authentication generally requires only a few seconds, and biometric systems are able to compare thousands of records per second. A forensic investigator performing a visual match against an ink fingerprint is not performing biometric authentication. By contrast, a system wherein a user places his

or her finger on a reader and a match/non-match decision is rendered in real time, is performing biometric authentication.

Physiological or behavioral characteristics. Biometrics is based on the measurement of distinctive physiological and behavioral characteristics. Finger-scan, facial-scan, iris-scan, hand-scan, and retina-scan are considered as physiological biometrics, based on direct measurements of a part of the human body. Voice-scan and signature-scan are considered as behavioral biometrics; they are based on measurements and data derived from an action and therefore indirectly measure characteristics of the human body. The element of time is essential to behavioral biometrics – the characteristic being measured is tied to an action, such as a spoken or signed series of words, with a beginning and an end. The physiological / behavioral classification is a useful way to view the types of biometric technologies, because certain performance- and privacy-related factors often differ between the two types of biometrics. However, the behavioral / physiological distinction is slightly artificial. Behavioral biometrics is based in part on physiology, such as the shape of the vocal cords in voice-scan or the dexterity of hands and fingers in signature-scan. Physiological biometric technologies are similarly informed by user behavior, such as the manner in which a user presents a finger or looks into the camera.

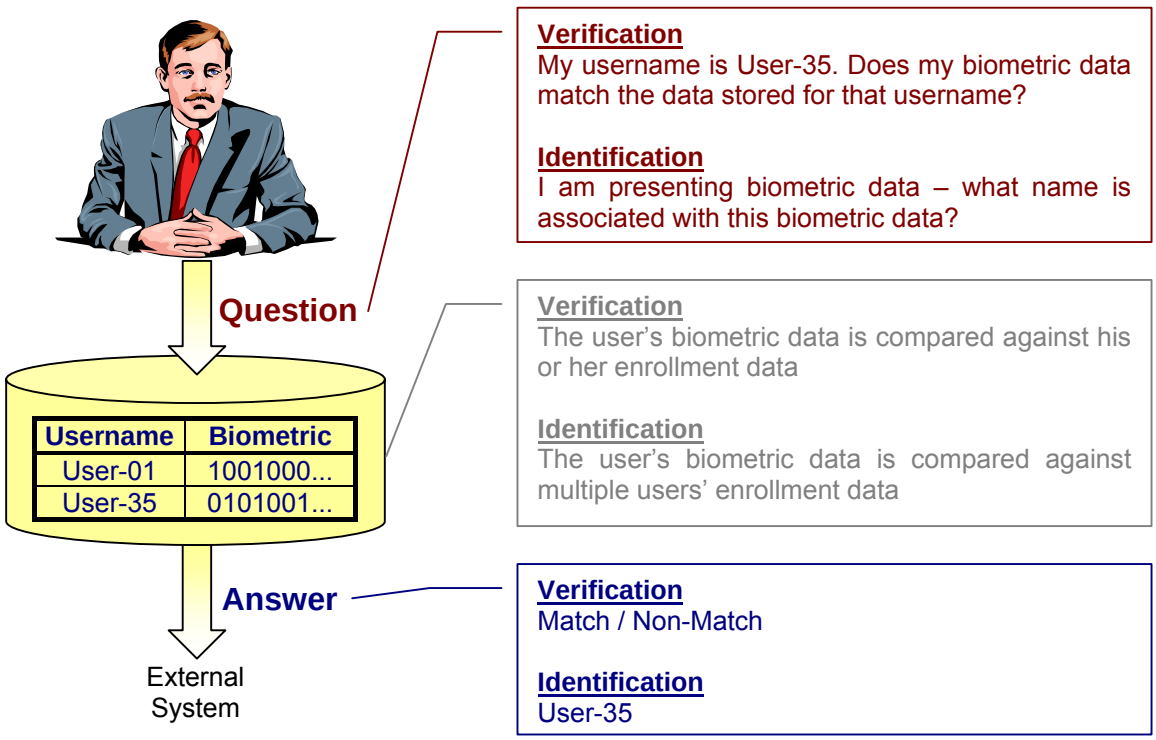


Fig. 1.4: Verification versus Identification

Determine versus verify. Determining versus verifying identity represents a fundamental distinction in biometric usage. Some biometric systems can determine the identity of a person from a biometric database without that person first claiming an identity. The traditional use of fingerprints in crime investigations – searching stored records of fingerprints in order to find a match – is an identification deployment. Identification systems stand in contrast to verification systems, in which a person claims a specific identity and the biometric system either confirms or denies that claim. Accessing a network is normally a verification event - the user enters an ID and verifies that he or she is the proper user of that ID by entering a password or biometric. Identification and verification systems differ substantially in terms of privacy, performance, and integration into existing systems (see also Figure 1.4).

Identity. Identity is often misunderstood in the context of biometrics, where a distinction must be drawn between an individual and an identity. An individual is a singular, unique entity – colloquially, a person – but an individual can have more than one identity. Identity distinction is important because it establishes limits on the type of certainty that a biometric system can provide. It can also have significant bearing on biometrics and privacy. Biometric identity verification and determination are only as strong as the initial association of a biometric with an individual. A user who enrolls in a biometric system under a false identity will continue to have this false identity verified with every successful biometric match.

Authentication is also frequently used in the biometric field, sometimes as a synonym for verification; actually, in the information technology language, authenticating a user means to let the system know the user's identity regardless of the mode (verification or identification).

Verification vs. Identification. Perhaps the most fundamental distinction in biometrics is between verification and identification (see Fig. 1.4). Nearly all aspects of biometrics – performance, benefits and risks of deployment, privacy impact, and cost – differ when moving between these two types of systems.

Verification systems answer the question, "*Am I who I claim to be?*" by requiring that a user claim an identity in order for a biometric comparison to be performed. After a user claims an identity, he or she provides biometric data which is then compared against his or her enrolled biometric data. Depending on the type of biometric system, the identity that a user claims might be a Windows username, a given name, or an ID number; the answer returned by the system is match or non-match. Verification systems can contain dozens, thousands, or millions of biometric records, but are always predicated on a user's biometric data being matched against only his or her own enrolled biometric data. Verification is often referred to as 1:1 (one-to-one). The process of providing a username and biometric data is referred to as *authentication*.

Identification systems answer the question, "*Who am I?*" and do not require that a user claim an identity before biometric comparisons take place. The user provides his or her biometric data, which is compared to data from a number of users in order to find a match. The answer returned by the system is an identity such as a name or ID number. Identification systems can contain dozens, thousands, or millions of biometric records. Identification is often referred to as 1:N (one-to-N or

one-to-many), because a person's biometric information is compared against multiple (N) records.

Within identification systems, there is a further distinction between positive and negative [71]. *Positive identification systems* are designed to find a match for a user's biometric information in a database of biometric information. A typical positive identification system would be a prison release program where individuals do not enter an ID number or use a card, but provide biometric data and are located within an inmate database. The anticipated result of a search in a positive identification system is a match. *Negative identification systems*, by contrast, are designed to ensure that a person's biometric information is not present in a database. This prevents people from enrolling twice in a system and is often used in large-scale public benefits programs in which users attempt to enroll multiple times to gain benefits under different names. Although the underlying biometric matching technology may be very similar to that of positive identification, the anticipated result of a search in a negative identification system is a non-match.

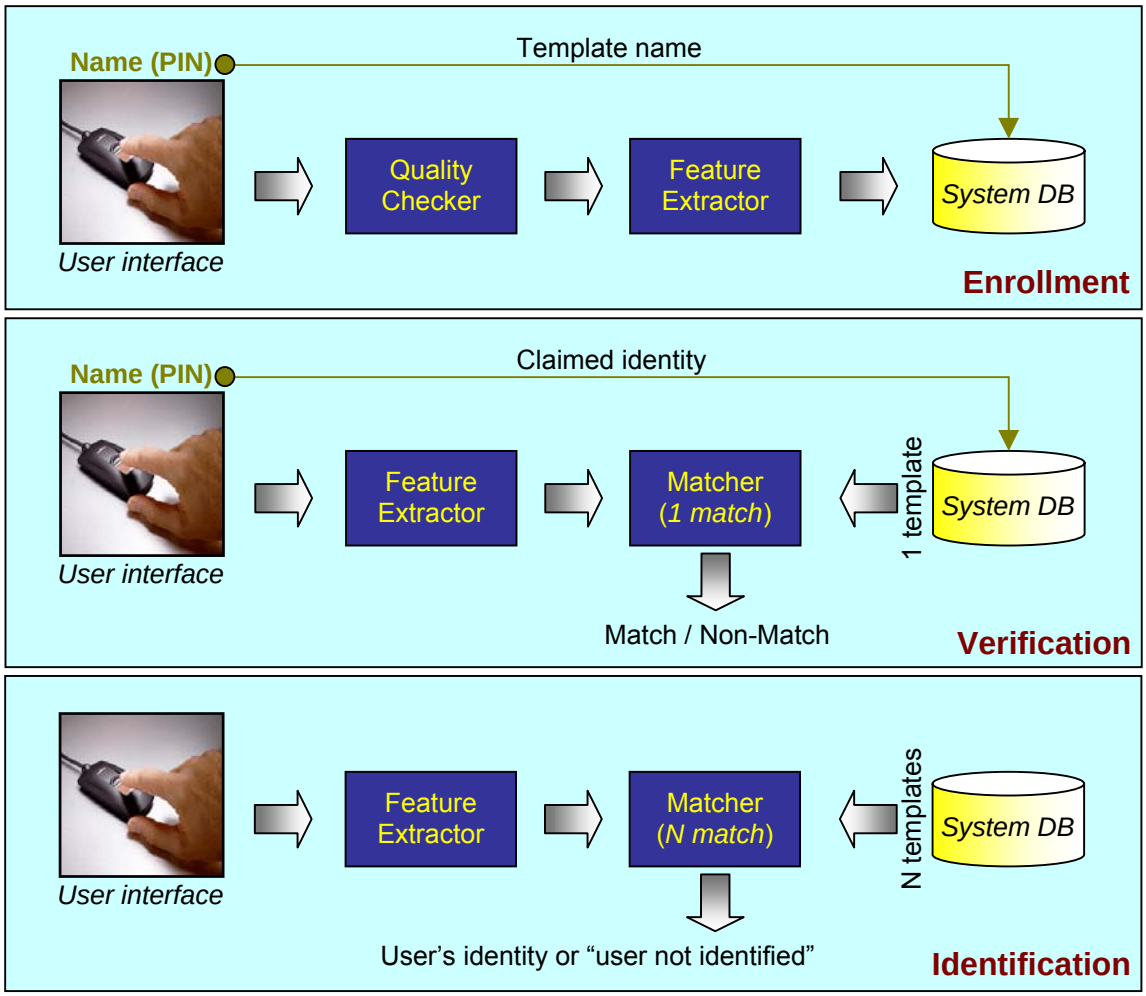


Fig. 1.5: Block diagrams of enrollment, verification and identification tasks

Identification systems with more than approximately 100 000 users are considered as a large-scale identification systems. *Large-scale identification systems* generally differ substantially from smaller-scale identification systems, especially in accuracy and response time, to the point where they effectively become a qualitatively different type of biometric technology.

Logical vs. Physical Access. Once a biometric system has determined or verified an identity, what happens? The answer depends on the purpose for which the system is deployed. Biometric systems, and in many ways the entire biometric industry, can be segmented according to the purposes for which verification and identification are being performed. The two primary uses for a biometric system are physical access and logical access.

Physical access systems monitor, restrict, or grant movement of a person or object into or out of a specific area. Most physical access implementations involve entry into a room or building: bank vaults, server rooms, control towers, or any location to which access is restricted. Time and attendance are a common physical access application, combining access to a location with an audit of when the authentication occurred. Physical access can also entail accessing equipment or materials, such as opening a safe or starting an automobile, although most of these applications are still speculative. When used in physical access systems, biometrics replace or complement keys, access cards, PIN codes, and security guards.

Logical access systems monitor, restrict, or grant access to data or information. Logging into a PC, accessing data stored on a network, accessing an account, or authenticating a transaction are examples of logical access. Biometrics replace or complement passwords, PINs, and tokens in logical access systems.

The core biometric functionality – acquiring and comparing biometric data – is often identical in physical and logical access systems. The same finger-scan algorithm and reader, for example, can be used for both desktop and doorway applications. What changes between the two is the external system into which the biometric functionality is integrated. In both physical and logical access systems the biometric functionality is integrated into a larger system (be it a door control system, for example, or an operating system). The biometric match is followed by an action such as at the opening of a door or access to an operating system.

Depending on the application domain, a biometric system could operate either as an *on-line* or an *off-line* system [73]. An on-line system requires the recognition to be performed quickly and an immediate response is imposed (e.g., a computer network log-on application). On the other hand, an off-line system usually does not require the recognition to be performed immediately and a relatively long response delay is allowed (e.g., an employee background check application).

How Biometric Matching works? Because most of us are accustomed to recognizing our friends and family through their faces and voices, as well as having to prove who we are with passwords and keys in our day-to-day lives, it is relatively easy to understand the concepts of biometric matching. However, the way in

which the biometric technology works – or the actual biometric matching functions – is more complex.

The following is the basic process flow of biometric verification and identification (see Figure 1.6):

- A user initially enrolls in a biometric system by providing biometric data, which is converted into a template (≈ digitized information).
- Templates are stored in the biometric system for the purpose of subsequent comparisons.
- In order to be verified or identified after enrollment, the user provides biometric data, which is then converted into a template.
- The verification template is compared with one or more enrollment templates.
- The result of a comparison among biometric templates is rendered as a score or confidence level, which is compared to a threshold used for a specific technology, system, user, or transaction.
- If the score exceeds the threshold, the comparison is a match, and that result is transmitted.
- If the score does not meet the threshold, the comparison is a non-match, and that result is not transmitted.

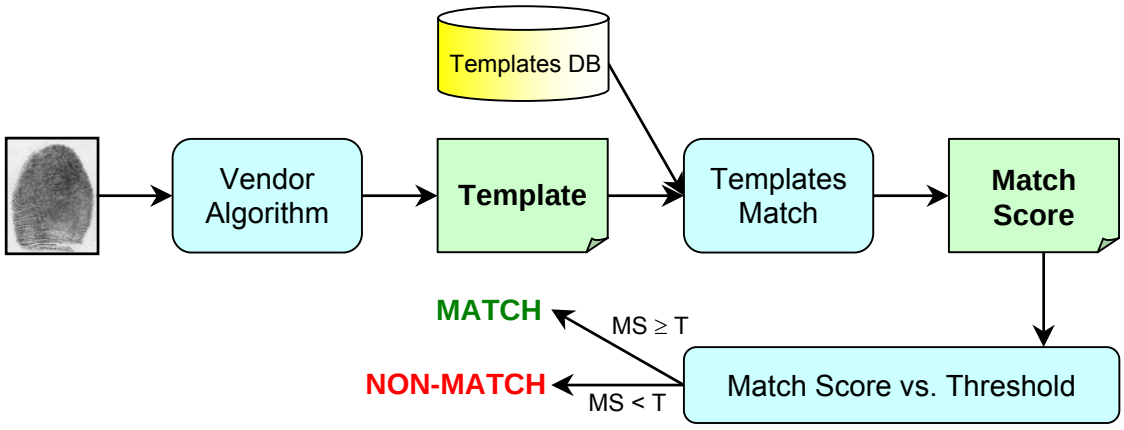


Fig. 1.6: Biometric Matching – Process Flow

Sensor. The sensor is an input device which transfers the real biometric information of the user into the electrical information and then into the digital information. The biometric system can further process the digital information. The sensor can be e.g. a fingerprint scanner, a microphone, a camera, etc. The technologies for fingerprint scanners are briefly described in the Chapter 2.1.1 and in [29]. Each sensor technology has some positive and negative features which influence the selection for practical use.

Enrollment and Template Creation. The following are key terms and processes involved in enrollment and template creation:

Enrollment. The enrollment process essentially introduces a person to the authentication system. The process by which a user's biometric data is initially acquired, assessed, processed and stored in the form of a template for ongoing use in a biometric system is called enrollment. Subsequent verification and identification attempts are conducted against the template(s) generated during enrollment. Enrollment takes place in both 1:1 and 1:N systems, although the way a user enrolls may vary substantially from system to system. Enrollment quality is a critical factor in the long-term accuracy of biometric systems. Low-quality enrollments may lead to high error rates.

Presentation. After a user provides required personal information to begin enrollment, such as name or user ID, he or she presents biometric data. Presentation is the process by which a user provides biometric data to an acquisition device – the hardware used to collect biometric data. Depending on the biometric system, presentation may require looking in the direction of a camera, placing a finger on a platen, or reciting a pass phrase. A user may also have to remove eyeglasses or remain still for a number of seconds in order to provide biometric data. Presentation of biometric data can take as little as one second or more than one minute. The manner in which a user presents biometric data to a system is also essential to long-term performance. Users must be cognizant of the manner in which they present biometric data in order to be verified or identified successfully.

Biometric data. The biometric data users provide, represents an unprocessed image or recording of a characteristic. This unprocessed data is also referred to as raw biometric data or a biometric sample. Raw biometric data cannot be used to perform biometric matches. Instead, biometric data provided by the user during enrollment and verification is used to generate biometric templates, and in almost every system is discarded thereafter. This needs to be repeated: Biometric systems do not store biometric user data for the template creation.

Depending on the biometric system, a user may need to present biometric data several times in order to enroll. For example, most finger-scan systems require the user to place each finger three to six times to gather sufficient data for template creation. The enrollment process may also gather data from more than one finger (or iris, or retina) to create multiple enrollment templates [34].

Enrollment requires the creation of an identifier such as a username or ID. This identifier is normally generated by the user or administrator during entry of personal data such as name and department. When the user returns to verify, he or she enters the identifier, then provides biometric data. Once biometric data has been acquired, biometric templates can be created by a process of feature extraction.

Feature extraction. The automated process of locating and encoding distinctive characteristics from biometric data in order to generate a template is called feature extraction. Feature extraction takes place during enrollment and verification – any time a template is created. The feature extraction process includes filtering and optimization of images and data in order to locate features accurately. For example, finger-scan technologies often thin the ridges present in a fingerprint image to

the width of a single pixel. Feature extraction processes of various vendors are generally patented and are always held secret. Since the quality of feature extraction directly affects a system's ability to generate templates, it is extremely important to the performance of a biometric system.

Templates. The template is a defined element of biometric technology and systems, and is critical to understanding how biometrics operate. A template is a small file derived from the distinctive features of a user's biometric data, used to perform biometric matches. Biometric systems store and compare biometric templates, not biometric data. Mathematical representation of templates is discussed in the Chapters 4.2 and 4.3.

Some important facts about biometric templates:

- Most templates occupy less than 1 kB; template sizes also differ from vendor to vendor. Such small file sizes allow for very rapid matching, allow biometrics to be stored on devices such as tokens and smart cards, and facilitate rapid transmission and encryption.
- Templates are proprietary to each vendor and each technology. There is nearly no commonly used biometric template format (see [104] – a standard for universal fingerprint template) – a template created in vendor A's system cannot be generally used through vendor B's technology [3]. This is beneficial from a privacy perspective, but the lack of interoperability has deterred some potential deployers who feared that they might be committed to a single technology.
- Biometric data such as fingerprints and facial images cannot be reconstructed from biometric templates. Templates are not merely compressions of biometric data, but extractions of distinctive features. These features alone are not sufficient to reconstruct the full biometric image or data.
- One of the most interesting facts about most biometric technologies is that unique templates are generated every time a user presents biometric data. Two immediately successive placements of a finger on a biometric device generate entirely different templates. These templates, when processed by a vendor's algorithm, are recognizable as being from the same person, but are not identical. In theory, a user could place the same finger on a biometric sensor for years and never generate identical templates. This is due to imperceptible changes in position, distance, pressure, and various other factors that affect biometric presentation.

Biometric Matching. The comparison of biometric templates to determine their degree of similarity or correlation is called matching. The process of matching biometric templates results in a match score, which, in most systems, is compared against a threshold. If the match score exceeds the threshold, the result is a match; if the match score falls below the threshold, the result is a non-match.

The matching process involves the comparison of a verification template, created when the user provides biometric data, with the enrollment template(s) stored in a biometric system. In verification systems, a verification template is matched against a user's enrollment template or templates (a user may have more than one

biometric template enrolled – for example, multiple fingerprints or iris patterns). In identification systems, the verification template can be matched against dozens, thousands, even millions of enrollment templates. The following are steps involved in matching:

- *Scoring*. Biometric match/non-match decisions are based on a *match score* – a number indicating the degree of similarity or correlation resulting from the comparison of enrollment and verification templates. Biometric systems utilize proprietary algorithms to process templates and generate match scores. There is no standard scale used for biometric scoring: Some biometric systems employ a scale of 1 to 100; others use a scale of -1 to 1. The scale is but unimportant; it depends on the used biometric system. Simplified can be said that the scale can be considered as some distance function (metric) to represent the results of the matching phase. These scores can be carried out to several decimal points and can be logarithmic or linear. Scoring systems vary not only from technology to technology, but from vendor to vendor.

  Scoring is a critical biometric concept and accounts for many of the strengths – and some of the weaknesses – of biometric systems. Traditional authentication methods such as passwords, PINs, keys, and tokens are binary, offering only a strict yes/no (or no, but try again) response. An attempt to verify via password will not succeed if it is close – it is either correct or incorrect. Biometric systems, by contrast, do not render absolute match/non-match decisions. Because different templates are generated each time a user interacts with a biometric system, there is no requirement of 100% correlation between enrollment and verification templates.
- *Threshold*. Once a score is generated, it is compared to the verification attempt's threshold. A threshold is a predefined number, generally chosen by a system administrator, which establishes the degree of correlation necessary for a comparison to be deemed a match. If the score resulting from the template comparison exceeds the threshold, the templates are a match (though the templates themselves are not identical). Thresholds can vary from user to user, from transaction to transaction and from verification attempt to verification attempt. Systems can be either highly secure or not secure at all, depending on their threshold settings. The flexibility offered by the combination of scoring and thresholds allows biometrics to be deployed in ways not possible with passwords, PINs, or tokens. For example, a system can be designed that employs a high security threshold for valuable transactions and a low security threshold for low-value transactions – the underlying comparison is transparent to the user.
- *Decision*. The result of comparison between the score and the threshold is a decision. The decisions of a biometric system can include match, non-match, and inconclusive, although varying degrees of strong matches and non-matches are possible. Depending on the type of biometric system deployed, a match might grant access to resources, a non-match might limit access to resources, while inconclusive may prompt the user to provide another sample. Therefore, for most technologies, there is simply no such thing as a 100% match. This is not to imply that the systems are not secure – biometric sys-

tems may be able to verify identity with error rates of less than 1 in 100 000 or 1 in 1 million. However, claims of 100% accuracy are misleading and do not reflect the basic operation of the technology.

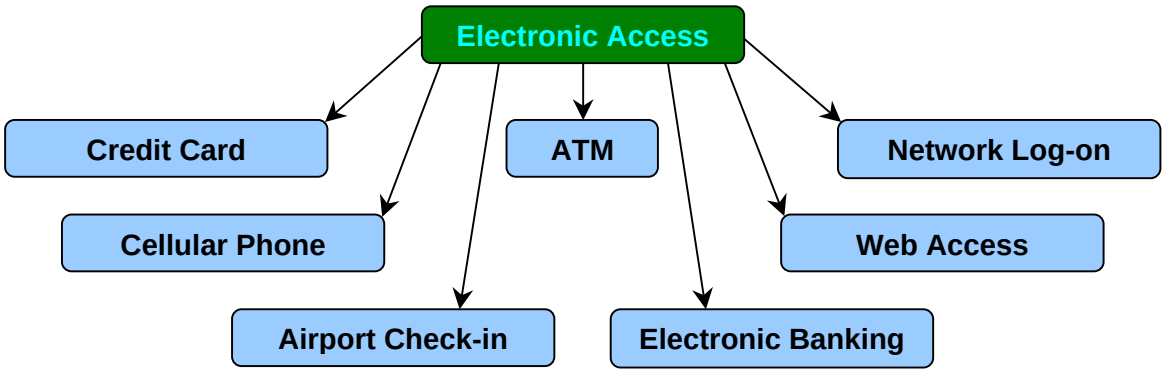


Fig. 1.7: Various electronic access applications in widespread use that require automatic recognition [71]

1.3 Error Classification and Performance Evaluation

To assess the performance of a biometric system, it can be analyzed in the framework of testing hypothesis [89]. Let the stored biometric sample or template be pattern $P' = S(B')$ and the acquired one be pattern $P = S(B)$. Then, in terms of testing hypothesis, we have null and alternative hypotheses:

$$H_0 : B = B', \quad \text{the claimed identity is correct} \tag{1.1}$$

$$H_1 : B \neq B', \quad \text{the claimed identity is not correct} \tag{1.2}$$

Certain measure of similarity $s = Sim(P, P')$ is often defined and H_0 is decided if $s \geq T_D$ and H_1 is decided if $s < T_D$, where T_D is a decision threshold. The measure of similarity s is also referred to as the *match score*. When $P = P'$, s is referred to as *a matching score* and B and B' are called a *matching pair*. When $P \neq P'$, s is referred to as a *non-matching score* and B and B' are called a *non-matching pair*.

With regard to the expressions (1.1) and (1.2), the decision H_0, when H_1 is true, gives *a false acceptance*; the decision H_1, when H_0 is true, results in a *false rejection*. The False Acceptance Rate (**FAR** – proportion of non-matching pairs resulting in False acceptance) and False Rejection Rate (**FRR** – proportion of matching pairs resulting in False rejection) together characterize the accuracy of a recognition system for a given decision threshold. Varying the threshold trades **FAR** off against **FRR**. In the Figure 1.8, **FAR** is the area under the H_1 density function to the right of the threshold and **FRR** is the area under the H_0 density function to the left of the threshold. In a more general framework, we can express the two errors as False Match Rate (**FMR**) and False Non-Match Rate (**FNMR**) [12, 3].

The Equal Error Rate (**EER**) corresponds to a point at some threshold (T_{EER}), where **FRR** = **FAR**, i.e. where the areas marked under the two curves (in Fig. 1.8) are equal.

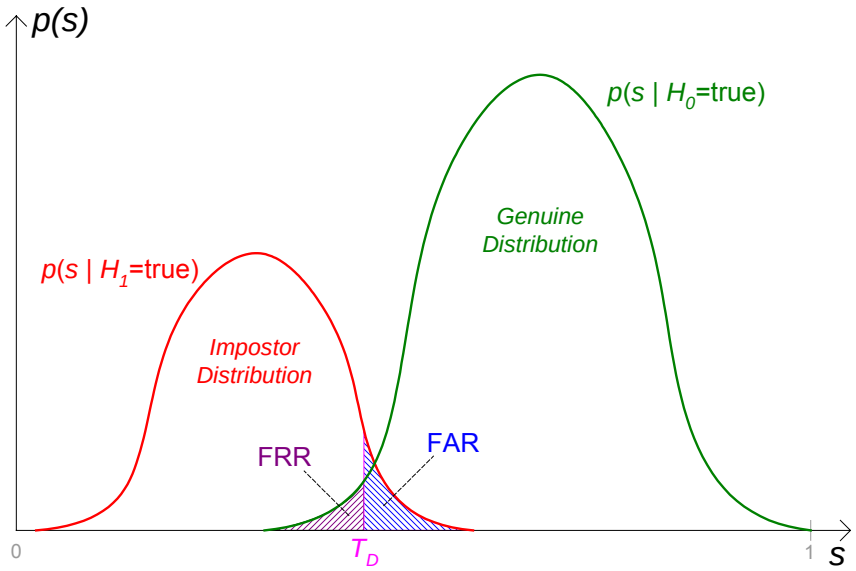


Fig. 1.8: Impostor and Genuine distributions

Rather than showing the error rates in terms of probability densities, as in Fig. 1.8, it is desirable to report system accuracy using a Receiver Operating Curve (**ROC**) [89, 12]. A **ROC** is a mapping $T_D \rightarrow$ (**FAR**, **FRR**) (Fig. 1.9):

$$ROC(T_D) = (FAR(T_D), FRR(T_D)) \tag{1.3}$$

Note that in a typical recognition system, all the information contained in the probability distribution functions is also contained in the **ROC**. The **ROC** can be directly constructed from the probability density functions as

$$FAR(T_D) = Prob(s \geq T_D \mid H_1 = true) = 1 - \int_0^{T_D} p(s \mid H_1 = true)ds \tag{1.4}$$

$$FRR(T_D) = Prob(s < T_D \mid H_0 = true) = \int_0^{T_D} p(s \mid H_0 = true)ds \tag{1.5}$$

If we let T_D go to zero, then **FAR** goes to one and **FRR** goes to zero; if we let T_D go to T_{max}, then **FAR** goes to zero and **FRR** goes to one.

The Failure to Acquire Rate (**FTA**) is defined as the expected proportion of transactions for which the system is unable to capture or locate an image or signal of sufficient quality. The **FTA** may depend on adjustable thresholds for image or signal quality [73, 3].

The Failure To Enroll Rate (**FTE**) is the expected proportion of the population for whom the system is unable to generate repeatable templates. This will include those unable to present the required biometric feature, those unable to produce an image of sufficient quality at enrollment, and those who cannot reliably match their template in attempts to confirm whether the enrollment is usable [73, 3].

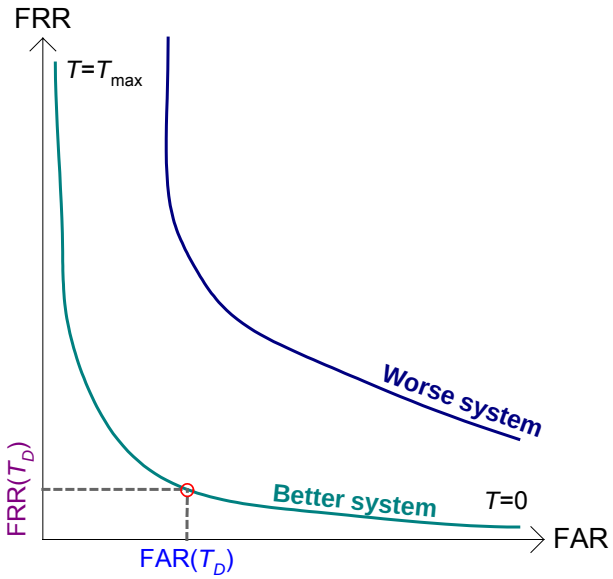


Fig. 1.9: Receiver Operating Curve (**ROC**)

The Failure to Match Rate (**FTM**) gives the percentage portion of the input biometric attributes, which cannot be compared with some saved template, or be processed. This failure rate expresses the meaning that the biometric system is unable to do, to some extent, any decision [12, 3].

1.4 Goals of this work

There are a number of biometric terms, concepts, and processes which potential deployers must fully understand in order to make reasonable decisions on biometrics. Familiarity with concepts such as biometric templates and biometric matching is essential to truly understanding privacy, security, and performance in biometric systems. The explanation of the combination of biometrics and cryptography will be outlined in this work.

There are three main goals which are new and newly published in this work.

The first goal is to estimate and compute the number of possibilities, which the fingerprint minutiae or details offer. This computation is important for an answer to the question, if there is enough information in the fingerprint to generate a key for the cryptographic purposes.

The second goal is to design a biometric security system, which supports the use of cryptography in conjunction with biometrics. This biometric security system should be open for other biometric technologies, such as voice, face or eye recognition, etc.

And the last goal of this work is the description of such fingerprint key generation. A vector should be generated from the fingerprint minutiae that could be considered as a key for symmetrical cryptography which would then protect appropriate confidential and/or secret data of the user.

2. Actual State

2.1 Problem Definition

In the context of fingerprint recognition, fingerprints or simply prints are generally used to refer to the impressions of human fingers. Operationally, fingerprint identification can be decomposed into the following three fundamental tasks [47]:

- Fingerprint acquisition,
- Fingerprint classification, and
- Fingerprint matching.

Fingerprints are acquired from fingertips or impressions of the ridges and furrows. Fingerprint classification assigns a fingerprint into a certain category according to its global ridge and furrow configuration. Fingerprint matching determines whether two fingerprints are from the same finger. Fingerprint recognition is one of the most reliable and valid personal recognition methods which has been in use for a long time.

2.1.1 Fingerprint Acquisition

Depending on whether the acquisition process is *on-line* or *off-line*, a fingerprint may be either (a) an *inked fingerprint* or (b) a *live-scan fingerprint. Inked fingerprint* is a term which is used to indicate that the fingerprint image is obtained from an impression of the finger on an intermediate media such as paper. Generally, inked fingerprint is obtained using the rolled method and called then *rolled inked fingerprint*. An example of a rolled inked fingerprint is shown in Figure 2.1 a) and b). Rolled fingerprints can be acquired by some special fingerprint scanners, which make fingerprint rolling possible.

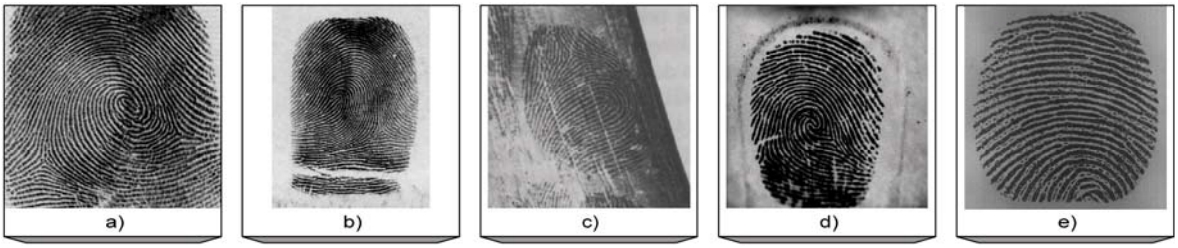


Fig. 2.1: Examples of fingerprints: a) and b) Inked fingerprints; c) Latent fingerprint; d) and e) Live-scan fingerprints

Rolled inked fingerprints impressed on paper can be electronically scanned into digital rolled fingerprints using optical scanners or video cameras. So far, rolled acquisition method remains the most popular acquisition technique in the area of criminal investigation. In fact, it has been essentially a standard technique for fingerprint acquisition for more than a hundred years [47]. Rolled inked fingerprints tend to have a larger area of valid ridges and furrows, but have large deformations due to the inherent nature of the rolled acquisition process. Direct feedback is not

available to the subject to control the acquisition process which, in turn, may result in difficulties in quality control. Acquisition of rolled fingerprints is cumbersome and slow. In the context of an automatic personal identification system, it is both infeasible and socially unacceptable to use the rolled inked fingerprint method to acquire fingerprints in the operational phase, although it may be feasible to use the rolled method in the enrollment phase.

In forensic techniques, a special kind of inked fingerprints, called *latent fingerprints*, is of great interest. Constant exudation of sweat pores on fingerprint ridges and intermittent contact of finger with other parts of human body and various objects result in a film of moisture and/or grease on the surface of fingers. When touching an object, the film of moisture and/or grease can be transferred to the surface of such object and leave an impression of ridges on it. This type of fingerprints is called a latent fingerprint. Actually, a major task in forensic fingerprint application is searching and reliably recording latent fingerprints which is beyond the scope of this thesis. A latent fingerprint is shown as an example in Figure 2.1 c).

The *live-scan fingerprint* is a collective term for a fingerprint image directly obtained from the finger without any intermediate step like getting an impression on paper. A number of scanning mechanisms can be used to scan ridges and furrows for finger impressions, including [22, 12] (a) optical frustrated total internal reflection (FTIR), (b) ultrasonic total internal reflection, (c) optical total internal reflection of edge-lit holograms, (d) thermal scanning of the temperature differential (across the ridges and valleys), (e) scanning of differential capacitance, and (f) non-contact 3D scanning. Optical scanning sensors use the light to acquire the fingerprint. Electrical field scanning sensors measure the local variation of the electrical field, which is caused by the emission of very small electrical signal on the relief of the finger surface. A capacitive scanning sensor forms some art of capacitor with the finger surface, whose capacity varies in relation to skin relief (ridges and furrows) A thermal sensor registers the thermal image of finger surface. The last type of sensor technology is based on ultrasonic scanning. These sensors measure the signals, which result from contact scattering of ultrasound on the scanned object, and compute the resulting image of the structure.

Sensors based on these physical processes can be used to acquire the impressions of human fingers, called *live-scan* fingerprints, in direct operation. These acquisition methods eliminate the process of intermediate digitization of inked impressions and make it possible to build on-line systems. Depending on the clarity of ridge structures of scanned fingers and acquisition conditions, acquired live-scan fingerprints vary in quality. However, since a direct feedback can be used on such type of devices, it is relatively easier to control the quality of acquired fingerprints. A live-scan fingerprint is usually obtained using the dab method, in which a finger is impressed on the acquisition surface of a device without rolling. A dab live-scan fingerprint only captures the ridges and furrows that are in contact with the acquisition surface. Therefore, it tends to have a smaller area of valid ridges and furrows and smaller deformations than a rolled fingerprint. The most popular technology to obtain a live-scan fingerprint image is based on optical frustrated total internal reflection (FTIR) concept [47]. When a finger is placed on one side of a glass plate (prism), ridges of the finger are in contact with the plate, whereas the furrows of the finger are not in contact with it. The rest of the imaging

system is an assembly consisting essentially of an LED light source and a CCD placed on the other side of the glass plate. The light source illuminates the glass at a certain angle and the camera can capture the light reflected from the glass. The light impinging on the plate at the glass surface touched by ridges is randomly scattered while the light impinging at the glass surface corresponding to furrows is totally reflected, what results in a fingerprint image on the imaging plane of the CCD. An example of live-scan fingerprint is shown in Figure 2.1 d) and e). Figure 2.2 shows some fingerprint scanners.

Fig. 2.2: Different fingerprint scanners [29]

Optical scanners are too large to be readily integrated in a number of applications such as laptop, cellular phone or notebook security devices. Recently, a number of different types of compact solid state fingerprint chips have become available. The quality of images acquired using these solid state chips is comparable to the quality of images acquired using optical scanners. These solid state chips can be manufactured with a very low cost if manufactured in a large quantity.

2.1.2 Fingerprint Classification

Global patterns of ridges and furrows in the central region of fingerprints form special configurations, which have a certain amount of intra-class variability. But these variations are sufficiently small, which makes a systematic classification of fingerprints possible. When considering the fingerprint classification, only a portion of a fingerprint, referred to as *the pattern area,* is of interest [47, 50]. The pattern area of a fingerprint consists of ridges encircled by *type lines* and is defined as the two innermost ridges that form a divergence tending to encircle or encompass the central portion of a fingerprint (Figure 2.3 shows an example of pattern area and type lines).

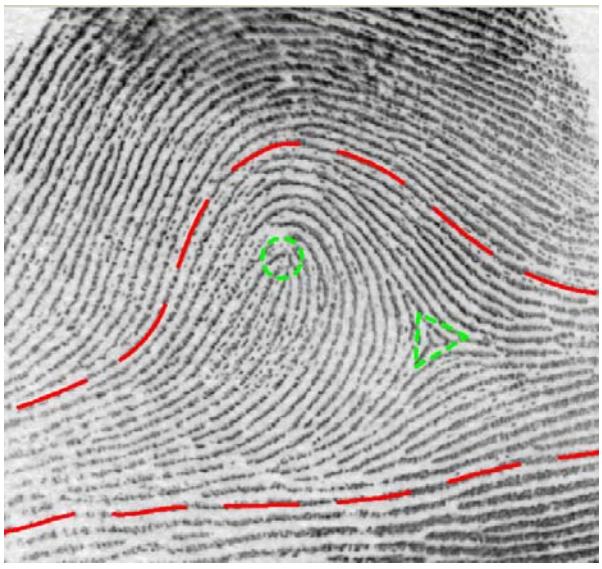


Fig. 2.3: Type lines (red), Core (green circle) and Delta (green triangle)

The pattern areas of loop or whorl types of fingerprints contain two types of singular points: (a) *delta* point and (b) *core* point. The *delta* point, sometimes called the outer terminus, is defined as the point of ridge at or in front of and nearest to the centre of the divergence of the type lines. It may be a ridge dot, a short ridge, the forking point of a bifurcated ridge, ridge ending, or the point on the ridge running in front of the divergence nearest to the centre between the innermost diverging ridges. The *core* point, sometimes called the inner terminus, is defined as the specific point located on or within the innermost sufficiently curved ridges. Due to large variations in the formations of curved ridges, the rules for selection of core points are very complicated. Another important notion in both fingerprint classification and fingerprint matching is *the ridge count*, which may be roughly defined as

the number of ridges that touch or cross an imaginary line drawn between the core and delta points (definition for dactyloscopic purposes). Due to the high complexity of ridge configurations, a precise definition of ridge count is difficult.

With the above definitions, fingerprint categories can be described as follows:

A *loop* is that type of fingerprint in which "one or more of the ridges enter on either side, recurve, touch or pass an imaginary line drawn from the delta point to the core point, and terminate or tend to terminate on or toward the same side from which such ridge or ridges entered" [22]. There are three essential ingredients for classifying a fingerprint into a *loop*: (a) at least one sufficiently recurved ridge, (b) a delta point, and (c) non-zero ridge count. *Loops* can be further divided into *lunar loop* and *radial loop* subcategories depending on the orientation tendency and fingers. About 50-60% of human fingerprints belong to this category [47].

A *whorl* is that type of fingerprint in which "at least two delta points are present with a recurve in front of each" [22]. This definition, though very general, captures the essence of the category. *Whorls* can be further divided into four subcategories: (a) *plain whorl*, (b) *central pocket loop*, (c) *double loop*, and (d) *accidental*. About 30% of human fingerprints belong to this category [47].

An *arch* is a special type of fingerprint configuration. Less than 15% of all fingerprints are *arches* [47]. *Arches* can be divided into two subcategories: (a) *plain arch* and (b) *tented arch*. A *plain arch* is that type of fingerprint in which ridges enter one side and flow out or tend to flow out the other side with a rise of wave in the centre. In a *tented arch*, most of the ridges enter one side and flow out or tend to flow out the other side with a rise of wave in the centre and the rest of the ridges form a definite angle, or up-wave.

Fingerprint classification still remains a very difficult problem for both human experts and automatic systems [47]. On the one hand, only a limited number of major fingerprint categories have been identified and the distribution of fingerprints into these categories is not uniform. On the other hand, as we mentioned above, there is a large variation in fingerprint configurations. The definition of each fingerprint category is both complex and vague. Figure 2.4 shows some examples of possible definable classes.

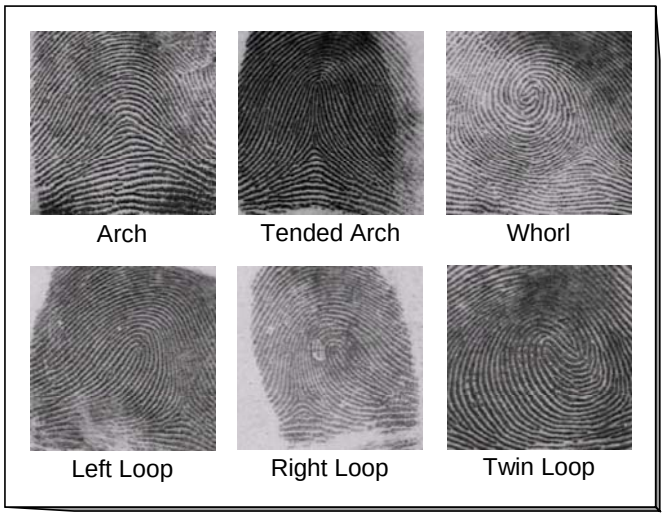


Fig. 2.4: Fingerprint classes [63]

2.1.3 Fingerprint Matching

Although the fingerprint category information and other global pattern configurations such as the number and positions of core and delta points, and the ridge count may indicate, to a certain extent, the individuality of fingerprints, the uniqueness of a fingerprint is exclusively determined by the local ridge characteristics and their relationships. Fingerprint matching depends on the comparison of local ridge characteristics and their relationships to determine the individuality of fingerprints. A total of 150 different local ridge characteristics, called *minutia* details, have been identified [47, 24, 58]. These local ridge characteristics are not evenly distributed. Most of them depend heavily on the impression conditions and quality of fingerprints and are rarely observed in fingerprints. The two most prominent ridge characteristics, called *minutiae*, are (a) *ridge ending* and (b) *ridge bifurcation*. A *ridge ending* is defined as the ridge point where a ridge ends abruptly. A *ridge bifurcation* is defined as the ridge point, where a ridge forks or diverges into branch ridges. Minutiae in fingerprints are generally stable and robust to the fingerprint impression conditions. Normally, they can be easily identified. Examples of minutiae are shown in Figure 2.5 [22]. For a given fingerprint, a minutia can be characterized by its *type*, its *x* and *y* coordinates, and its *direction* whose definition is shown in Figure 2.6.

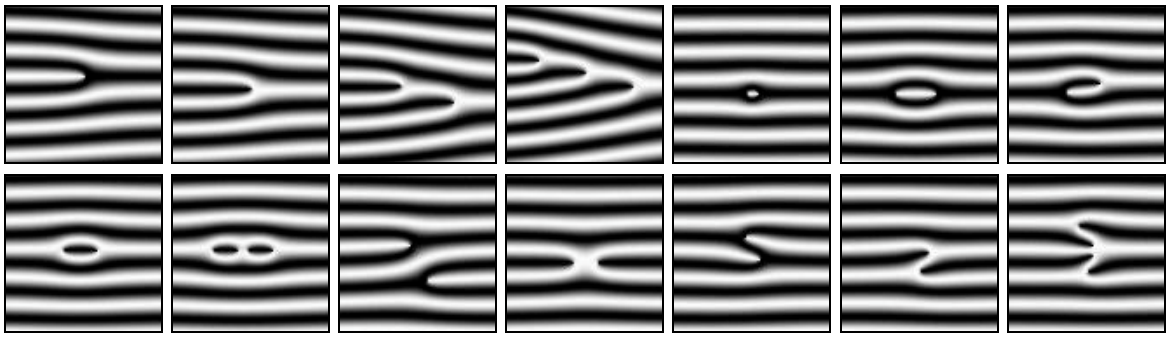


Fig. 2.5: Minutiae examples (Line Ending, Single Bifurcation, Double Bifurcation, Triple Bifurcation, Point, Interval, Hook; Single Whorl, Double Whorl, Through Line, Crossing, Side Contact, Single Bridge, Twin Bridge)

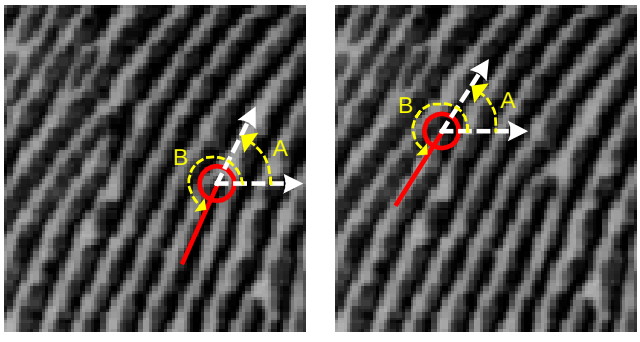


Fig. 2.6: Minutia orientation (A = Standard angle, B = FBI/AFIS angle) [42]

If two fingerprints belong to the same category and have a sufficient number of minutiae details that are identical, then it can be concluded confidently that they are from the same finger. Generally, in order to determine that two fingerprints are from the same finger (general description for dactyloscopic systems), four factors must be evaluated: (a) correspondence of general pattern configuration, which means that two fingerprints must be of the same pattern configuration, (b) qualitative correspondence which requires that the corresponding minutiae details must be identical, (c) quantitative factor which specifies that at least a certain number (a minimum of 12 according to the forensic guidelines in the United States and Germany) of corresponding minutiae details must be found, and (d) relationship of minutiae details which specifies that the corresponding minutiae details must be identically interrelated. In practice, many complex identification guidelines or schemes have been defined for fingerprint matching. These guidelines are carefully designed based on the knowledge of fingerprint experts. A detailed flow chart is available to guide fingerprint examiners in performing fingerprint matching.

Although various guidelines or schemes for fingerprint matching may differ in concept definition and decision making processes, the major steps in associated flow charts are essentially the same. Typically, a fingerprint matching process is represented by an iterative three-stage process. First of all, two fingerprints to be matched are compared to determine whether they are similar to each other in global pattern configuration. If the two fingerprints are totally different in terms of global pattern configuration, it is impossible that these two fingerprints are from the same finger. Next, a pattern comparison process is conducted in which several important feature points are first located in the fingerprints and then an approximate comparison of the fingerprints is performed. Finally, a matching process is conducted in which corresponding minutiae details are evaluated in the valid fingerprint pattern areas and a decision is made, based on the identified corresponding pairs and pattern configuration. Due to variations in fingerprint quality, impression deformation, fingerprint ridge configuration, and skin conditions, it is quite difficult to define certain steps in FP matching schemes clearly and precisely.

2.2 Fingerprint Recognition Algorithms

The whole process of fingerprint recognition could be divided into 5 main steps – see Figure 2.7 (results of processing):

1) *Acquirement of fingerprint*. The quality of acquired fingerprint is important for the fingerprint recognition. It is recommended to use a fingerprint sensor with a very good quality which could tolerate miscellaneous skin types, dryness or humidity of the finger grain. The basics were described in the Chapter 2.1.1.
2) *Fingerprint enhancement.* This step should enhance structures of papillary lines in damaged images. However, it is difficult to develop an algorithm which would be able to enhance all types of defects in fingerprint images. See description in the Chapter 2.2.1.
3) *Fingerprint classification.* It relates to the assignment of any fingerprint to the corresponding class [22, 47]. The classification is very demanding process, because in some cases it is difficult to say which class some fingerprint belongs to. The classification is based on the method of analysis of the Orienta-

tion Field. The basics were described in the Chapter 2.1.2. Further description can be found in the Chapter 2.2.2.

4) *Minutiae extraction.* In this step the structure of papillary lines is examined and the anomalies are detected and extracted as features (minutiae) – see the description in the Chapter 2.2.3. There are many minutiae points in the structure of papillary lines but only two of them are used for access systems, namely the ridge ending and ridge bifurcation. All arts of the minutiae points are used for the dactyloscopic system. At the moment, some other algorithms also exist which do not use minutiae points but some image parts as patterns.
5) *Fingerprints Matching.* The process of matching is based on the comparison of two fingerprints. The first fingerprint is the assumed original of the second one and is usually saved as a template (e.g. on the smart card). During the process, the saved minutiae points in the template and the extracted minutiae points from the newly acquired image are compared. The matching process actually corresponds with the application of pattern comparison algorithm on extracted parts of images.

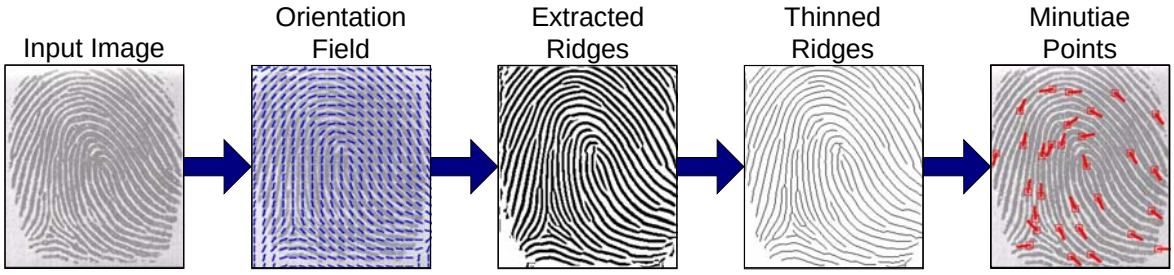


Fig. 2.7: Flowchart results of the minutiae extraction algorithm [93]

2.2.1 Fingerprint Enhancement

The performance of currently available minutiae extraction algorithms depends heavily on the quality of input images. In an ideal fingerprint image, ridges can be easily detected and minutiae can be precisely located from the thinned ridges. However, in practice, due to the factors of sensor environment and the state of human finger, a significant percentage of acquired fingerprint images (approximately 10%) is of poor quality [47]. The ridge structures in poor-quality fingerprint images are not always well-defined and hence they cannot be correctly detected. This leads to the following problems: (a) a significant number of unreliable minutiae patterns can be created, (b) a large percentage of genuine minutiae can be ignored, and (c) large errors in their localization (position and orientation) can be introduced. In order to ensure the robustness of performance of the minutiae extraction algorithm with respect to the quality of input fingerprint images, it is therefore necessary to use an image enhancement algorithm which can improve the clarity of the ridge structures of input fingerprint images.

Ideally, the ridge structures in a fingerprint image are well-defined. Each ridge is separated by two parallel narrow furrows, each furrow is separated by two parallel narrow ridges; and minutiae are anomalies in the ridges, i.e., ridge endings and

ridge bifurcations. When a fingerprint image is disrupted, such well-defined ridge structures are no longer visible. However, despite the existence of certain disruption, a fingerprint expert is often able to identify the minutiae correctly by using various visual clues such as local ridge orientation, ridge continuity, and ridge tendency. It is possible to develop an enhancement algorithm that can exploit these visual clues to improve the clarity of ridge structure in fingerprint images which in turn will improve the performance of the minutiae extraction algorithm.

2.2.2 Fingerprint Classification

There are two classification systems. The first one is the *Galton Classification System*, which includes only three fingerprint classes [65]. The second one is the *Henry Classification System*, which includes five fingerprint classes and is used by the most fingerprint recognition systems [65].

The mostly used classification method is the *exclusive classification*, which assigns a fingerprint to certain predefined classes according to their macro-features. The second classification method is the *continuous classification*, which characterizes each fingerprint with a numerical vector [70]. The similarity degree is determined with respect to a predefined set of class prototypes.

The FBI adopted the Henry Classification System as follows (see Chapter 2.1.2 and Figure 2.4):

- *Arch* and *Tended Arch*,
- *Right Loop* and *Left Loop***,**
- *Whorl*.

The fingerprints have been traditionally classified into classes based on information in the global patterns of ridges. Figure 2.8 shows the functionality of fingerprint classification scheme [58, 59], an abbreviated explanation can be found on the next page. The “re-compute” option involves starting the classification algorithm with a different preprocessing (e.g. smoothing) of the image.

A fingerprint classification system should not be sensitive to rotation, translation, and elastic distortion of the frictional skin. In addition, often a significant part of the finger may not be imaged (e.g. dabs frequently miss delta points) and the classification methods requiring information from the entire fingerprint may be too restrictive for many applications.

A number of approaches to fingerprint classification have been developed. Some of the earliest approaches have not exploited rich information in the ridge structures and exclusively depended on the Orientation Field information (see the Chapter 4.2.1, Method based on the Orientation Field). Although fingerprint landmarks provide very effective fingerprint class clues, methods relying on the fingerprint landmarks alone may not be very successful due to lack of availability of such information in many fingerprint images and due to difficulties in extracting the landmark information from disrupted fingerprint images.

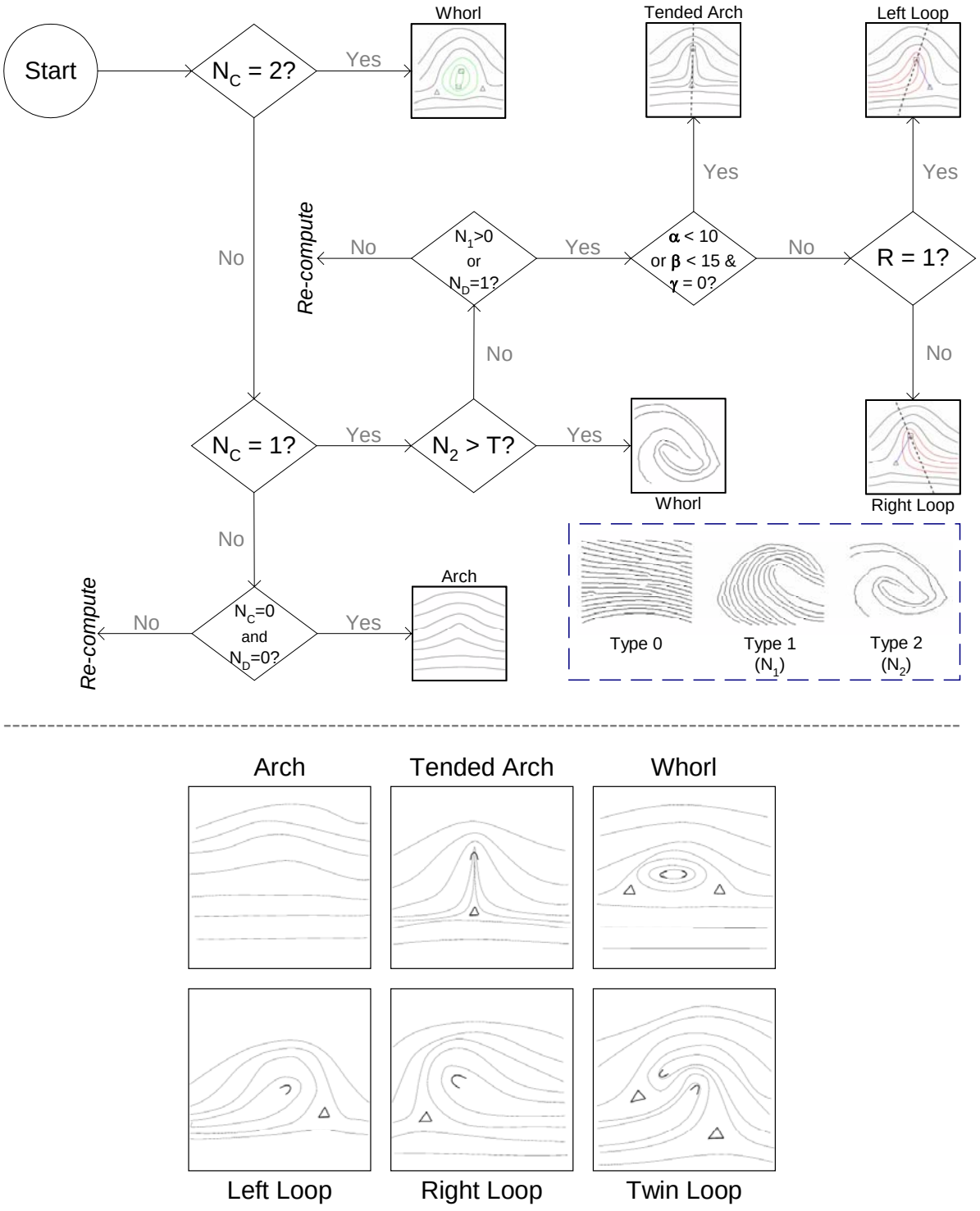


Fig. 2.8: a) Flowchart of fingerprint classification algorithm [58];
b) Cores (⊂) and Deltas (Δ) in FP images belonging to different classes [63]

The orientation field determined from the input image may not be very accurate and the extracted ridges may contain too many features and cannot be therefore directly used for fingerprint classification. The ridge verification stage assesses the

reliability of extracted ridges based upon the length of each connected ridge segment and its alignment with other adjacent ridges. Parallel adjacent subsegments typically indicate a good quality fingerprint region; the ridge/orientation estimates in these regions are used to refine the estimates in the orientation field/ridge map.

Computational steps are as follows (see also Fig. 2.8a):

1. **Singular Points**: The *Poincare Index* [58] on the orientation field is used to determine the number of delta (N_D) and core (N_C) points in a fingerprint. A digital closed curve Ψ about 25 pixels long around each pixel is used to compute the *Poincare Index* as defined below:

$$Poincare(i,j) = \frac{1}{2\pi}\sum_{k=0}^{N_\Psi}\Delta(k), \tag{2.1}$$

where

$$\Delta(k) = \begin{cases} \delta(k) & \text{if } |\delta(k)| < (\pi/2) \\ \pi + \delta(k) & \text{if } \delta(k) \le (-\pi/2) \\ \pi - \delta(k) & \text{otherwise} \end{cases} \tag{2.2}$$

$$\delta(k) = \mathrm{O}'(\Psi_x(i'),\Psi_y(i')) - \mathrm{O}'(\Psi_x(i),\Psi_y(i)) \tag{2.3}$$

$i' = (i+1) \mod N_\Psi$,

O is the orientation field, $\Psi_x(i)$ and $\Psi_y(i)$ denote coordinates of the i^{th} point on the length of arc of the parameterized closed curve Ψ.

2. **Symmetry**: The feature extraction stage also includes the estimation of position of an axis (see Fig. 2.8a – Tended Arch, Left Loop and Right Loop) locally symmetric to the ridge structures at the core, and computation (a) α, angle between the symmetry axis and the line segment joining core and delta points, (b) β, average angle difference between the ridge orientation and the orientation of the line segment joining the core and delta points, and (c) γ, the number of ridges crossing the line segment joining core and delta points. The relative position R, of delta point with respect to the symmetry axis is determined as follows: R = 1 if the delta point is on the right side of the symmetry axis, otherwise R = 0.

3. **Ridge Structure**: The classifier utilizes not only the information on orientation but also on structure of extracted ridges. This feature summarizes the overall nature of the ridge flow in the fingerprint. In particular, it classifies each ridge of the fingerprint into three categories:

- Type 0 Nonrecurring ridges: the ridges which do not curve very much.
- Type 1 Recurring ridges: ridges which curve approximately by π angle.
- Type 2 Fully recurring ridges: ridges which curve by more than π angle.

The distribution of fingerprint classes is following [58, 42]: Plain Arch = 25,3%; Tended Arch = 10,9%; Left Loop = 21,6%; Right Loop = 20,4% and Whorl = 21,7%. Approximately 0,1% of fingerprints are unclassifiable – they have a special pattern combination of more classes. Compare with the values on the page 24 in this work.

Another possibility to classify the fingerprints is the neural network method. This method has the following steps [42, 114]:

1) **Ridge-Valley Orientation**. This step detects, at each pixel location of the fingerprint image, the local orientation of the ridges and furrows of the finger surface, and produces an array of regional averages of these orientations. This is the basic feature extractor of the classification. The routine is based on the ridge-furrow fingerprint binarizer, as described in [42].

2) **Registration** is a process that the classifier uses in order to reduce the amount of variation of the translation between similar orientation arrays. If the arrays from two fingerprints are similar except for the translation, the feature vectors that subsequent processing steps will produce from these orientation arrays can differ very much because of the translation. This problem can be improved by registering each array (finding a consistent feature and essentially translating the array, bringing that feature to standard location). Description of registration process could be found in [42].

3) **Karhunen-Loève Transformation**. The size of registered orientation array representing each fingerprint is about 1 000 elements. The size of these arrays makes them computationally impractical for use as feature inputs into either of the neural network classifiers. It would be helpful to transform these high-dimensional feature vectors into low-dimensional ones in such a way that would not be detrimental to the classifiers. A version of *Karhunen-Loève Transform.* which reduces an original feature vector u (an orientation array) to a vector w of n elements can then be defined as $w = \Psi^t u$. A more detailed description can be found in [42, 22].

4) **Probabilistic Neural Network Classifier**. This step takes as its input the low-dimensional feature vector that is the output of *the Karhunen-Loève Transformation* and determines the class of the fingerprint. The algorithm classifies an incoming feature vector by computing the value, at its point in feature space, of spherical Gaussian kernel functions, centered at each of a large number of stored prototype feature vectors. For each class, activation is made by adding up the values of the kernels centered at all prototypes of that class; the hypothesized class is then defined to be the one whose activation is largest. The activations are all positive, being sums of exponentials. Dividing each of the activations by the sum of all activations produces a vector of normalized activations which, as Specht pointed out, can be used as estimates of the posterior probabilities of the several classes. In mathematical terms, the above definition of PNN classifier can be written as follows:

$$a_i = \sum_{j=1}^{M_i} e^{\left(-\beta \cdot (w - x_j^{(i)})^t \cdot (w - x_j^{(i)})\right)}, \tag{2.4}$$

where $\tilde{a}_h = \dfrac{a_h}{\sum_{i=1}^{N} a_i}$ and (2.5)

N is number of classes; M_i is number of prototype prints of class i; $x_j^{(i)}$ is feature vector from the j^{th} prototype print of the class i; w is feature vector of the print to be

classified; β is a smoothing factor; a_i is an activation for the class i; $\tilde{a}_h$ is normalized activation for the class i and h is the hypothesized class (i for a_i is of the greatest value).

2.2.3 Minutiae Extraction

The purpose of minutiae extraction is to extract representative features, called minutiae, from the input fingerprint images. Generally, this representation [47] should:

- retain the discriminating power of raw digital fingerprint images,
- be compact,
- be amenable to matching algorithms,
- be robust with regard to noise and distortions, and
- be easy to compute.

The pattern of the minutia details of a fingerprint forms a valid representation of the fingerprint. It is compact, amenable to matching algorithms, robust to noise and distortions, and easy to compute. However, most of the 150 types [47] of minutia details in fingerprint images are not stable and cannot be reliably identified. In an automatic fingerprint matching, only the two most prominent types of minutiae details are used for their stability and robustness: a) *ridge ending* and b*) ridge bifurcation*. Each minutia is completely characterized by the following parameters:

- type,
- *x*-coordinate,
- *y*-coordinate, and
- orientation (gradient).

Typically, in a live-scan fingerprint image of good quality, there are about 50-100 minutiae. A good minutiae extraction algorithm should be both reliable and efficient. When the quality of a fingerprint image is good, the ridges and furrows which alternate and flow in a locally constant direction are well-defined and clearly differentiated from each other. In such situations, ridge endings and ridge bifurcations, which are essentially the distinctive anomalies of ridges, can be easily identified and precisely located from the binary ridges.

Minutiae Extraction Algorithm

In order to introduce the minutiae extraction algorithm, a list of notations and some basic definitions are included below [47]:

A *gray-level fingerprint image*, I, is defined as an $N \times N$ matrix (square images are considered), where $I(i, j)$ represents the intensity of the pixel at the i^{th} row and j^{th} column.

An *orientation field,* O, is defined as an $N \times N$ image, where $O(i, j)$ represents the local ridge orientation (direction) at the pixel (i, j) – see Fig. 2.7 (Orientation Field).

The local ridge orientation is usually specified for a region (block) rather than at every pixel; an image is divided into a set of $w \times w$ non-overlapping blocks and a single local ridge orientation is defined for each block. Note that in a fingerprint image, the ridges oriented at 0° and the ridges oriented at 180° in a local neighborhood are not differentiated from each other (the orientation for angles from 180° to 360° is transferred to angles from 0° to <180°).

A *ridge map*, R, is an $N \times N$ binary image, where $R(i, j) = 1$ indicates that the pixel (i, j) is a ridge pixel and $R(i, j) = 0$ indicates that the pixel (i, j) is not a ridge pixel – see Fig. 2.7 (Extracted Ridges). A ridge in a ridge map is an 8-connected component. A thinned ridge has a width of 1 pixel and a *thinned ridge map*, TR, consists of thinned ridges – see Fig. 2.7 (Thinned Ridges).

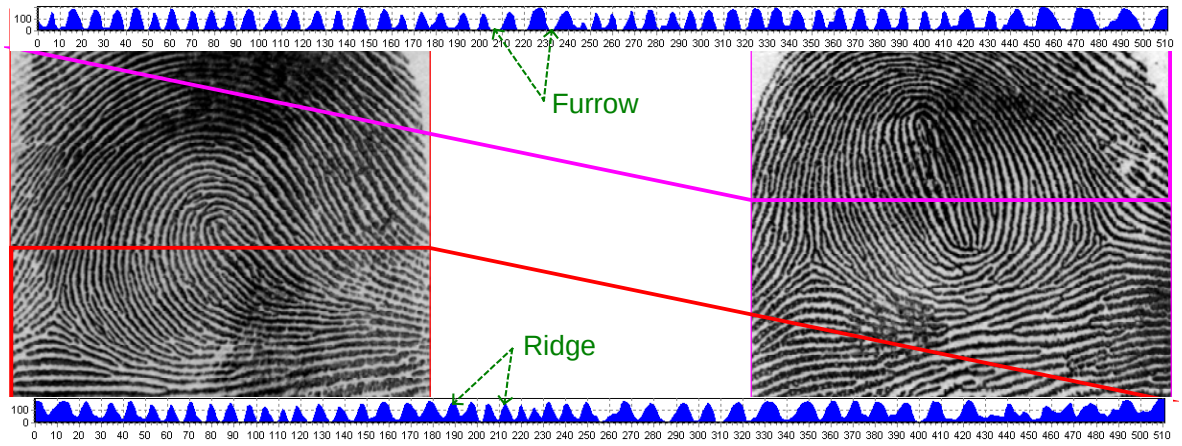


Fig. 2.9: Cross section of the images (comparison to 2D sine wave)

Ridge detection. An important property of the ridges is that the gray-level values on ridges attain their local maxima along a direction that is orthogonal to the local ridge orientation and the gray-level values of furrows attain their local minima along the same direction. Locally, ridges and furrows run in parallel to each other and the gray-level values in the orthogonal direction form a two-dimensional sine wave (see Fig. 2.9). Therefore, certain pixels can be identified as ridge pixels in a local neighborhood based on this characteristic. In the minutiae detection algorithm, a fingerprint image is at first analyzed using two masks $h_t(i, j; u, v)$ and $h_b(i, j; u, v)$, of size $L \times H$, respectively. These two masks are essentially the same except that one is rotated by 180° with respect to the other (see Figure 2.10) [47]:

$$h_t(i,j;u,v) = \begin{cases} -\frac{1}{\sqrt{2\pi}\delta} e^{-\frac{u^2}{\delta^2}} & \text{if } u = (v \cdot \cot(\mathrm{O}(i,j)) - \frac{H}{2\cos(\mathrm{O}(i,j))}, v \in \Omega \\ \frac{1}{\sqrt{2\pi}\delta} e^{-\frac{u^2}{\delta^2}} & \text{if } u = (v \cdot \cot(\mathrm{O}(i,j))), v \in \Omega \\ 0 & \text{otherwise} \end{cases} \quad (2.6)$$

$$h_b(i,j;u,v) = \begin{cases} -\frac{1}{\sqrt{2\pi}\delta} e^{-\frac{u^2}{\delta^2}} & \text{if } u = (v \cdot \cot(\mathrm{O}(i,j)) + \frac{H}{2\cos(\mathrm{O}(i,j))}, v \in \Omega \\ \frac{1}{\sqrt{2\pi}\delta} e^{-\frac{u^2}{\delta^2}} & \text{if } u = (v \cdot \cot(\mathrm{O}(i,j))), v \in \Omega \\ 0 & \text{otherwise} \end{cases} \quad (2.7)$$

$$\Omega = \left[-\left| \frac{L \sin(\mathrm{O}(i,j))}{2} \right|, \left| \frac{L \sin(\mathrm{O}(i,j))}{2} \right| \right] \quad (2.8)$$

where $\mathrm{O}(i, j)$ represents the local ridge direction at the pixel (i, j). These two masks are capable to accentuate the local maximum gray-level values along a direction that is orthogonal to the local ridge orientation. They can also adaptively smooth the fingerprint images along the local ridge orientation and thus enhance the ridges. The smoothing effect depends on the value of δ. The larger the value of δ, the more robust are the filters with regard to noise but they are also more sensitive to highly curved ridges.

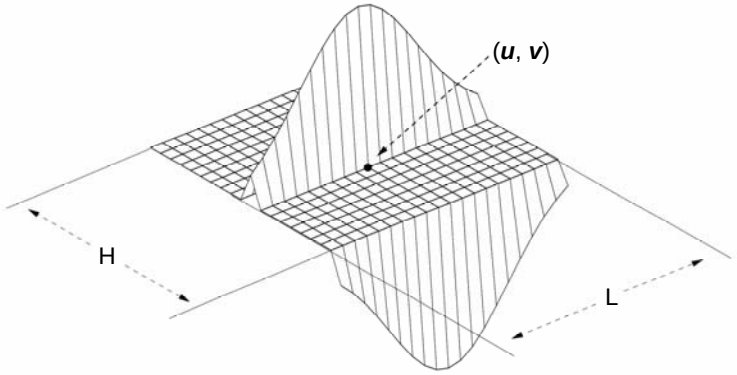


Fig. 2.10: Ridge filter $h_t(i, j; u, v)$ [47]

Minutiae detection is a trivial task when an ideal thinned ridge map *TR* is available. Following conditions can determine the types of minutiae (see Fig. 2.11):

- If $(TR(i, j) = 1 \;\&\; \sum_{u=-1}^{1}\sum_{v=-1}^{1} TR(i+u, j+v) = 2) \Rightarrow$ the pixel (i, j) is a *ridge ending*.
- If $(TR(i, j) = 1 \;\&\; \sum_{u=-1}^{1}\sum_{v=-1}^{1} TR(i+u, j+v) > 3) \Rightarrow$ the pixel (i, j) is a *ridge bifurcation*.

However, the presence of undesired spikes and breaks in a thinned ridge map can lead to the situation in which many spurious minutiae are detected. Therefore, before the minutiae detection, a smoothing procedure is applied to remove spikes

and to join broken ridges [47, 22]. The ridge smoothing algorithm [47] uses the following heuristics:

- If the angle formed by a branch and the trunk ridge is greater than T_{Lower} (=70°) and smaller than T_{Upper} (=110°) and the length of the branch is smaller than T_{Branch} (=20 pixels), then the branch is removed.
- If a break in a ridge is shorter than T_{Break} (=15 pixels) and no other ridges pass through it, then the break is connected.

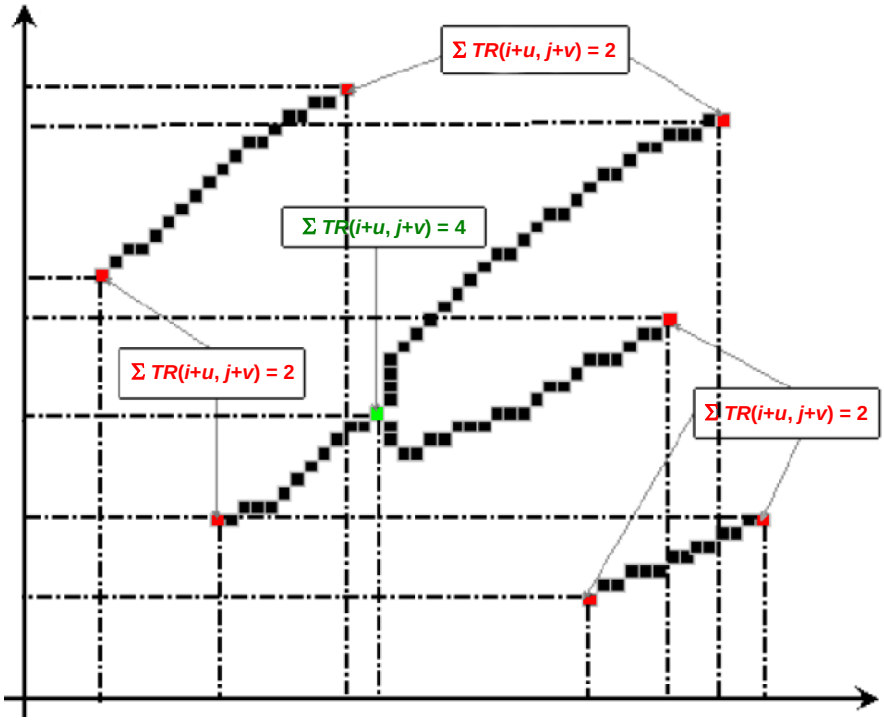


Fig. 2.11: Minutiae Extraction

For each detected minutiae, the following parameters are recorded: type; *x*-coordinate; *y*-coordinate and orientation (gradient) which is defined as the local ridge orientation of the associated ridge (see Fig. 2.6).

Of course, there are some other methods for fingerprint recognition. Not only minutiae can be used for the comparison of two fingerprints. One example of other methods is the correlation-based fingerprint verification [7]. Unlike the traditional minutiae-based systems, this system directly uses the richer gray-scale information of the fingerprints. The correlation-based fingerprint verification system at first selects appropriate templates in the primary fingerprint, uses template matching to locate them in the secondary print, and compares the template positions of both fingerprints.

More detailed description of such non-minutiae-based systems exceeds the scope of this work because this work is limited to minutiae-based systems and further computations reflect this limitation.

2.3 Actual Solutions

2.3.1 Fingerprint Technology

Below are indicated some fingerprint technology solutions available in the market which use special arts of secrecy or generation with the combination of biometrics. In this subchapter, some solutions of other companies will be mentioned. The classical access and special dactyloscopic systems are not considered. However, no results of practical tests of mentioned systems have been published and therefore no such results can be shown in this chapter.

Giesecke & Devrient
security at work.

The company Giesecke & Devrient developed a system that uses a fingerprint as an instrument for secrecy encryption. The secrecy (a code word) is multiplied by a generation matrix of some redundancy code. Then the fingerprint technology is used to protect this secrecy. As a result, a biometrically enciphered secrecy (see Fig. 2.12) is generated. By the inverse algorithm, a fingerprint from the same finger is used to decipher the enciphered secrecy. With regard to the impossibility to acquire the same fingerprint and to generate the same features, a redundancy code has been used. In the next stage, this deciphered secrecy is multiplied by the control matrix of the self-reconstructing code and the sequence errors are corrected. As a result of the inverse stage, the original secrecy (the same code word) is computed or reconstructed. The company Giesecke & Devrient has very good results with iris technology. Of course, there are high requirements on the iris images. What concerns fingerprints, it is very difficult to find always the same reference point. The code word should have the size of approximately 100 bits [33].

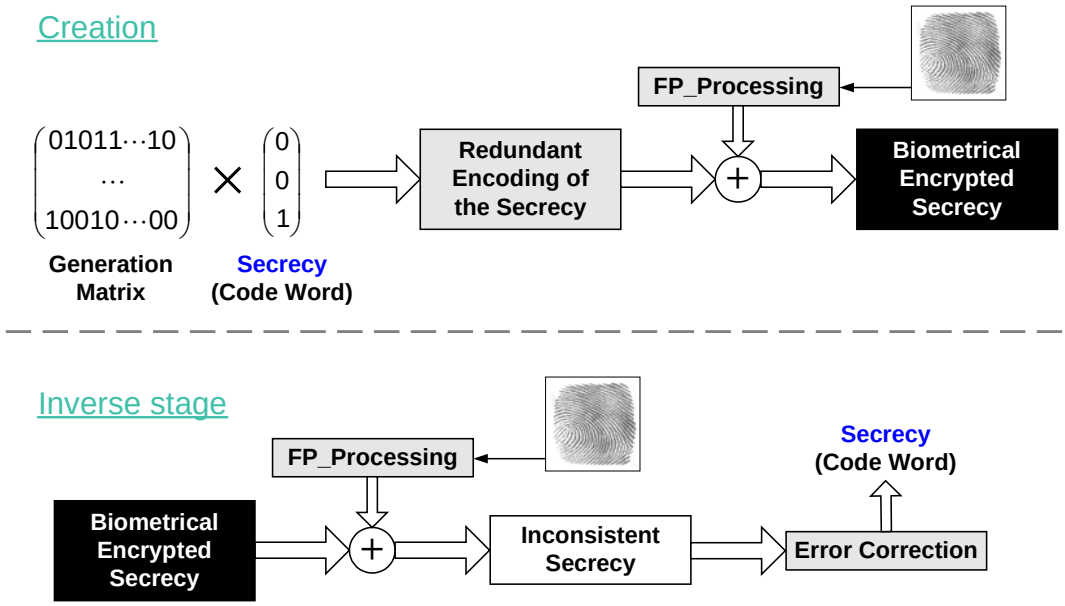


Fig. 2.12: Solution of the company Giesecke & Devrient

ITSI Integrated Technology Solutions, Inc

The solution of the company ITSI is oriented more to police investigation. This solution uses fingerprints as some art of signature for documents. When a trespasser is caught by the police, a police report is written in the field and a fingerprint of this trespasser is scanned on spot. The fingerprint is attached to the end of the document. Then the whole document (considered as one unit) is sent wireless to the police station. Police officers at that station make a search in the fingerprint database, in terms of fingerprint identification. If the data found in the fingerprint database match with the statements in the police report, the validity of the report data is confirmed, or otherwise refused. The scheme of this functionality is shown in the Fig. 2.13 [33]. This solution was presented with the sensor Identix, encryption procedure BlowFish, Silanis Electronic Signature System and Microsoft Word.

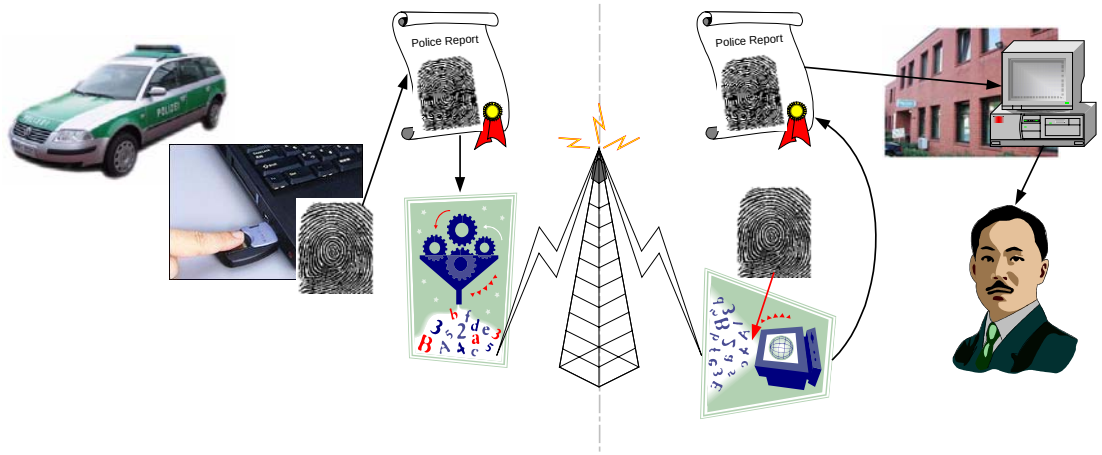
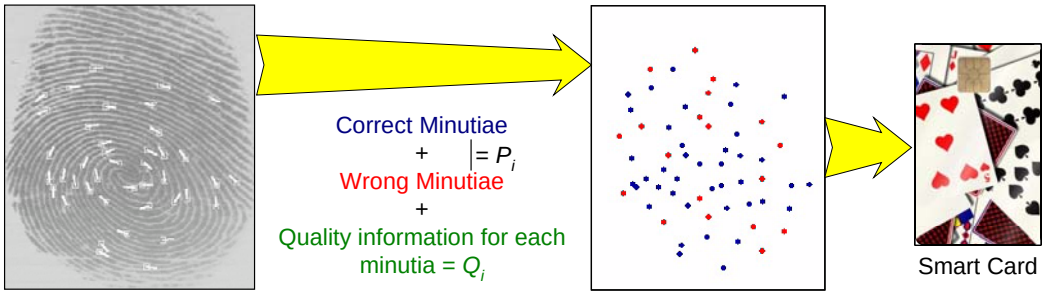


Fig. 2.13: Solution of the company ITSI

GEMPLUS beyond smart

The solution of the company Gemplus focuses more to confusion of the attacker than to data protection. Appropriate minutiae are found in the actual scanned fingerprint. But then a computer adds wrong minutiae to the correct set. This resulting set is then stored on the smart card. In the inverse process, a fingerprint is acquired and respective minutiae are extracted. Then the template with confused minutiae set needs to be downloaded from the smart card. The computer adds again wrong minutiae to the acquired set and the two sets (including both correct and wrong minutiae) are then compared using some minutiae-based comparison algorithm. If the match is found, the verification is successful. Both stages can be seen in the Fig. 2.14 (above is the generation stage and below the inverse stage). The template is compatible for all CBEFF systems.



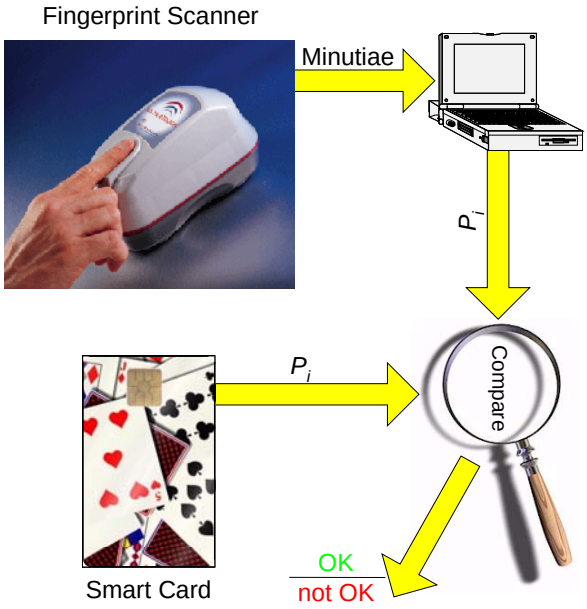


Fig. 2.14: Solution of the company Gemplus

bioscrypt on board

The company Bioscrypt (formerly Mytec Technologies Inc.) introduced the Biometric Encryption system [100, 33], which can protect *N*-bit long key using fingerprint information. This method is based on the image filtering and correlation; it does not use minutiae. Two stages of this Biometric Encryption system are presented in Fig. 2.15 and Fig. 2.16. In both figures, the overview of each process is shown at the top and below is the detailed scheme. In both stages the fingerprint is processed and some pattern is generated. This fingerprint pattern is linked with the key by a link algorithm. As the result, a user data set is recorded. In the inverse stage, this user data set is reused for key reconstruction.

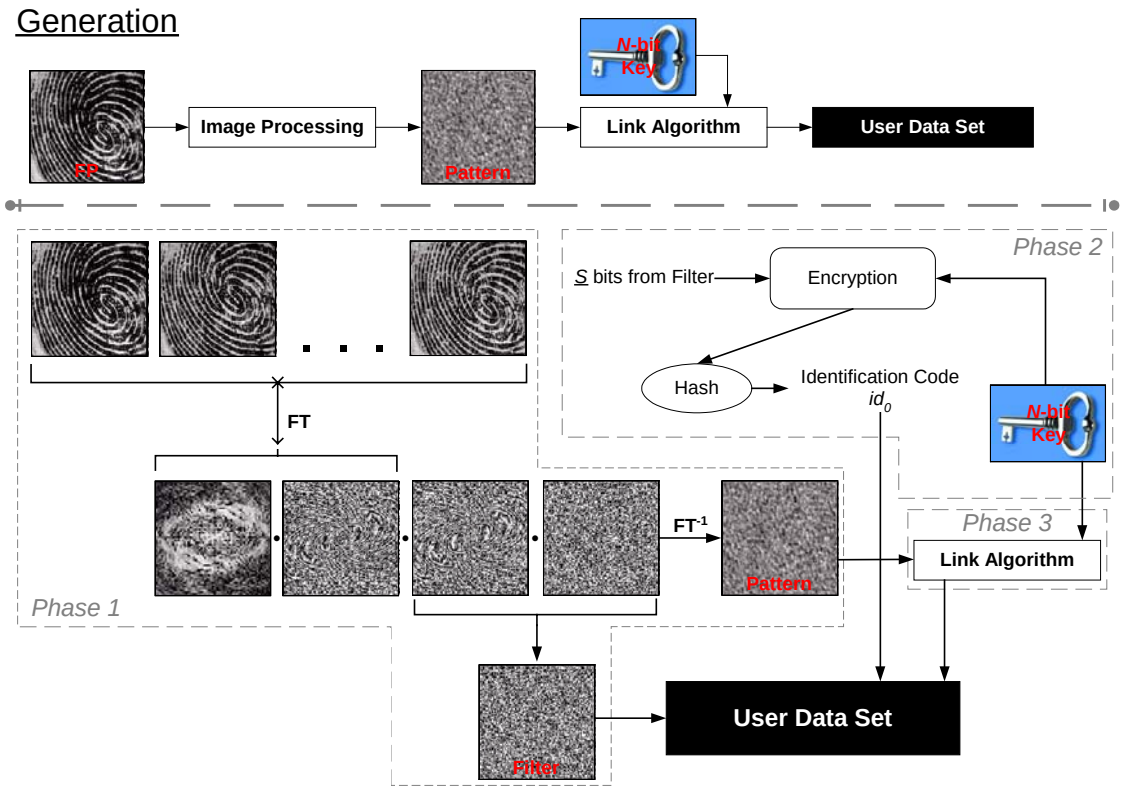


Fig. 2.15: Overview of the enrollment process for Biometric Encryption

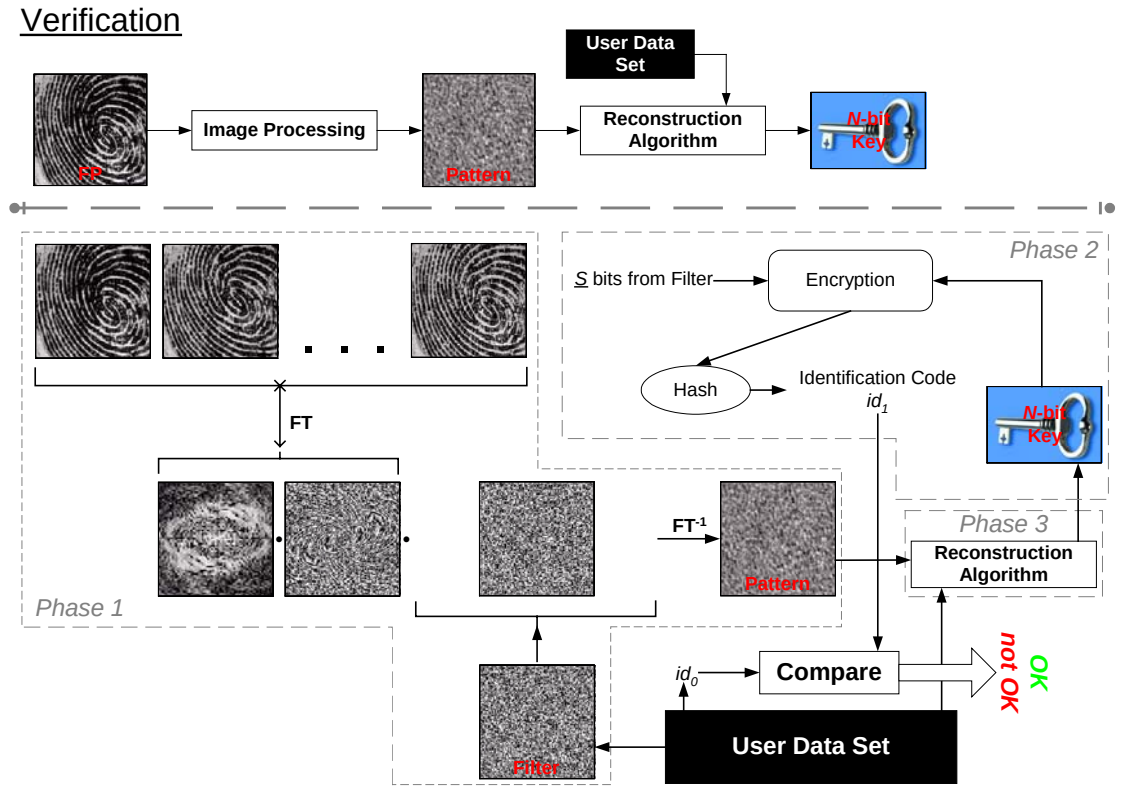


Fig. 2.16: Overview of the verification process for Biometric Encryption

The phases in the Fig. 2.15 have the following description [100]: Phase 1 – Image Processing (it combines a series of input fingerprint images with a random array to create two output arrays $H_{Stored}(u)$ and $c_0(x)$); Phase 2 – Key Linking (it links a cryptographic key k_0 with the pattern $c_0(x)$ via the link algorithm); Phase 3 – Identification Code Creation (it creates an identification code id_0 derived from the key k_0). The following phases can be recognized in the Fig. 2.16 [100]: Phase 1 – Image Processing (it combines $H_{Stored}(u)$ with a new series of input fingerprint images to create an output pattern $c_1(x)$); Phase 2 – Key Retrieval (it extracts a key k_1 from $c_1(x)$ using the retrieval algorithm); Phase 3 – Key Validation (it validates k_1 by creating a new identification code id_1 and comparing it with id_0).

Certain devices from various companies include some art of biometric encryption. Some devices of this type with an integrated fingerprint sensor are shown in Fig. 2.17. The fingerprint is used to protect the stored data. Sometimes, there is a special encryption that processes the information from the fingerprint, but the purpose is only to protect such data against unauthorized access. The following devices are shown in Fig. 2.17 (from left to right): iKey™ SuperToken (Rainbow Technologies), FIU-810 Puppy (Sony), ClipDrive Bio™ (ROG GmbH), Victoria 120 (Thanko Ltd.), ThumbDrive® Touch (ThumbDrive®).

Fig. 2.17: Different solutions using fingerprint to protect data

2.3.2 Other Technologies

Voice. Particular steps of the voice based solution are shown in the Fig. 2.18. There are three main phases in the voice (speech) recognition. The first one is recording of the speech signal. The second one is pre-processing of such signal (framing, windowing, pre-emphasizing and deletion of non-speech signal using Voice Activity Detection (VAD)). The last phase is feature extraction (generation of LPC coefficients, MFCC and Speaker Dependent Frequency Cepstrum Coefficients). The outputs of the last phase are appropriate features which can be used as biometric key for cryptographic tasks [81, 33, 30].

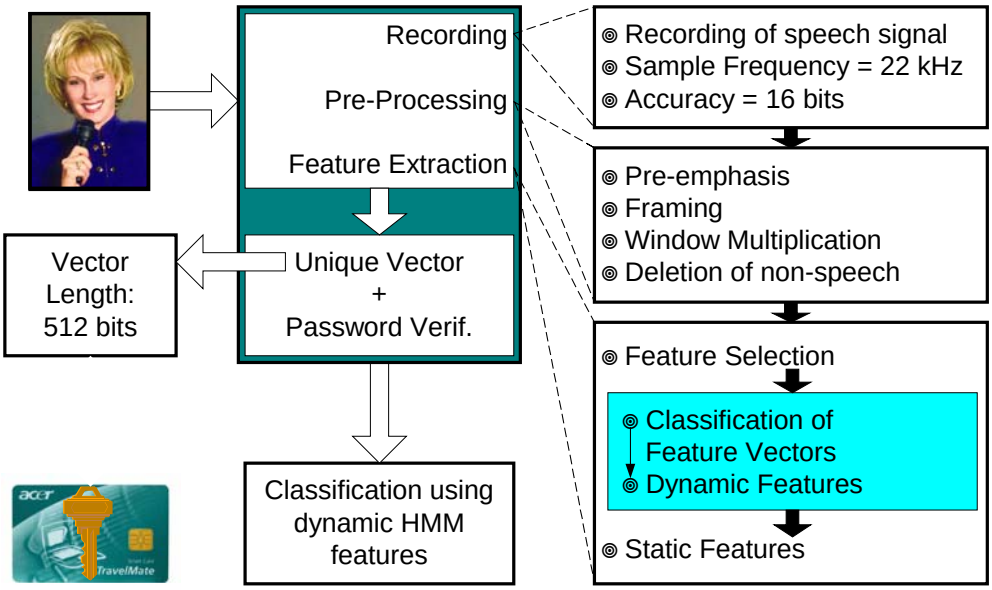


Fig. 2.18: Solution for key generation using voice

Face. This solution comes from the Multimedia University in Malaysia. The face is used as an input for key generation. The whole scheme includes four main steps (see Fig. 2.19): 1. Image pre-processing (finding and positioning of the face in the image, Hambridge Framing and reduction of data size and sensibility against variations), 2. Biometric Eigenanalyse (*Sirovich-Kirby* Algorithm), 3. Creation of discrete biometric data with a Token (biometric information is transformed to a bit set {0,1} and a bit string is generated), 4. Interpolation of the Key with a Token [33].

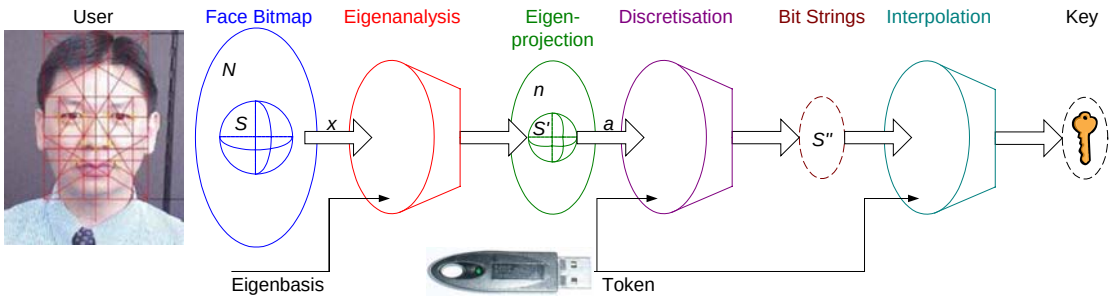


Fig. 2.19: Solution for key generation using face imaging

3. Strength of Fingerprint Information

Although the word “fingerprint” is popularly perceived as a synonym for individuality, the uniqueness of fingerprints is not a proven fact but an empirical observation [71]. With the accepted widespread use of fingerprints, however, there is a rightfully growing public concern about the scientific basis associated with the uniqueness of fingerprints. Widespread doubts of general public in this regard would have disastrous consequences, especially if fingerprints are to be commonly used for unambiguous identification of a person due to their efficiency, convenience and reliability in fighting constantly increasing identity fraud in the society. Furthermore, automated fingerprint matching systems do not use the whole differentiating information contained in the fingerprints, but only sectional representation extracted by a machine unsupervised by human fingerprint experts [71].

The Chapter 3.2 examines answers to some of these questions of fingerprint uniqueness. More specifically, it presents some results from the examination of uniqueness of fingerprint information in the context of an automated fingerprint matching system in detail and lays out the implications of a fingerprint uniqueness model in terms of verification and identification systems. The Chapter 3.3 contains my own estimations of information content (cryptographic strength) in fingerprints, which directly corresponds to the probability of success of a brute force attack on a fingerprint template, that is, the likelihood of generating a synthetic (false) fingerprint input template that would match the registered fingerprint template of an enrolled user.

3.1 Basics of Entropy and Attack Possibilities

3.1.1 Shannon’s Theory

There are two basic approaches to discussions on the security of cryptographic systems or, in short, cryptosystems [101]:

- **Computational security** [101]. This notion concerns the computational effort necessary to break a cryptosystem. We could call a cryptosystem as computationally secure if the best algorithm for breaking it requires at least *N* operations, where *N* is some specified very large number. The problem is that no known practical cryptosystem can be proved to be secure under this definition. In practice, people will call a cryptosystem “computationally secure” if the best known method of breaking the system requires an unreasonably large amount of computer time. Another approach is to provide evidence of computational security by reducing the security of the cryptosystem to some well-defined problem that is considered to be difficult.
- **Unconditional security** [101]. This notion concerns the security of cryptosystems when there is no limitation on the amount of computation that an attacker is allowed to do. A cryptosystem is defined to be unconditionally secure if it cannot be broken, even with infinite computational resources.

The unconditional security of a cryptosystem obviously cannot be studied from the point of view of computational complexity, since we would need to allow the computational time to be infinite. The appropriate framework in which it is possible to study the unconditional security is the theory of probability.

DEFINITION 3.1 Suppose X and Y are random variables. We denote the probability that X takes on the value x by $p(x)$, and the probability that Y takes on the value y by $p(y)$. The joint probability $p(x,y)$ is the probability that X takes on the value x and Y takes on the value y. The conditional probability $p(x|y)$ denotes the probability that X takes on the value x given that Y takes on the value y. The random variables X and Y are said to be independent if $p(x,y) = p(x)\cdot p(y)$ for all possible values x of X and y of Y.

Joint probability can be related to conditional probability by the formula

$$p(x,y) = p(x \mid y)\cdot p(y) \tag{3.1}$$

Interchanging x and y, we have that

$$p(x,y) = p(y \mid x)\cdot p(x) \tag{3.2}$$

From these two expressions, we immediately obtain the following result, which is known as *Bayes' Theorem* [101].

THEOREM 3.1 (*Bayes' Theorem*)

If $p(y) > 0$, then

$$p(x \mid y) = \frac{p(x)\cdot p(y \mid x)}{p(y)} \tag{3.3}$$

COROLLARY 3.1

X and Y are independent variables if and only if $p(x \mid y) = p(x)$ for $\forall\, x, y$.

We assume that a particular key is used for only one encryption [101]. Suppose that there is a probability distribution on the plaintext (input data) space, P. We denote the hypothetical probability that plaintext x occurs by $p_P(x)$. We also assume that the key k is chosen using some fixed probability distribution (often a key is chosen randomly). Let us denote the probability that a key k is chosen by $p_K(k)$. We make the assumption that the key k and the plaintext x are independent events.

The two probability distributions on P and K induce a probability distribution on C. Indeed, it is not difficult to compute the probability $p_C(y)$ that y is the ciphertext which is transmitted. For a key $k \in K$, let us define

$$C(k) = \{e_k(x); x \in P\} \tag{3.4}$$

That is, $C(k)$ represents the set of possible ciphertexts if k is the key. Then for every $y \in C$, we have that

$$p_C(y) = \sum_{\{k:y \in C(k)\}} p_K(k) \cdot p_P(d_k(y)) \tag{3.5}$$

We also observe that for any $y \in C$ and $x \in P$, we can compute the conditional probability as

$$p_C(y \mid x) = \sum_{\{k:x=d_k(y)\}} p_K(k) \tag{3.6}$$

It is now possible to compute the conditional probability using *Bayes' Theorem*. The following formula is obtained [101]:

$$p_P(x \mid y) = \frac{p_P(x) \cdot \sum_{\{k:x=d_k(y)\}} p_K(k)}{\sum_{\{k:y \in C(k)\}} p_K(k) \cdot p_P(d_k(y))} \tag{3.7}$$

3.1.2 Entropy

In the previous section, we restricted our attention to the special situation where a key is used for only one encryption. We now want to look at what happens when more and more plaintexts are encrypted using the same key, and how likely a cryptanalyst will be able to carry out a successful ciphertext-only attack, given sufficient time.

The basic tool in studying this question is the idea of *entropy*, a concept from information theory introduced by *Shannon* in 1948. Entropy can be considered as a mathematical criterion of information or uncertainty, and can be computed as a function of probability distribution.

Suppose we have a random variable X which takes on a finite set of values according to a probability distribution $p(X)$. What is the information gained by an event which takes place according to distribution $p(X)$? Equivalently, if the event has not (yet) taken place, what is the uncertainty about the outcome? This quantity is called the *entropy* of X and is denoted by $H(X)$.

An event occurring with the probability p might be encoded by a bit string with the length of approximately $\log_2(p)$. Given an arbitrary probability distribution p_1, p_2, ..., p_n for a random variable X, we take the weighted average of the quantities $-\log_2(p_i)$ to be our criterion of information. This motivates the following formal definition.

DEFINITION 3.2 Suppose X is a random variable which takes on a finite set of values according to the probability distribution $p(X)$. Then the entropy of this probability distribution is defined to be the quantity [101]

$$H(X) = -\sum_{i=1}^{n} p_i \cdot \log_2(p_i) \tag{3.8}$$

If possible values of X are x_i, $1 \leq i \leq n$, then we have

$$H(X) = -\sum_{i=1}^{n} p(X = x_i) \cdot \log_2(p(X = x_i)) \tag{3.9}$$

3.1.3 Properties of Entropy

In this subchapter, we prove some fundamental results concerning entropy.

DEFINITION 3.3 A real-valued function f is a concave function in an interval I if

$$f\left(\frac{x+y}{2}\right) \geq \frac{f(x)+f(y)}{2} \tag{3.10}$$

for $\forall\, x,y \in I$, f is a strictly concave function in an interval I if

$$f\left(\frac{x+y}{2}\right) > \frac{f(x)+f(y)}{2} \tag{3.11}$$

for $\forall\, x,y \in I$, $x \neq y$.

THEOREM 3.2 (*Jensen's Inequality*)
Suppose f is a continuous strictly concave function on the interval I,

$$\sum_{i=1}^{n} a_i = 1 \tag{3.12}$$

and $a_i > 0$, $1 \leq i \leq n$. Then

$$\sum_{i=1}^{n} a_i \cdot f(x_i) \leq f\left(\sum_{i=1}^{n} a_i \cdot x_i\right) \tag{3.13}$$

where $x_i \in I$, $1 \leq i \leq n$. Further, the equality occurs if and only if $x_1 = x_2 = ... = x_n$.

In the next theorem, we make use of the fact that the function $\log_2(x)$ is strictly concave on the interval $(0, \infty)$.

THEOREM 3.3
Suppose X is a random variable having probability distribution $p_1, p_2, ..., p_n$, where $p_i > 0$, $1 \leq i \leq n$. Then $H(X) \leq \log_2(n)$, with the equality if and only if $p_i = 1/n$, $1 \leq i \leq n$.
The proof can be found in [101].

THEOREM 3.4

$H(X,Y) \leq H(X) + H(Y)$, with the equality if and only if X and Y are independent events.

DEFINITION 3.4 Suppose X and Y are two random variables. Then for any fixed value y of Y, we get a (conditional) probability distribution $p(X \mid y)$. Clearly,

$$H(X \mid y) = -\sum_{x} p(x \mid y) \cdot \log_2(p(x \mid y)) \tag{3.14}$$

We define the conditional entropy $H(X \mid Y)$ to be the weighted average (with respect to the probabilities $p(y)$) of the entropies $H(X \mid y)$ over all possible values y. It has been computed to be

$$H(X \mid Y) = -\sum_{y} \sum_{x} p(y) \cdot p(x \mid y) \cdot \log_2(p(x \mid y)) \tag{3.15}$$

The conditional entropy measures the average amount of information about X that is revealed by Y.

THEOREM 3.5

$$H(X,Y) = H(Y) + H(X \mid Y) \tag{3.16}$$

COROLLARY 3.2

$H(X \mid Y) \leq H(X)$, with the equality if and only if X and Y are independent.

3.1.4 Pseudorandom Bits and Sequences

The security of many cryptographic systems depends upon the generation of unpredictable quantities [77]. Examples can be the secret key in the DES encryption algorithm, the primes p and q in the RSA encryption and digital signatures. In all these cases, the generated quantities must be of sufficient size and “random” in the sense that the probability of any particular value being selected must be sufficiently small to prevent an attacker from gaining advantage through optimizing a search strategy based on such probability.

3.1.5 Attacks

Some possible attacks can deceive biometric systems. In the Table 3.1 you can see the average attack spaces for different methods [99].

Fig. 3.1: Securing of communication to decrease the number of attacks [40]

Tab. 3.1: Examples of attacks [99]

Example	Type of Attack	Average Attack Space
Random 8-char password	Interactive	2^{45}
Dictionary Attack	Off-line	2^{15} to 2^{23}
Practical Off-line Attack	Off-line	2^{40} to 2^{63}
Token with Public Key	Off-line	2^{63} to 2^{116}
Biometric with 1% **FAR**	Team	2^{6}
Biometric with 0,01% **FAR**	Team	2^{12}

Some art of attacks on a cryptosystem are called brute force attacks. This art of attack is represented by the attempt to generate all possible key combinations. One of these keys would be the right one and would be used for deciphering of a ciphertext. Closer description of brute force attack is in the Chapter 3.3.6 and in [113].

3.2 Uniqueness of Fingerprints

Fingerprint based personal identification has been routinely used in forensic laboratories and identification units around the world and it has been accepted by the courts of law which use it as a method of evidence for nearly a century. The Supreme Court (of USA) stated in 1993 that the following five factors should be considered when assessing reliability:

- Whether any particular technique or methodology in question has been subject to a statistical hypothesis testing.
- Whether its error-rate has been established.
- Whether the standards controlling the technique's operations exist and have been maintained.
- Whether it has been peer reviewed, and published.
- Whether it has a general widespread acceptance.

The two fundamental premises on which fingerprint identification is based are [82, 22]:

- Fingerprint details are permanent.
- Fingerprints of an individual are unique.

The validity of the first premise has been established based on the anatomy and morphogenesis of friction ridge skin. The second premise is debatable. The notion of fingerprint uniqueness has been widely accepted, based on manual inspection (by dactyloscopic experts) of millions of fingerprints.

What do we mean by fingerprint uniqueness? The problem of fingerprint uniqueness can be formulated in many different ways. Two typical formulations are [82]:

- The uniqueness problem may be considered as determining the probability that any two individuals may have sufficiently similar fingerprints in a given target population.
- Given a sample fingerprint, determine the probability of finding a sufficiently similar fingerprint in a target population.

The problem of uniqueness is defined in a study [82] as the probability of a false association: For given two fingerprints from two different fingers, determine the probability that they are “sufficiently” similar. If two fingerprints originating from two different fingers are examined at a very high level of detail (resolution), we can find out that the fingerprints are indeed different. However, most human experts and automatic fingerprint identification systems (AFIS) declare that the fingerprints originate from the same source if they are “sufficiently” similar. The degree of similarity depends on typical (intra-class) variations observed in multiple impressions of a finger.

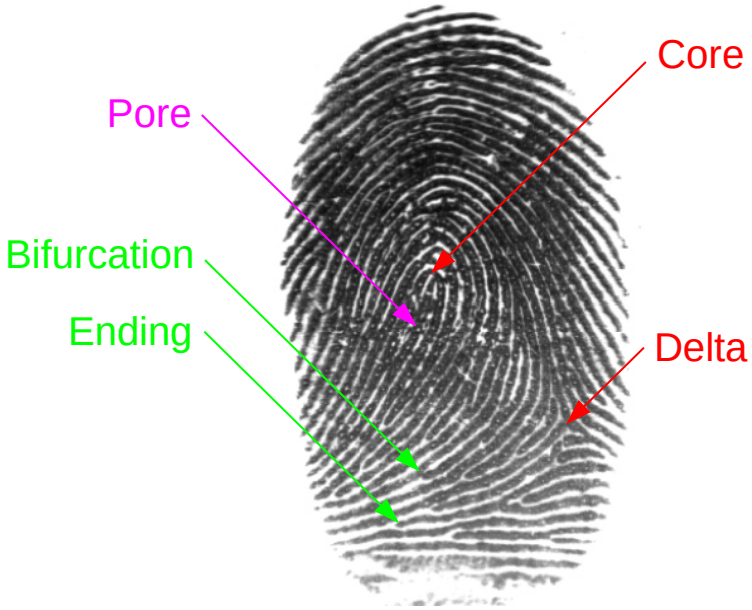


Fig. 3.2: A fingerprint image with typical features [FVC2004[1], DB1, 103_3]

In order to solve the problem of uniqueness, it is necessary to define the representation of a fingerprint (pattern) and the similarity metric. Fingerprints can be represented by a large number of features including the overall ridge flow pattern, ridge frequency, location and position of singular points (core(s) and delta(s)), type, direction, and location of minutiae points and ridge counts between the pairs of minutiae (see Figure 3.2). The representation of fingerprints minutiae, which is exploited by forensic experts, has been demonstrated to be relatively stable and has been adopted by the majority of automatic fingerprint matching systems. The similarity metric is the number of corresponding minutiae between two minutiae sets (see Figure 3.3 and Figure 3.4).

[1] http://bias.csr.unibo.it/fvc2004/

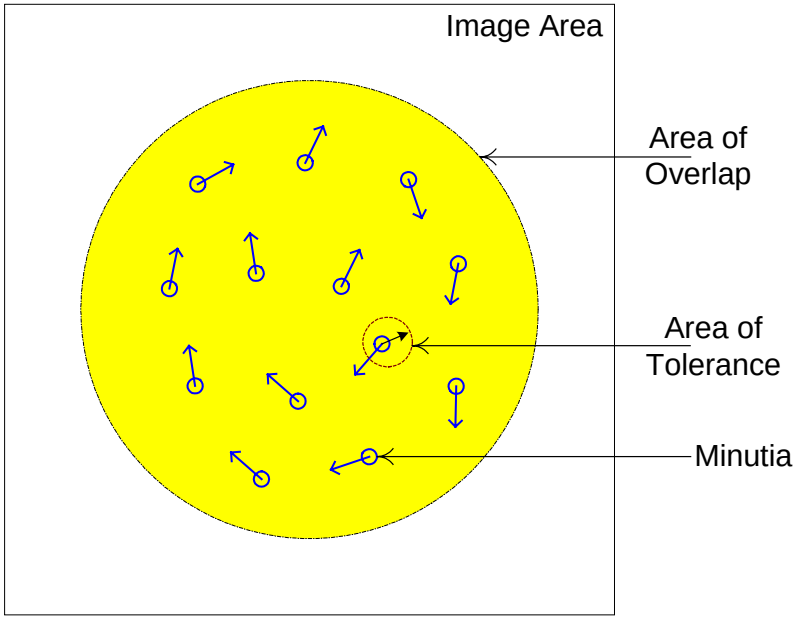


Fig. 3.3: Fingerprint and its minutiae

There are two approaches for determining the uniqueness of the fingerprints. In the *empirical approach*, representative samples of fingerprints are collected and using a typical fingerprint matcher, the accuracy of the matcher on the samples provides an indication of the uniqueness of the fingerprint with respect to the matcher. In the *theoretical approach*[2] to the estimation of uniqueness, all realistic phenomena affecting inter-class and intra-class pattern variations are modelled. Given the similarity metric, it is then possible to estimate the probability of a false association. A theoretical formulation of the fingerprint uniqueness model based on a number of parameters derived from the database of fingerprint images has been proposed.

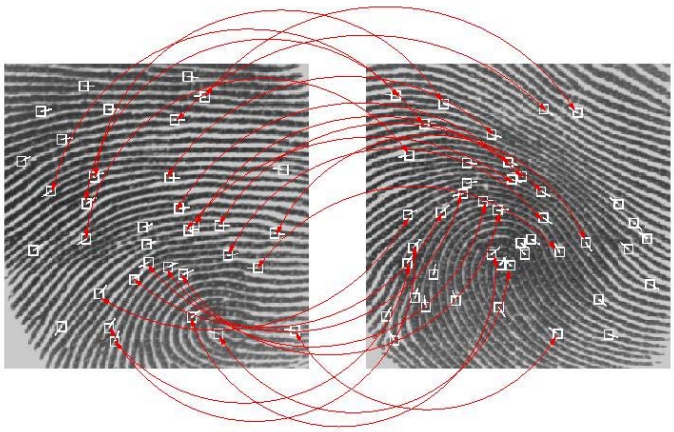


Fig. 3.4: Automatic matching of minutiae

[2] http://bias.csr.unibo.it/research/biolab/bio_tree.html ; “Research Topics” and “Generation” - *SFINGE*

3.2.1 Background

The problem of fingerprint uniqueness was first addressed by *Galton* in 1892 [82, 22], who considered a square region extending across six ridges in a given fingerprint. He assumed that a fingerprint can be covered by 24 (on average) such six-ridge wide independent square regions. *Galton* estimated that he could correctly reconstruct any of the regions with a probability of 1/2 , by looking at the surrounding ridges. Accordingly, the probability of a specific fingerprint configuration, based on the surrounding ridges, is $(1/2)^{24}$. He multiplied this conditional probability with the probability of finding the surrounding ridges to obtain the probability of occurrence of a fingerprint as [82]

$$p(FP\ \ Configuration) = \frac{1}{16} \cdot \frac{1}{256} \cdot \left(\frac{1}{2}\right)^{24} = 1{,}45 \cdot 10^{-11} \tag{3.17}$$

where 1/16 is the probability of occurrence of a specific fingerprint type (such as arch, left loop, right loop, whorl, etc.) and 1/256 is the probability of occurrence of the correct number of ridges entering and exiting each of such 24 square regions. *Galton's* formulation gives the probability that a particular fingerprint configuration in an average size fingerprint will be observed in nature. *Pearson* [82] argued that there could be 36 (6×6) possible minutiae locations within one of *Galton's* six-ridge-square regions, and replaced *Galton's* probability of a six-ridge-square region of 1/2 by 1/36 . Several subsequent models are interrelated and are based on a fixed probability, *p*, for the occurrence of a minutia. Such models compute the probability of a particular *N*-minutiae fingerprint configuration as $p(Fingerprint\ Configuration) = p^N$, where *p* is a fixed probability for the occurrence of a minutia.

Osterburg [82] divided fingerprints into discrete cells with the size 1 *mm* $\times$ 1 *mm*. He computed the frequencies of 13 types of minutiae events (including an empty cell) from 39 fingerprints (8.591 cells) and estimated that 12 ridge endings will match between two fingerprints based on an average fingerprint area of 72 mm^2 with the probability $1{,}25 \cdot 10^{-20}$.

Stoney and *Thornton* [82] critically reviewed earlier fingerprint uniqueness models and attempted to characterize pairwise minutiae dependence. They proposed a linear ordering of minutiae and recursively estimated the probability of an *n*-minutiae configuration from the probability of a (*n* - 1)-minutiae configuration and the occurrence of a new minutiae of certain type/orientation at a particular distance/ridge counts from its nearest minutiae within the (*n* - 1)-minutiae configuration. The probabilities of observing a particular minutiae configuration from different models are compared in Table 3.2 (the assumptions for the Tab. 3.2 are: – an average size fingerprint has 24 regions (*R* = 24) as defined by *Galton*, 72 regions (*M* = 72) as defined by *Osterburg*, and has 36 minutiae on average (*N* = 36)).

Tab. 3.2: Comparison of probability distributions for different models [82]

Author	*p*(*Fingerprint Configuration*)	*N*=36, *R*=24, *M*=72	*N*=12, *R*=8, *M*=24
Galton (1892)	$\frac{1}{16}\times\frac{1}{256}\times\left(\frac{1}{2}\right)^{R}$	$1{,}45\cdot10^{-11}$	$9{,}54\cdot10^{-7}$
Pearson (1930)	$\frac{1}{16}\times\frac{1}{256}\times\left(\frac{1}{36}\right)^{R}$	$1{,}09\cdot10^{-41}$	$8{,}65\cdot10^{-17}$
Henry (1900)	$\left(\frac{1}{4}\right)^{N+2}$	$1{,}32\cdot10^{-23}$	$3{,}72\cdot10^{-9}$
Wentworth & Wilder (1918)	$\left(\frac{1}{50}\right)^{N}$	$6{,}87\cdot10^{-62}$	$4{,}10\cdot10^{-21}$
Cummins & Midlo (1943)	$\frac{1}{31}\times\left(\frac{1}{50}\right)^{N}$	$2{,}22\cdot10^{-63}$	$1{,}32\cdot10^{-22}$
Gupta (1968)	$\frac{1}{10}\times\frac{1}{10}\times\left(\frac{1}{10}\right)^{N}$	$1{,}00\cdot10^{-38}$	$1{,}00\cdot10^{-14}$
Roxburgh (1933)	$\frac{1}{1000}\times\left(\frac{1.5}{10\times2.412}\right)^{N}$	$3{,}75\cdot10^{-47}$	$3{,}35\cdot10^{-18}$
Trauring (1963)	$(0.1944)^{N}$	$2{,}47\cdot10^{-26}$	$2{,}91\cdot10^{-9}$
Osterburg (1980)	$(0.766)^{M-N}\times(0.234)^{N}$	$1{,}33\cdot10^{-27}$	$1{,}10\cdot10^{-9}$
Stoney (1985)	$\frac{N}{5}\times0.6\times(0.5\cdot10^{-3})^{N-1}$	$1{,}20\cdot10^{-80}$	$3{,}50\cdot10^{-26}$

The models discussed above measure the amount of detail in a single fingerprint, i.e., they estimate the probability of a fingerprint configuration. However, these models did not emphasize the intra-class variations in multiple impressions of a finger. The quantifications of fingerprint uniqueness explicitly consider the intra-class variations as the probability of correspondence.

3.2.2 Fingerprint Uniqueness Model

A model for obtaining a realistic and accurate probability of correspondence between fingerprints is introduced in [82]. To estimate the probability of correspondence, the following assumptions should be made [82]:

- Only two types of minutiae details will be considered: ridge endings and ridge bifurcations. The model does not distinguish between the two types of minutiae because it cannot accurately discriminate between each other. Since minutiae can reside only on ridges which follow certain “flow” pattern, the model evaluates the statistical dependence between minutiae directions and locations.

- A uniform distribution of minutiae in a fingerprint will be assumed, with the restriction that two minutiae cannot be very close to each other. This assumption approximates a slightly overdispersed uniform distribution of minutiae as found by *Stoney* [82].
- Correspondence of a minutiae pair is an independent event and each correspondence is equally important.
- The fingerprint image quality is not taken into account, since it is very difficult to assign reliably a quality index to a fingerprint.

The fingerprint correspondence problem involves the process of matching a template fingerprint with an input fingerprint. We assume that a reasonable alignment can be established between the template and the input. The alignment of the input minutiae set with the template minutiae set is done under the condition that respective minutiae correspondences can be determined within a small tolerance. Given an input fingerprint containing *n* minutiae, our goal is to compute the probability that any arbitrary fingerprint (template in a database of fingerprints) containing *m* minutiae will have exactly *q* corresponding minutiae with the input. Since the fingerprint minutiae are defined by their location, (*x*, *y*), and by the angle of the ridge on which they reside, θ, the input and template minutiae sets, *I* and *T*, respectively, are defined as follows [82]:

$$I = \{\{x'_1, y'_1, \theta'_1\}, \{x'_2, y'_2, \theta'_2\}, \ldots, \{x'_n, y'_n, \theta'_n\}\} \tag{3.18}$$

$$T = \{\{x_1, y_1, \theta_1\}, \{x_2, y_2, \theta_2\}, \ldots, \{x_m, y_m, \theta_m\}\} \tag{3.19}$$

A minutia *j* in the input fingerprint is considered as "corresponding to" or "matching" the minutia *i* in the template, if and only if

$$\sqrt{(x'_i - x_j)^2 + (y'_i - y_j)^2} \leq r_0 \tag{3.20}$$

$$\min\left(\left|\theta'_i - \theta_j\right|, 360° - \left|\theta'_i - \theta_j\right|\right) \leq \theta_0 \tag{3.21}$$

where r_0 is the tolerance in distance (see Fig. 3.3) and θ_0 is the tolerance in angle.

Let *A* be the total area of overlap (see Fig. 3.3) between the input and template fingerprints after a reasonable alignment has been achieved. The probabilities that any arbitrary minutiae in the input will match any arbitrary minutiae in the template only in terms of location and only in terms of direction, are expressed by the Equations (3.22) and (3.23), respectively. The Equation (3.22) assumes that (*x*, *y*) and (x_0, y_0) are independent and the Equation (3.23) assumes that θ and θ_0 are independent. Let $\delta_x = x'_i - x_j$, $\delta_y = y'_i - y_j$, and $d_k = \sqrt{\delta_x^2 + \delta_y^2}$, then

$$P(d_k \leq r_0) = \frac{Area\ of\ Tolerance}{Total\ Area\ of\ Overlap} = \frac{\pi \cdot r_0^2}{A} = \frac{C}{A} \tag{3.22}$$

$$P\left(\min\left(\left|\theta'_i - \theta_j\right|, 360° - \left|\theta'_i - \theta_j\right|\right) \leq \theta_0\right) = \frac{Angle\ of\ Tolerance}{Total\ Angle} = \frac{2\theta_0}{360°} \tag{3.23}$$

If the template contains *m* minutiae, then the probability that only one minutia in the input will correspond to any of the *m* template minutiae is expressed by (*mC* / *A*). Now, given two input minutiae, the probability that only the "first" one cor-

responds to any of the *m* template minutiae is the product of the probabilities that the first input minutia has a correspondence (*mC* / *A*) and the second minutia does not have a correspondence (*A* – *mC*) / (*A* – *C*). Thus, the probability that exactly one of the two input minutiae matches any of the *m* template minutiae is 2 × (*mC* / *A*) × ((A – mC) / (A – C)), since either the first input minutia alone may have a correspondence or the second input minutia alone may have a correspondence. If the input fingerprint has *n* minutiae, the probability that exactly one input minutia matches one of the *m* template minutiae is [82]:

$$p(A,C,m,n,1) = \binom{n}{1} \cdot \left(\frac{mC}{A}\right) \cdot \left(\frac{A-mC}{A-C}\right) \tag{3.24}$$

The probability that there are exactly ρ corresponding minutiae between the *n* input minutiae and *m* template minutiae is then expressed by

$$p(A,C,m,n,\rho) = \binom{n}{\rho} \cdot \underbrace{\left(\frac{mC}{A}\right) \cdot \left(\frac{(m-1)C}{A-C}\right) \cdots \left(\frac{(m-\rho-1)C}{A-(\rho-1)C}\right)}_{\rho \text{ terms}} \bullet \tag{3.25}$$

$$\bullet \underbrace{\left(\frac{A-mC}{A-\rho C}\right) \cdot \left(\frac{A-(m-1)C}{A-(\rho+1)C}\right) \cdots \left(\frac{A-(m-(n-\rho+1))C}{A-(n-1)C}\right)}_{(n-\rho) \text{ terms}}$$

The first ρ terms in Eq. (3.25) denote the probability of matching ρ minutiae between the template and the input; and remaining (n-ρ) terms express the probability that (n-ρ) minutiae in the input do not match any minutiae in the template.

Let *M* = (*A* / *C*), assuming *M* to be integer (*A* » *C*) and rearranging, we obtain [82]

$$p(M,m,n,\rho) = \frac{\binom{m}{\rho} \cdot \binom{M-m}{n-\rho}}{\binom{M}{n}} \tag{3.26}$$

The above analysis considers a minutiae correspondence based solely on the minutiae location. The following analysis considers a minutiae correspondence that depends on minutiae directions as well as on minutiae positions. For the sake of this analysis, let assume that the minutiae directions are completely independent of the minutiae positions, so that the matching of minutiae positions and minutiae directions are therefore independent events. Let *I* be such that $P\left(\min\left(\left|\theta_i' - \theta_j\right|, 360° - \left|\theta_i' - \theta_j\right|\right) \le \theta_0\right) = I$ in Eq. (3.23), see [82].

Given *n* input and *m* template minutiae, the probability of ρ minutiae falling into the similar positions can be estimated by the equation (3.26). Once ρ minutiae positions are matched, the probability that *q* ($q \le \rho$) minutiae among them have similar direction can be expressed by

$$\binom{\rho}{q} \cdot (I)^q \cdot (1-I)^{\rho-q} \tag{3.27}$$

where I is the probability of two position-matched minutiae having similar direction and $(1 - I)$ is the probability of two position-matched minutiae taking different directions. Therefore, the probability of matching q minutiae in both position and direction is expressed by

$$p(M,m,n,q) = \sum_{\rho=q}^{\min(m,n)} \left(\frac{\binom{m}{\rho} \cdot \binom{M-m}{n-\rho}}{\binom{M}{n}} \cdot \binom{\rho}{q} \cdot (I)^q \cdot (1-I)^{\rho-q} \right) \tag{3.28}$$

Until now, it was assumed that the minutiae locations are uniformly distributed within the entire fingerprint area. However, the number (or the area) of ridges across all fingerprint types is approximately the same. Since A is the area of the overlap between the template and input fingerprints, the ridges occupy approximately $(A/2)$ of the area, with the other half being occupied by the furrows. Since the minutiae can lie only on ridges, i.e., along a curve of the length (A/w), where w is the ridge period, the value of M in Eq. (3.28) should therefore be changed from $M = (A/C)$ to $M = (A/w)/2r_0$, where $2r_0$ is the length tolerance in minutiae location. This analysis assumes that the ridge direction information/uncertainty is completely captured or expressed by Eq. (3.23).

3.2.3 Parameter Estimation

The proposed uniqueness model [82] has several parameters, namely, r_0, I, w, A, m, n, and q. The value of I further depends on θ_0. The values of r_0, I, and w are estimated in this section for a given sensor resolution.

The value of r_0 should be determined to account for the variation in the different impressions of the same finger. The value of r_0 was found to be 15 pixels for fingerprint images scanned at the resolution of 500 *dpi* [82]. The value for θ_0 was found to be $\theta_0 = 22{,}5°$ [82]. The distribution is $P(\min(|\theta' - \theta|, 360° - |\theta' - \theta|) \leq 22{,}5°) = 0{,}267$, i.e., $I = 0{,}267$. Under the assumption that minutiae directions are uniformly distributed and the directions for the minutiae that match in their location are independent, we obtain $I = (2 \times 22{,}5°) / 360° = 0{,}125$. For FP sensors with the resolution of 500 *dpi*, the ridge period is converted to ~ 9,1 pix./ridge $\Rightarrow w \sim 9{,}1$.

3.2.4 Experimental Results

Fingerprints from two fingerprint sensors, namely of the companies Veridicom Inc. (500 *dpi*) and Digital Biometrics Inc. (500 *dpi*), were included in a database [82]. The probabilities of fingerprint correspondence obtained for different values of M, m, n and q are indicated in Table 3.3. The theoretical curve in Fig. 3.5 provides an upper limit on the performance of an automatic fingerprint verification system. Based on the data and information as mentioned above, it has been possible to compare experimental and theoretical probabilities for the specified number of matching minutiae in impostor fingerprint matches for the Digital Biometrics Inc. sensor as shown in Fig. 3.5.

Tab. 3.3: Fingerprint correspondence probabilities [82]

M, *m*, *n*, *q*	*p*(*FP Correspondence*)
104, 26, 26, 26	$5{,}27 \cdot 10^{-40}$
104, 26, 26, 12	$3{,}87 \cdot 10^{-9}$
176, 36, 36, 36	$5{,}47 \cdot 10^{-59}$
176, 36, 36, 12	$6{,}10 \cdot 10^{-8}$
248, 46, 46, 46	$1{,}33 \cdot 10^{-77}$
248, 46, 46, 12	$5{,}86 \cdot 10^{-7}$
70, 12, 12, 12	$1{,}22 \cdot 10^{-20}$

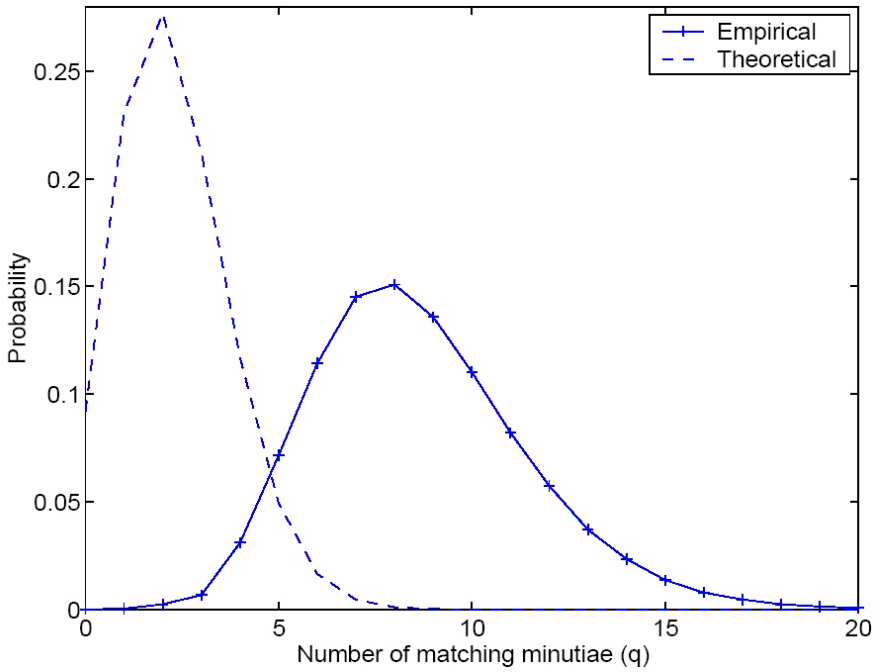


Fig. 3.5: Comparison of experimental and theoretical probabilities [82]

3.3 Strength of Information from Fingerprints

The uniqueness of fingerprints was established only on the basis of experience and observation. Three dactyloscopic axioms have been defined (as the extension of premises from page 47) [60]:

- There are no such two people in the world which would have an identical pattern structure of papillary lines.
- The pattern of papillary lines of any person remains relatively stable or unchanged for his or her whole life.
- The papillary lines regenerate with the growth of the skin. The papillary lines cannot be destroyed, only when very deep removal of the skin occurs.

The second dactyloscopic axiom has been proven by the medical science and has been examined, e.g. within the project BioFinger [3]. The third dactyloscopic axiom has also been explained by the medical science.

However, the first dactyloscopic axiom is subject of ambiguity. The first obvious method how to prove the validity of the first axiom would be to compare all available fingerprints in the world. But this is impossible. This axiom says that not only all people living in the world have different fingerprints, but that each human has the fingerprint which is at least a little bit different from some other fingerprint. We are not able to compare all fingerprints, although there are huge amounts of fingerprints saved in criminal police databases. These fingerprints are mostly rolled inked fingerprints or latent fingerprints, i.e. they contain more information than live-scan fingerprints. It is clear that the method based on comparing is not sufficient for the proof of this statement. The second method would be to find some theoretical model, which describes the uniqueness of the fingerprint, so that it would be possible to determine the validity of the first dactyloscopic axiom. In this chapter, I try to describe my ideas concerning the uniqueness of fingerprints. As a result, it should be found a sufficiently big number which would be big enough that the uniqueness of the fingerprint could be then declared. Of course, another type of uniqueness is needed for cryptographic tasks, namely the uniqueness for random sequences, or more precisely for cryptographic key generation. The main question which should be answered is: Is there enough usable information hidden in a biometric pattern and is such information suitable for cryptographic purposes? It is well established that random information can be used as the cryptographic key data as long as it is unpredictable and unguessable (see Chapter 3.1.1) [32].

As an example of a random information source, we can discuss the information input from a fingerprint scan. The fingerprint data recognition is an old traditional technique. It exploits the visible structure of papillary lines and their discontinuity. Therefore the common examination scale is related to the visible image of a print and measures typically 1 cm^2 [32]. Relevant objects of interest on the fingerprint, so called minutiae, can be sufficiently well identified or discerned by the naked eye or with the help of a magnifying glass. Nowadays, the measurement of fingerprint minutiae will be performed with appropriate sensors which record the two-dimensional structure with a fixed space resolution and can detect additional local properties like reflexivity, conductivity, etc., of the skin, in most cases with the resolution depth of 8 bits. Following this simple limitation, it is not surprising that the number of possible different pattern combinations is – although huge – still finite.

3.3.1 Resolution

In common fingerprint systems, the information on the position of ridge ending or ridge bifurcation is usually converted into the *x-y* coordinates of the minutia. In addition to this, some additional topological information about the type of minutia and corresponding ridge direction (gradient or angle) is used to create an information vector consisting of digitized values. For our rough estimation of an upper limit of

available information in a fingerprint, we start with the definition of a typical dimension of a minutia.

There is very wide offer of fingerprint sensor technologies in the market (see Chapter 2.1.1). The present technologies include capacitive, optical, thermal, ultrasound and other sensor principles [29]. Each of these technologies has its own characteristic features. Some of them can acquire very big area of the fingerprint, other ones are recommended for rolled fingerprints (e.g. scanners of the company Smiths Heimann Biometrics). When using sweep sensors, the finger needs to be swept across a small area sensor. The fingerprints are then, of course, different. When miscellaneous sensor technologies are applied at the same finger, the images are different. The color dispersion or distribution in this section is not so important, but the resolution of the sensor plays a significant role.

If we want to analyze the aspects of resolution, we need to include the resolution of papillary lines on a human finger. Not only sensor resolution is so important.

First, we define the resolution (size) of papillary lines on the finger itself. Let σ_F is the size of a minutia. This means also the actual average thickness of a papillary line. The minutiae points (ridge bifurcation and ridge ending) are built of papillary lines; hence the size of minutiae points is the same as for papillary line, σ_F. The average size of minutiae was established on the basis of measurements on real fingers and measurements on fingerprints with known image resolution. The Fig. 3.6 contains several fingerprints (from the left to the right: FVC2004[3], DB1, 106_4; FVC2004, DB3, 109_7; rolled fingerprint) and shows the determination of σ_F in three fingerprints. Each of the databases FVC2004 contains 80 fingerprints (Σ=240). The database of rolled dactyloscopic fingerprints contains 75 fingerprints. Total 315 fingerprints were examined and, in each of them, two representative papillary lines were selected and then measured. The average values can be found in the Table 3.4. The average of all values of the Table 3.4 is σ_F = 0,331625 *mm* ≅ 0,33 *mm*. So we can take the size σ_F = 0,33 *mm* as the size of typical papillary line (i.e. minutia).

Tab. 3.4: Average results of fingerprint resolution σ_F

Database	Size [pixel]	Resolution [*dpi*]	Pixel size [*mm* / pixel]	∅ thickness [pixel]	∅ thickness [*mm*]
FVC04_DB1	640×480	500	0,0508	6	0,3048
FVC04_DB2	328×364	500	0,0508	5	0,2540
FVC04_DB3	300×480	512	0,0496	7	0,3472
Rolled FDB	512×512	302	0,0841	5	0,4205

The Fig. 3.7 shows the histogram for three papillary lines (at the resolution of 512 *dpi*). The maximum local value of the top of the peak is taken for the computation of 18% (practical experience value on the basis of the previous text) of the peak

[3] http://bias.csr.unibo.it/fvc2004/

height. All pixels running from left to right at this 18% level are then counted and this number corresponds to the thickness of the papillary line. The number of pixels corresponds to the thickness of respective papillary line. This computation confirms the resulting value of thickness σ_F = 0,33 *mm*.

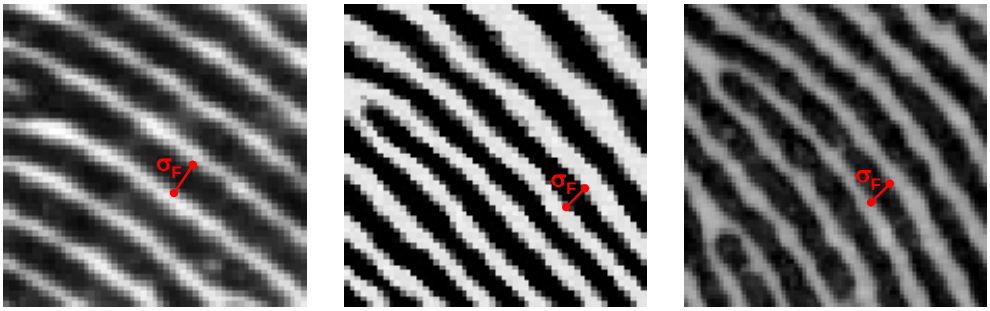


Fig. 3.6: Determination of fingerprint (minutiae) resolution σ_F

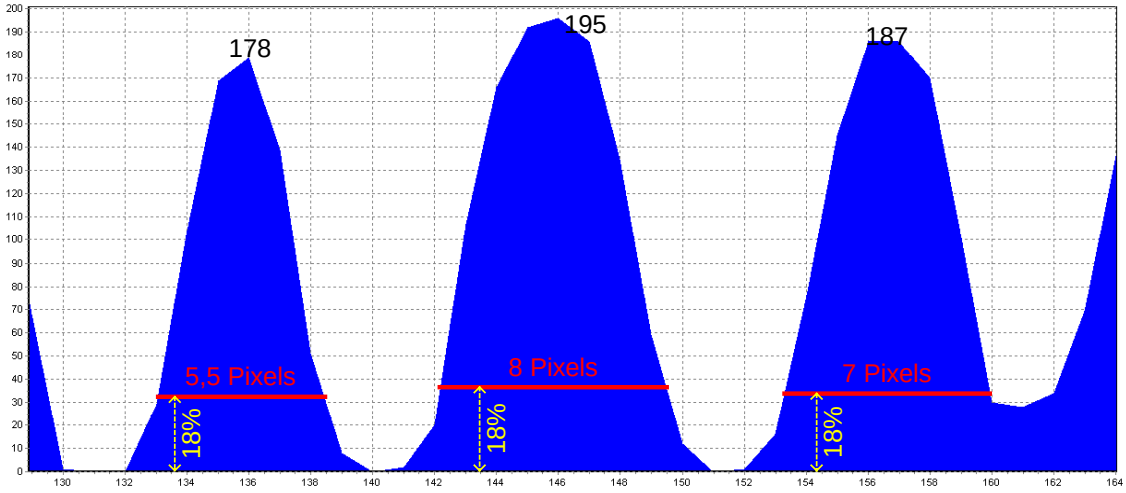


Fig. 3.7: Histogram for three papillary lines

What concerns the second part of the question of resolution, we need to consider the resolution of fingerprint scanners. The resolutions of scanners vary a little bit. The worse ones have the resolution by 250 *dpi* and the best ones by 1000 *dpi* [29]. The usual sensor resolution of up-to-date sensors ranges from 500 *dpi* to 600 *dpi*. The sensors with low resolution lose some information, not only because the resolution is insufficient, but because the image information often contains a lot of noise. The noise plays important role for cases of small resolution, as the information can be irrecoverably destroyed. Hence, the higher the resolution is, the more information can be found in the fingerprint. Due to problems with image noise, the present sensors have the resolution of 500 *dpi* or 600 *dpi* as a standard. The sensors with even higher resolution are often used for dactyloscopic purposes or other police applications.

The size of one pixel in a sensor image can be denoted as σ_S. When we use the resolution of 600 *dpi*, then we have the pixel size σ_S = 0,043 *mm*. The relation

between the minimal resolvable scales σ_F and σ_S is $\sigma_F \gg \sigma_S$. This condition of inequality should be always complied with. If the pixel size of the sensor is greater than the size of minutiae, then some pieces of information are lost. When any non-compliance with the above condition of inequality occurs, there will be a risk that the right minutiae may not be found. The typical situation of compliance with the condition of proper relation between biological resolution and sensor resolution is shown in the Fig. 3.8. In this case, the relation between σ_F and σ_S is: $\sigma_F = 7{,}7 \times \sigma_S$.

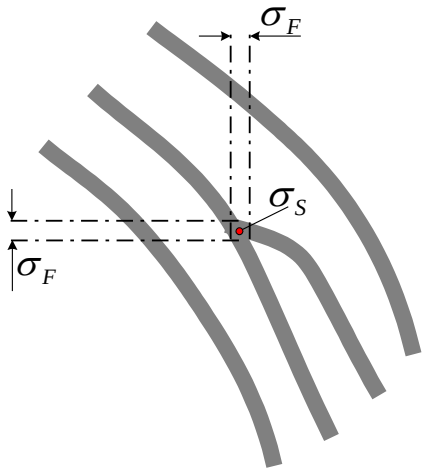


Fig. 3.8: Ridge bifurcation in corresponding biological resolution σ_F and final reduction to sensor resolution σ_S

3.3.2 Fingerprint Size

Each person has different finger size. For example, a female finger is smaller than a male one. And male fingers have, of course, different areas. But this fact does not play any substantial role in the automated fingerprint recognition. Fingerprint sensors can even scan only a limited finger area. Most sensors are of classical type, requiring that the finger is laid on the sensor. If we do not consider special sensors, like sweep sensors or sensors for rolled or latent fingerprints, then the area of the sensor corresponds with the maximal area of resulting fingerprint. The software stops fingerprint processing, if the quality is not good. The criteria for tests of quality are described in [3]. Two of them are: the color distribution in the whole image and the area of the fingerprint in the image. The software can include some specification of threshold which determines the border between sufficient and insufficient fingerprint area. It can be set that e.g. 70% of the image must contain papillary lines. Such software condition for refusal of bad fingerprint images can be deduced from the values of **FTE** or **FTM** rates (see Chapter 1.3). These rates in fact determine the correct fingerprint area.

In the Figure 3.9 (from the left: FVC2004, DB1, 101_2; FVC2004, DB2, 108_5; FVC2004, DB3, 106_6), there are three fingerprints and the surrounding environment. We consider only the fingerprints from classical fingerprint scanners. The rolled and latent fingerprints cannot be considered, because they are acquired under special conditions and are not common in the access systems. When we see

the particular images, it is clear that the real fingerprint area is smaller than the image area. Some selected fingerprints are presented in the Fig. 3.9, as they are suitable from the point of view of the rate of image size to fingerprint size. The comparison of fingerprint areas in all three FVC2004 databases enables us to say that the average fingerprint area is approximately 10 *mm* × 15 *mm* (width × height). Such area size is well accepted by all algorithms and the processing of such fingerprints is quite reliable. Some fingerprints in the FVC2004 databases are so small or so unclear that they would be refused by the acquirement software. On the other side, there are some fingerprints, whose area is so large that they can be considered almost as rolled fingerprints (e.g. with delta(s)). The area of 1,0 *cm* × 1,5 *cm* can be expressed in terms of σ_F or σ_S. This area is then $31\sigma_F \times 46\sigma_F$ or $233\sigma_S \times 349\sigma_S$. The second expression gives us the probable range of area in pixels, which the real fingerprint should have (without considering unimportant image surroundings).

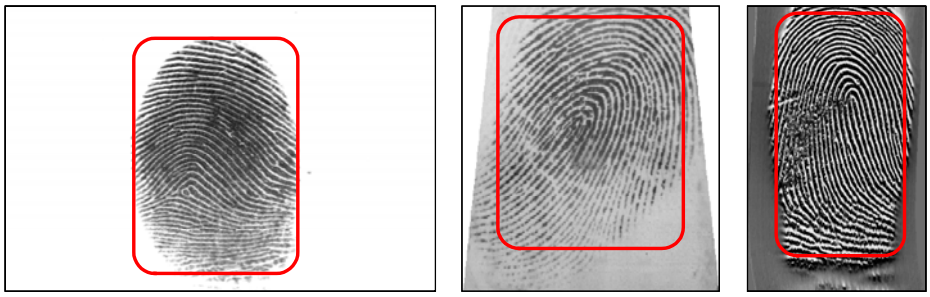


Fig. 3.9: Different fingerprint areas

3.3.3 Minutia and Antiminutia

We have defined the sizes of papillary line (or minutia) and sensor image pixel. Further the fingerprint area of some common fingerprint scanners has been defined. We suppose that the occurrence of *minutia* and *antiminutia* can be considered as mutually independent (see *Baye's Theorem* in the Chapter 3.1.1, Eq. (3.3)). Under these assumptions, we can define the term *antiminutia*.

Let the square $\sigma_F \times \sigma_F$ be called the elementary cell (see Figure 3.8). During the recognition procedure, only those minutiae are discovered which can be clearly distinguished from the "background signal" (i.e. other papillary lines, ridges or other minutiae). In other words, there should be at least a small area surrounding the minutia *m* (see Figure 3.10). We call the eight elementary cells *A* around the minutia *m* as the *antiminutiae* and *A* gets for simplicity the same scale σ_F. In the estimation, the biological dimension of a minutia and an antiminutia is defined as the elementary cell, i.e. 0,33 *mm* × 0,33 *mm* (σ_F=0,33 *mm*). The choice of such characteristic resolution σ_F guarantees sufficient invariance against small changes in the fingerprint structure due to aging or other systematic effects. In the model, two closely adjacent minutiae need to be separated by at least one antiminutia *A* and their centers of gravity need to have the distance at least $\geq 2\times\sigma_F$ (see Fig. 3.10).

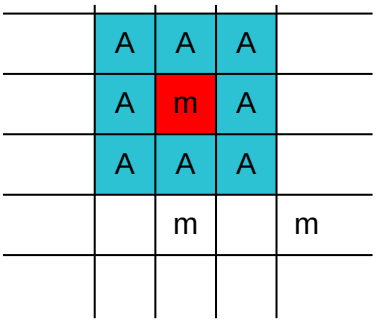


Fig. 3.10: Definition of minutia (*m*) and antiminutia (*A*)

As an example – we consider the square area of 1 mm^2. This can be formulated as $3\sigma_F \times 3\sigma_F$ and only 4 minutiae could be placed here without two minutiae touching each other.

If we consider a rectangular fingerprint with the typical dimension of 10 *mm* × 15 *mm*, which corresponds to the matrix of $31\sigma_F \times 46\sigma_F$, the maximal number of minutiae is 368. Generally, we can use the following equation for the computation of the number of minutiae [27, 31]:

$$P_M = \left\lfloor \frac{a+1}{2} \right\rfloor \cdot \left\lfloor \frac{b+1}{2} \right\rfloor \tag{3.29}$$

where P_M is the number of minutiae contained in the rectangle of $a \cdot \sigma_F \times b \cdot \sigma_F$ – any fraction of such number should be rounded down. The dimensions *a* and *b* are the numbers of σ_F units on both sides of a typical fingerprint area (10 *mm* × 15 *mm*). For the above mentioned data, the result is 368 minutiae which can be present in our typical fingerprint area. It is, of course, very big number of minutiae. In practice, this number is much lower. As a standard, there are about 30 to 60 minutiae in most live-scan fingerprints; this number is higher for rolled fingerprints, but of course, the area is bigger too. Although the resolution σ_S is better than σ_F, we cannot exploit this better resolution. When considering minutiae matching in the fingerprint, we can use the information from papillary lines pattern. No other information in the fingerprint is considered, as e.g. the pore structures [12] or other features.

3.3.4 Strength of Information Contained in Fingerprints

Considering the method of minutiae recognition, the following pieces of information are usually stored with each extracted minutia μ_i [22, 36]:

- Relative position $\vec{r}_i$ (*x* and *y* coordinates),
- Type t_i (ridge ending or ridge bifurcation),
- Gradient $\vec{g}_i$ (angle or direction of respective papillary line).

These data are saved for each extracted minutia. There are some other schemes, which use e.g. the curvature of papillary line in the minutia point or the image of

surroundings of the minutia point. But let us consider only that information which is set out by the dactyloscopic axioms and proven by experts from practice, i.e. the information on proper minutiae (their position, type and gradient).

When considering these minutiae data from the point of view of informatics, we can use bit streams to encode appropriate information. More specifically, we need to encode the position, type and gradient of minutiae. This information on minutiae is shown in the Fig. 3.11; the colors indicate the type of minutiae. Relative positions are expressed by vector differences, e.g. $\vec{r}_2 - \vec{r}_1$ and $\vec{r}_3 - \vec{r}_1$.

First of all, one minutia of the complete set of extracted minutiae needs to be selected as the *reference minutia*. The position of such *reference minutia* is then taken as the origin of the coordinate system and hence the coordinates and gradient of the reference minutia are equal to zero. This reference minutia is shown in the Fig. 3.11 as the point of origin in red color. Another assumption, valid for all factor computations, is that we have available the quantity of *M* minutiae. These are all real minutiae which have been found in the fingerprint. The definition of Entropy (see Chapter 3.1.2 and 3.1.3) is considered as a basic reference for the computation of factors.

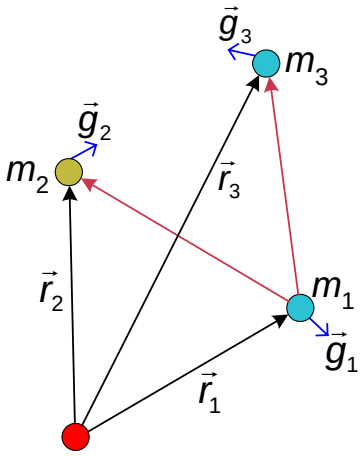


Fig. 3.11: Information which characterizes minutiae

To encode the positions, all *x* and *y* coordinates need to be stored. If we consider the fingerprint rectangle of 10 *mm* × 15 *mm* with the resolution of 600 *dpi*, then we will get the fingerprint image with (rounded down) 230 × 350 image points (pixels). Since we use the reference minutia as the origin of coordinates, then we can store other minutiae points as vectors $\vec{r}_i$. These vectors correspond to the *x* and *y* coordinates of the minutiae. The pixel matrix, which should contain all the vectors, has the size 230 × 350 pixels; this means that we have total 80.500 places in this matrix. To encode 80.500 places or positions, we need about 17 bits (in fact, this is enough to encode 131.072 places). The number 80.500 corresponds to the number of all possible, non-redundant, different vectors in a raster of 230 × 350 pixels. The factor for positions is 2^{17}. For *M* minutiae [26, 32]:

$$\left(2^{17}\right)^{M-1} \tag{3.30}$$

In the Equation (3.30) only *M*-1 minutiae are used because the reference minutia has already been defined as having the position in the origin of the coordinate system, i.e. the reference position.

The encoding of minutiae types is simple. We have only two types – ridge bifurcation and ridge ending. These two minutiae types can be encoded only with a single bit, which results in the factor [26, 32]:

$$2^M \tag{3.31}$$

The factor for types uses *M* minutiae, as the reference minutia has also a type assigned.

The last factor to be created is based on the gradients. Here again only *M*-1 minutiae can be used, as the reference minutia has no direction (it is neutral). In our estimation of factors, we have used the matrix of $5\sigma_F \times 5\sigma_F$ for the computation of the gradient [22] (see Fig. 4.4, all directions are used in this case). In this matrix, the gradient can be determined only in 16 possible directions. The angular resolution is therefore 22,5°. The standard DIN V 66400 - Fingerprint Template Format for Matching on Card [104] applies the resolution of direction 1,40625°. The question is, whether we can reliably determine the gradient of any minutia in each fingerprint (from the same finger) so precisely. In our example, very rough gradient scale is used but the factor does not change so much. For example, if we have 16 directions (per 22,5°) or 72 directions (per 5°), the factor of change is 2^3. This is also the reason why we do not use a special variable scale for different directions. The resulting factor for the resolution equal to 22,5° can be computed as follows [26, 32]:

$$16^{M-1} = \left(2^4\right)^{M-1} \tag{3.32}$$

In summary, we need approximately 17 bits for encoding of the above mentioned number of relative vectors, one bit for encoding of the minutia type and four bits for encoding of the gradient. Indeed, there is a small reduction of phase space for the number of vectors, because each additional minutia reduces the possible surface area by an area between $4\sigma_F^2$ and $9\sigma_F^2$. We neglect this fact in our estimation.

For the consequent factors, we should consider the minimal and maximal entropy factors. What concerns the minimal entropy factor, we should consider 12 minutiae. The number of 12 minutiae is recommended by FBI (Federal Bureau of Investigation) and BKA (Bundeskriminalamt) as the minimum quantity of minutiae which must to be found in a fingerprint to be able to compare two fingerprints unambiguously. On the basis of this condition, the *minimal entropy* can be computed [27, 33, 32] as follows:

$$\left(2^{17}\right)^{12-1} \cdot \left(2^{12}\right) \cdot \left(2^4\right)^{12-1} = 2^{243} = 1{,}4135 \cdot 10^{73} \tag{3.33}$$

What concerns the maximal entropy factor, it is possible to use up to 368 minutiae (see Chapter 3.3.3). These 368 minutiae correspond to the maximal number of minutiae which can be placed (considering minutiae and antiminutiae) in the matrix of 230 × 350 pixels. Then the *maximal entropy* factor is [27, 33, 32]:

$$\left(2^{17}\right)^{368-1} \cdot \left(2^{368}\right) \cdot \left(2^4\right)^{368-1} = 2^{8075} = 6{,}5647 \cdot 10^{2430} \tag{3.34}$$

What concerns the approximate entropy factor, we can consider approximately 50 minutiae, which can be often found in a (rolled) fingerprint. This latter quantity of minutiae corresponds to the factor $2^{1079} = 6{,}4768 \cdot 10^{324}$.

3.3.5 Vector Quantization

The next step is positions quantization in the fingerprint; suppose that the topological resolution σ_S is used for this procedure. However, the resolution on the level of current sensor technology is somewhat too precise or fine. The topological positioning with this level of resolution can lead to minutiae locating failures – e.g. when the minutiae extraction algorithm finds a minutia on another position differing only about one pixel in the resolution interval σ_S. To avoid this art of failures, the quantization of positions is needed. This can be explained on the example in Fig. 3.12. The red points represent the real positions of minutiae with the resolution σ_S. The grid with cells having dimensions of $\sigma_Q \times \sigma_Q$ corresponds to the quantized positions of minutiae. Such quantization intervals can be equal to or greater than the resolution σ_F.

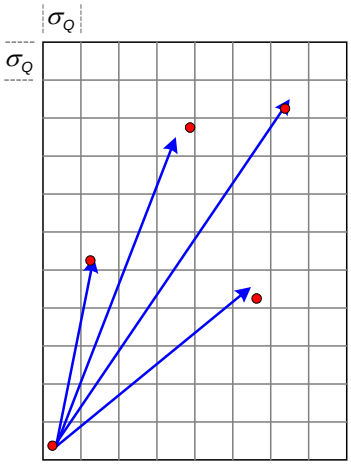


Fig. 3.12: Topological quantization of minutiae positions

In general, the topological quantization can be expressed as [27, 32]:

$$\sigma_Q = n \cdot \sigma_F, \quad n = 1,2,\ldots \tag{3.35}$$

The quantization of resolution reduces the sensitivity of minutiae extraction procedure to systematic and statistical effects [27, 32]. The Table 3.5 shows the influence of quantization on the number of grid cells (dimension of quantized matrix). The first column contains values of the coefficient *n* and the second column the total number of places/locations in the quantized matrix grid and corresponding number of bits to encode the position of minutiae in such grid. The third column contains the maximal values of the number of placeable minutiae in the corresponding quantized matrix computed with the use of the Eq. (3.29). The fourth column shows the values of the maximal entropy factor for the corresponding maximal number of placeable minutiae computed with the use of the Equations (3.30), (3.31), (3.32), (3.34); these values correspond to the number of bits for the

position encoding. The last column contains the values of entropy for the corresponding quantization resolution; these values result from the computation considering the minimal acceptable number of minutiae (which is equal to 12). The Fig. 3.13 presents the graph for the progression of the number of grid cells and the Fig. 3.14 shows the graph for the progression of the number of encoding bits and exponents using $P_{M,n}$.

It is clear that the numbers of vectors and combinations decrease with the increasing level of quantization. But this tendency is necessary for the successful generation of appropriate cryptographic key, as the finger positioning requires certain level of tolerance, in order to find next time the same quantized position.

Tab. 3.5: Combinations and entropy factors for various coefficients of quantization

Resolution $\sigma_Q = n \cdot \sigma_F$	No. of places $\Rightarrow$ No. of bits to encode appropriate positions	No. of placeable minutiae $P_{M,n}$	Maximal entropy factor for $P_{M,n}$	Entropy for $P_M = 12$
$n = 1$ (σ_Q=0,33 *mm*)	$30 \times 45 = 1350 \Rightarrow 2^{11}$	345	$\sim 2^{5505}$	$\sim 2^{177}$
$n = 2$ (σ_Q=0,66 *mm*)	$15 \times 22 = 330 \Rightarrow 2^{9}$	88	$\sim 2^{1219}$	$\sim 2^{155}$
$n = 3$ (σ_Q=1,00 *mm*)	$10 \times 15 = 150 \Rightarrow 2^{8}$	40	$\sim 2^{508}$	$\sim 2^{144}$
$n = 5$ (σ_Q=1,66 *mm*)	$6 \times 9 = 54 \Rightarrow 2^{6}$	15	$\sim 2^{155}$	$\sim 2^{122}$

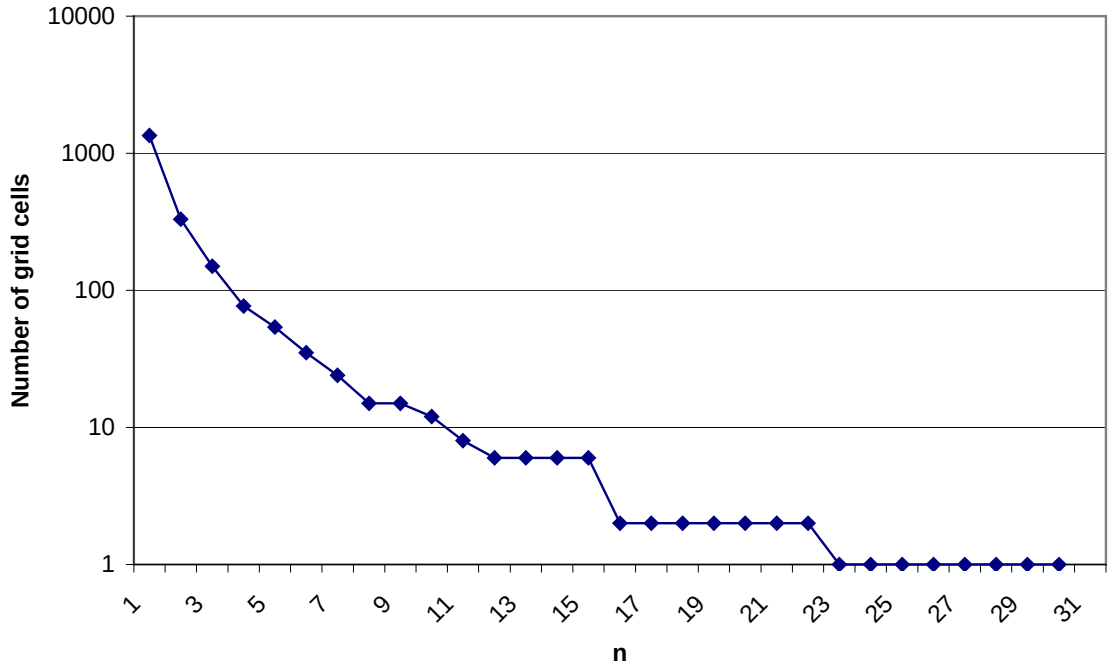


Fig. 3.13: Graph of the progression of the number of grid cells

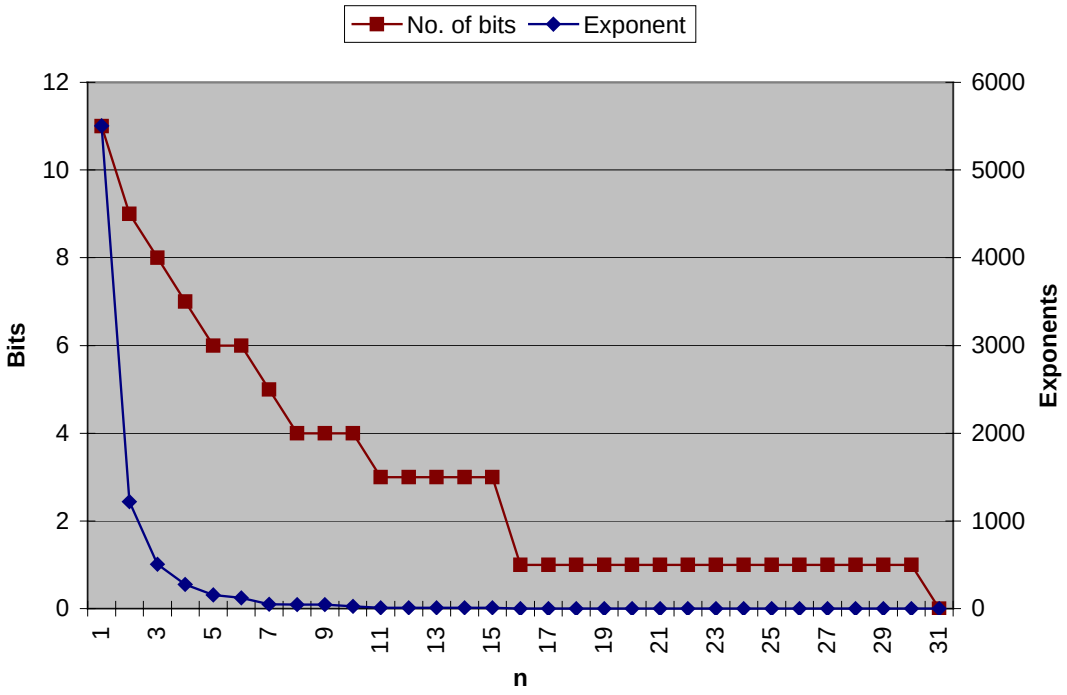


Fig. 3.14: Graph of the progression of the number of encoding bits and exponents

3.3.6 Key Length

Key length is an important issue in cryptography. Good cryptographic systems are based on algorithms that are known. It is considered as acceptable to use the algorithm publicly known, as the decrypting of encrypted data is impossible without knowing the key [115]. In a brute force attack, an attacker simply generates all possible keys and applies each one to the ciphertext until a key finally works. The more bits reserved (see Chapter 3.1.4) to comprise a key (*Baye's Theorem*), the more values the key can attain, and therefore the more possible key combinations that any attacker needs to test. The key should therefore be large enough to make a brute force attack unfeasible.

The *Shannon's* theory on information (Chapter 3.1.1 and 3.1.2) has shown that to achieve perfect secrecy, it is necessary that the key length is at least as large as the message to be transmitted. With regard to this condition, modern cryptography has resigned from the requirement of perfect secrecy for encryption and focused instead on computational security. This new condition of computational security means that the computational requirements for breaking a cipher should exceed practical feasibility for any potential attacker.

We should distinguish between the *symmetric* and *asymmetric cryptography*. In both cases, the algorithms are commonly known. But the keys are different. Only one key is sufficient in the case of symmetrical cryptography both for encryption and decryption. The protection of this key is very important. The properties of symmetric keys can be found in [115, 113, 77, 101]. On the other side, the asymmetric cryptography uses two keys, namely a private key and a public key. These

two keys represent a key pair. The properties of asymmetric keys can be found in [115, 113, 77, 101]. Required theoretical times for brute force attack (see Chapter 3.1.5) are shown in the Table 3.6; these times depend on different key lengths (valid for symmetric cryptography) [115].

Tab. 3.6: Times for brute force attack on symmetrical cryptography [115]

Key size (bits)	Number of alternative keys	Time required at 1 decryption / *μs*	Time required at 10^6 decryptions / *μs*
32	$2^{32} = 4{,}3 \cdot 10^9$	2^{31} *μs* = 35,8 *min*.	2,15 *ms*
56	$2^{56} = 7{,}2 \cdot 10^{16}$	2^{55} *μs* = 1142 *years*	10 *hours*
128	$2^{128} = 3{,}4 \cdot 10^{38}$	2^{127} *μs* = $5{,}4 \cdot 10^{24}$ *years*	$5{,}4 \cdot 10^{18}$ *years*
168	$2^{168} = 3{,}7 \cdot 10^{50}$	2^{167} *μs* = $5{,}9 \cdot 10^{36}$ *years*	$5{,}9 \cdot 10^{30}$ *years*

The Table 3.7 shows the historical development of key lengths for cases of symmetrical, asymmetrical and elliptic curves cryptography. It results from this table that the minimal quantity of 12 minutiae with the minimal entropy factor 2^{243}, i.e. the key length 243 bits (see Eq. (3.33)), is sufficient for symmetric keys as well as for elliptic curves cryptography but not for asymmetric keys (even when neglecting the key generation itself). The maximal entropy factor (Eq. (3.34)) is, indeed, suitable for all key lengths and types. If we consider the entropy factor for 50 minutiae, which is 2^{1079}, we arrive at the conclusion that the bit stream with the length of 1079 bits is not suitable for asymmetric cryptography. Further, if we consider the quantization step, then the average key length (for 12 minutiae) is around 144 bits (see Tab. 3.5) and such length is suitable for both symmetric cryptography and elliptic curves cryptography.

Tab. 3.7: Evolution of the key lengths [115]

Year	Symmetric Key Length (bits)	Asymmetric Key Length (bits)	Elliptic Curves Key Length (bits)
1982	56	417	105
1990	63	622	117
2000	70	952	132
2002	72	1028	135
2004	73	1108	138
2005	74	1149	139

3.3.7 Summary of Fingerprint Information Strength

In the beginning of this chapter, the *Shannon's* theory and basics of the entropy were described. In the following subsection, the description of fingerprint unique-

ness, published in [82], was introduced. The Chapter 3.2 shows that (when comparing two fingerprints) the probability of correspondence between these two fingerprints lies in the range of $<5{,}47\cdot10^{-59}; 5{,}86\cdot10^{-7}>$. The more important question is how much information is hidden in the fingerprint and whether this amount of information is sufficient for generation of cryptographic keys. The answer to this question can be found in the Chapter 3.3 where my own computations and results are described. The resulting information entropy lies in the range of $<1{,}41\cdot10^{73}; 6{,}57\cdot10^{2430}>$ (when neglecting the quantization). The practical experiences have shown that the entropy factor is closer to the lower range limit rather than to the higher. The result is that a closed 100-bit data stream is sufficient for the generation of cryptographic keys for the symmetric cryptography. Elliptic curves cryptographic keys are also conceivable with this data stream but the generation of the cryptographic key pair is difficult. Furthermore, such bit stream size is not sufficient for general asymmetric cryptography.

The following equation can be considered as a general result for the entropy factor. First of all the quantity of minutiae P_M is computed using Eq. (3.29) and then we receive the following expression for the matrix $a\cdot\sigma_F \times b\cdot\sigma_F$:

$$E = \left(2^{N_B}\right)^{P_M-1} \cdot 2^{P_M} \cdot \left(2^{N_G}\right)^{P_M-1} \tag{3.36}$$

where E is the entropy factor, N_B is the number of bits needed to encode the positions of minutiae in the matrix $a\cdot\sigma_F \times b\cdot\sigma_F$, and N_G is the number of bits to encode the gradients (directions) of minutiae. The number N_G is defined as equal to 4, because we have 16 directions and $16=2^4$. The number N_B needs to be computed each time for the corresponding matrix $a\cdot\sigma_F \times b\cdot\sigma_F$. And the last value P_M is limited to the matrix $a\cdot\sigma_F \times b\cdot\sigma_F$, as well as the value N_B.

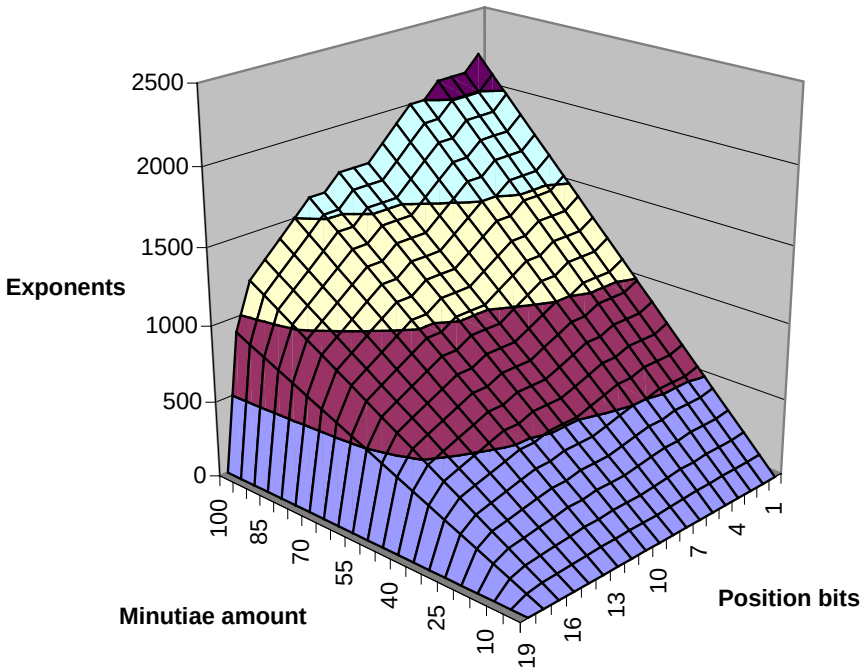


Fig. 3.15: Relation among numbers of minutiae, position bits and exponents

Of course, such very high resolution (600 *dpi*) reduces dramatically the potential for correct key generation during the generation phase. In order to achieve a higher process reliability and performance, we introduced a new additional step into the standard recognition procedure that makes the algorithm more robust and resistant against systematic effects. The price for this is the reduction of the maximal available information data size to a limit having the value between approximately 2^{177} and 2^{122} (for 12 minutiae and the quantization factors n=1 and n=5, respectively). Although this number is big, it is still negligible when compared with the amount of data necessary for the generation of an asymmetric key. On the other hand, this number is big enough for the use as a symmetric key and has much higher information entropy then commonly used passwords or PINs.

The relation among numbers of minutiae (maximum 100 and minimum 0), position bits (number of bits to encode the positions of minutiae in the matrices, beginning with 230×350 pixels with the decrease of 10% to the next one) and exponents (the amount of information or the entropy factor $2^{Exponent}$) is shown in the Fig. 3.15.

4. Key Generation

It has been quite long time since the first biometric systems were introduced but they have not become widely used until now. This holds true especially in case of the security systems using biometrics. Although the popularity of the biometric systems is not at a high level yet, they are supposed to become very important in near future.

Like many other technologies, the biometric systems have some advantages and disadvantages. It is usual and it is necessary to ask some important questions before a biometric system is applied. The questions can be as follows:

- Which object is to be protected?
- How valuable is the protected object?
- Where should be the biometric system installed?
- How strong should be the protection?
- Will the biometric system pay off?

These and many other questions should be answered before a biometric system is installed. It is useful to look at the advantages and disadvantages of the biometric systems to make a correct decision.

Advantages [31]:

- *Reliability and uniqueness*: The biometric systems are supposed to be very reliable. Reason for this is the uniqueness of some human features used in the process of authentication in the biometric system.
- *Security*: The biometric systems are very secure and the protected devices and services can hardly be misused due to the biometric protection. The biometric features can be acquired only from the authorized individual. In any case, the liveness test of the user should be done before starting the standard operation of the system. By using such liveness test, it is possible to avoid the acceptance of impostors.
- *Scalability*: The individual biometric systems may be connected together to build a multibiometric authentication system [26, 31, 34]. This may include standard login, fingerprint login, voice login, etc. These stages of login process increase the security of the whole system. Should one of them fail or be cheated, the others are able to cope with possible breakthrough. The possibility of a breakthrough decreases with increasing number and strength of the login stages.

Disadvantages [31]:

- *Exactingness*: Some of biometric features are very demanding in the process of acquirement. These features may be very precise and unique, but at the same time difficult to acquire. Among them could be included e.g. the DNA.
- *Difficult Implementation*: It is quite difficult to implement very reliable biometric system (see e.g. Fingerprint Verification Competition FVC2004 or BioFinger

project). Nowadays, many teams of developers and researchers are working on the secure and reliable implementation of biometric security systems.

- *Cooperation Unwillingness*: Some humans are not happy with providing their biometric features. A lot of them dislike scanning of their retinas, fingerprints or faces.
- *Inconstancy*: The human features are not perfectly constant during the whole life [31, 3].

The above discussed advantages and disadvantages are only typical examples of those which are important. In any case, they enable us to understand the concept of biometric systems.

Of course, another problem is the biometric data security. The questions relating to data security can be e.g. as follows [33]:

- Whom I give my biometric information to? (for driving license, passport, registration for e-voting or bank transactions, for access to a company area or runway)
- Could my biometric data be released freely from the database to all ranges of usage?
- How can I protect the templates in my smart card or in my passport?

The above mentioned aspects (pros and cons + data security) have to be considered when designing a biometric system.

The general concept of the biometric system can be extended. The biometric information can serve not only for access or authentication, but also for data protection. Some devices using biometric information for protecting private personal data are described in the Chapter 2.3 (Actual Solutions). We can ask if it is possible to generate some art of key from the biometric information that can be used with some cryptographic algorithm to encipher and decipher data. As shown in the Chapter 3 (Strength of Fingerprint Information), there is enough information entropy in the fingerprint to generate the key which is suitable for symmetrical cryptography (asymmetrical cryptography, including elliptic curves cryptography, is discussed in the same chapter). On the basis of these statements, we have introduced a cryptobiometric system using the combination of key generation from the fingerprint and voice [81], and a cryptographic algorithm. Let us call this system the *Biometric Security System* (for further discussion, it is considered only that part of the biometric security system which is based on the fingerprint technology).

4.1 Biometric Security System

The Biometric Security System consists of a general biometric system (based on fingerprints in our case) and a general cryptographic system. A new special step, an art of pipe through both systems, has been developed and tested. This connecting step corresponds to the key generation from the fingerprint. The biometric system acquires and processes biometric data (fingerprint images). Then a key is generated in an intermediate step and such key is delivered to the cryptographic module at the end. These three main steps are schematically shown in Fig. 4.1.

The following subchapters describe concrete processes within each step from the Fig. 4.1. Any biometric attributes can be used as the input biometric information, as it has been said in the general introduction of the Biometric Security System. The only requirement is the entropy power in the selected biometric attribute. If there is not enough entropy information, it is impossible to generate strong cryptographic keys, even if the process of key generation is realizable.

Fig. 4.1: Three main phases of the Biometric Security System

Besides, the individual phases shown in the Fig. 4.1 can be described in a more detailed way. In the first phase, called Acquirement, not only fingerprint scanning, but also all image processing algorithms are applied with the aim to extract the minutiae from the fingerprint. The output of the first phase is the set of minutiae points with their characteristic information data (position, gradient and type). The second phase called Key Generation is dedicated to biometric key generation. The minutiae from the previous phase are taken as an input and some mathematical operations are done with them. These mathematical operations generate sub-vectors from the set of minutiae points and these sub-vectors can be considered as keys. Although it is not absolutely necessary to generate more sub-vectors, as a single vector representing the whole minutiae set would be theoretically sufficient, it is not recommendable to limit ourselves to such single vector (see the Chapter 4.2.1). As the output of the second phase, the set of sub-vectors is processed by a cryptographic module. Well-tried cryptographic modules exist and therefore is not necessary to develop an own cryptomodule. That is why some common cryptographic algorithm can be used, such as DES or 3DES (only symmetric cryptography is considered further).

In order to make it more complicated, the whole Biometric Security System must be divided into two separate concepts. The first one is the Certificate Creation concept, and the second one is the Certificate Usage concept. Both concepts have some common steps, but there are some differences in these two main parts of the Biometric Security System. Both concepts consist of the same phases as discussed above (i.e. Acquirement, Key Generation and Cryptomodule). But the difference is visible in each phase, and indeed, the execution and handling modes

are not identical. Let us describe both concepts of the Biometric Security System in a more detailed way.

4.2 Certificate Creation Concept

In this concept, the certificate is generated, including appropriate biometric information. The certificate could be based on the X.509 standard [94, 105, 1], but an own format of the certificate structure can also be used. The main idea is that the certificate should be generated only by that administrator who possesses the key pair of the certification authority. This guarantees that the whole process of key generation and storage of relevant information to the certificate will be successful and trustful. If the whole process is completed without problem and all steps of certificate generation are under control, then the content of the certificate can be signed first. The signed certificate guarantees the correctness of all items (name, organization, department, personal number, biometric data, etc.) and the fact that these items were saved under control by a trustworthy person. Such certificates can be used also in other areas and not only for private data protection, but this will be discussed later. It is important that this Certificate Creation is done only once. Since this creation is done by an administrator, the certificate does not need to be reloaded and naturally, there is no reason to generate the certificate several times, only if some part of the certificate is to be modified or when the certificate is no more valid. In such cases, a new certificate has to be created. The same biometric attribute can be used in other applications, as the biometric data are not saved (not even as a template) in the certificate and therefore cannot be compromised.

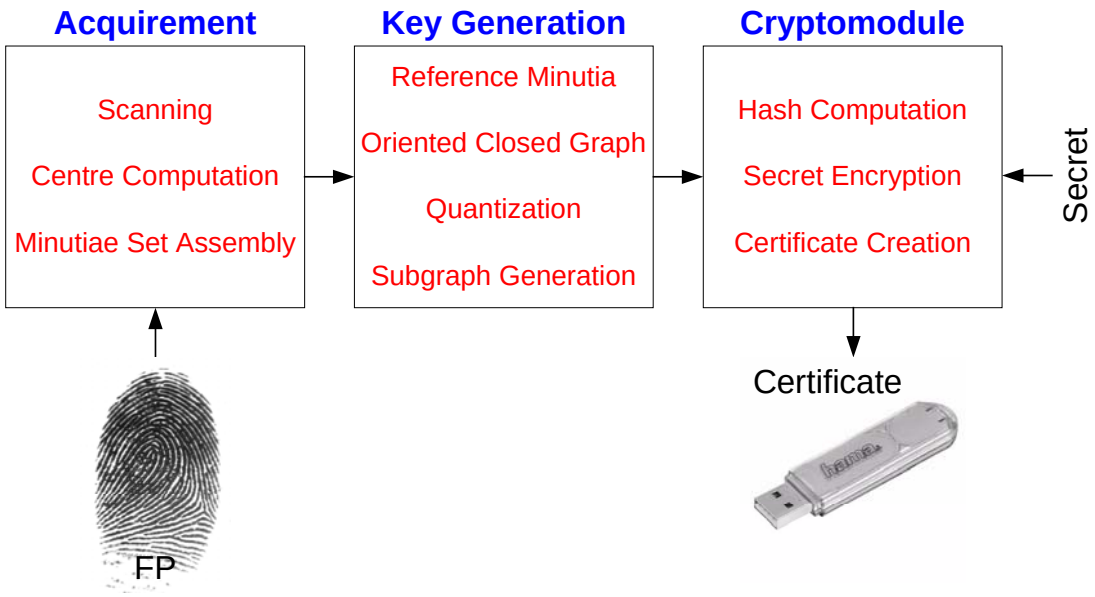


Fig. 4.2: Concept of Certificate Creation for the Biometric Security System

4.2.1 Acquirement Phase

The acquirement phase is simple. Nearly all algorithms for acquirement have been described in the Chapter 2. For the fingerprint acquirement, some sensor has to be used. The sensor technology [29] plays no critical role. The output of fingerprint sensor is represented by an image with the resolution of approximately 500 *dpi* and generally with 256 gray levels. Further steps for image enhancement and processing (see Chapter 2.2.1 and 2.2.3) could differ in accordance with miscellaneous minutiae extraction models. We consider the following steps [22, 47, 48]: image enhancement, thresholding (comparing with a threshold), ridge thinning and minutiae extraction. Some other models do not need to use e.g. the image enhancement because the image quality is acceptable, or do not need to use thresholding because some sensors provide black and white images at the output, or there exist some algorithms which do not need thinned papillary lines as they use other minutiae detection techniques. Only the result is important here – a set of minutiae points.

The whole acquirement phase should be repeated approximately five times. The reason for this is the quality assurance of respective set of minutiae points. The quality of this set means that the minutiae included in this set will be found next time again with a very high probability.

In each step (there are five of them in our case) the minutiae of the appropriate fingerprint are stored. Let us call the minutiae μ_i^j, where i is the descriptor of the fingerprint image (i = 1, ..., 5), and j is the number of minutiae n_i found in the appropriate fingerprint (j = 1, ..., n_i). Each minutia (only ridge ending and ridge bifurcation are considered) has three items (position is regarded as a single item) [22, 24, 35]:

$$\mu_i^j = \left(x_i^j, y_i^j, \phi_i^j, t_i^j\right) \tag{4.1}$$

where x_i^j is the x-coordinate position, y_i^j is the y-coordinate position, ϕ_i^j is the gradient and t_i^j is the type of a particular minutia.

Computation of fingerprint centre

Very important piece of information is the position of the centre or midpoint of the fingerprint. The fingerprint centre position should be determined before assembling appropriate set of minutiae. Important thing is that the centre of the fingerprint is not affected by any translation or rotation of the fingerprint image – each of the methods described below is therefore independent of rotation or translation of the fingerprint image. In the Certificate Creation concept, it is possible to interchange both steps (computation of centre position and assembly of minutiae set) but there is no assembling of minutiae set in the concept of the Certificate Usage and this also means that the centre position must be computed first (it is necessary to respect the same order of processing steps). All three methods for fingerprint centre computation can be described as follows:

- *Method based on the minutiae gravity centre*. This method is based on the position of all the minutiae μ_i^j, more precisely on their x and y coordinates. The

computational procedure is based on the definition of the Euclidean distance as the straight line distance between two points [9]. Thanks to only two dimensional nature of fingerprint image, the Euclidean distance can be computed from relatively simple formula [9]:

$$d = \sqrt{\left|x_1 - x_2\right|^2 + \left|y_1 - y_2\right|^2} \tag{4.2}$$

When extending this formula to our minutiae set μ_i^j, we receive the following expression for the Euclidean distance:

$$d_i^j = \frac{1}{n_i - 1} \cdot \sum_{k=\{1..n_i\}\setminus\{j\}} \sqrt{\left|x_i^j - x_i^k\right|^2 + \left|y_i^j - y_i^k\right|^2} \tag{4.3}$$

This expression is then used for the computation of the approximate Euclidean distance of each minutia in each fingerprint (as mentioned before, we should have up to five fingerprints for each Certificate Creation). In the next step, we need to find out the minimum value of these distances in each fingerprint. This minimum can be computed as:

$$\delta_i = \min\left(d_i^1, \ldots, d_i^{n_i}\right) \tag{4.4}$$

The minutia with the minimal distance δ_i has the same coordinates as the centre of the fingerprint with coordinates $[C_X;C_Y]$.

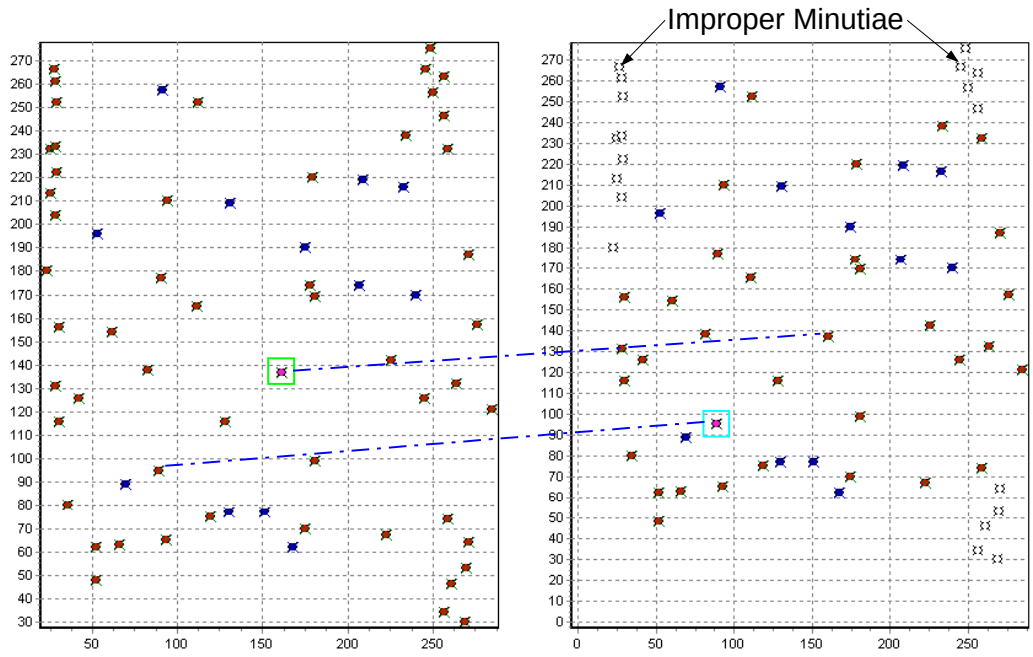


Fig. 4.3: Proper vs. improper minutiae

It should be mentioned that this centre can vary. If the sensor provides images which contain a lot of information noise in the image data, then improper minutiae (often on the edges of the fingerprint) can be selected. The minutiae extraction algorithm can delete these improper minutiae, but some additional

information on goodness of the minutiae is needed. That is the reason, why it is recommended to use sensors and algorithms with a very high quality, when using this type of the fingerprint centre determination. But, on the other hand, if the sensor has some damaged cells, then the minutiae will be found always on the same place and the centre may not vary too much. In the Figure 4.3, two fingerprints from the same finger are shown; one with improper minutiae on the edge (yellow points) and the other one with all proper minutiae (improper ones have been deleted).

- *Method based on ridge count* (RC). Papillary lines are ridges in the fingerprint which can be represented (in the cross-section) as some art of the sine wave (see Fig. 3.7). For better representation, we can consider the papillary lines (from aspects of fingerprint image pattern) as homocentric circles with the origin in the real centre of the fingerprint. We can compute the number of throughpasses in the horizontal and vertical directions. It is clear that the number of circle throughpasses in the centre of all circles is greater than in outlying regions. These throughpasses define the ridge count for each column or row. For better idea, see Fig. 4.4.

  The following expression can be used for the computation of vertical ridge counts (in rows):

  $$RC_{V,All} = \{RC_i \,|\, i = 0 \ldots Height\}, \tag{4.5}$$

  where *Height* is the number of pixels in the vertical direction (image height) and RC_i is the ridge count for the corresponding row in the image. The following condition needs to be defined for RC_i: Increase the value of RC_i only if at least the following p (in our case p = 5) pixels are higher than certain predefined threshold (it is recommended to apply the global image pixel threshold as the middle value of the gray scales used in image); the next increase of such value can happen only if the throughpass of minimally p pixels with the values lower than the defined threshold occurs. For the selection of the vertical centre, the value RC_V needs to be computed:

  $$RC_V = avg(\max(RC_{V,All})), \tag{4.6}$$

  which represents the value C_Y and is computed as an average of the region with maximal value (more rows in the middle can have the same ridge count value) of the ridge count from the whole set $RC_{V,All}$.

  Similar equations can be used for the horizontal ridge count (in columns):

  $$RC_{H,All} = \{RC_i \,|\, i = 0 \ldots Width\} \text{ and } RC_H = avg(\max(RC_{H,All})) \tag{4.7}$$

  The meaning of respective terms in the Eq. (4.7) corresponds to the meaning of the same terms in the previous vertical ridge count description. The value of RC_H represents the centre position C_X. At the end of this computation procedure, we obtain the coordinates of the fingerprint centre $[C_X;C_Y]$.

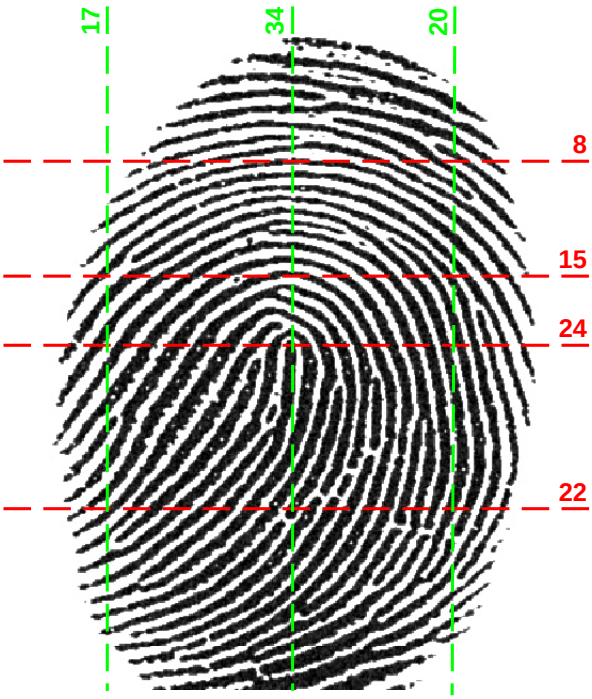


Fig. 4.4: Different ridge counts in the fingerprint

- *Method based on the Orientation Field* (OF). Fingerprint images can be considered as an oriented texture pattern [87]. When applying the taxonomy described in [87], fingerprints can be classified as a weakly-ordered texture. The orientation field is used to compute the optimal dominant ridge direction in each $w \times w$ window or block. Several methods for the determination of the orientation field have been proposed, e.g. *Rao's* algorithm [55, 58, 87] or Ridge-Valley Orientation Detector [42, 63]. We apply the method based on *Rao's* algorithm, described in [50, 56].

  The orientation image represents an intrinsic property of fingerprint images and defines invariant coordinates for ridges and valleys in local neighborhood. The main steps in determining the orientation image using the algorithm based on the least mean square iteration method are as follows [47]:

  1) Divide the input fingerprint image into blocks of size $w \times w$. For 500 *dpi* images, the initial recommended value of w is 16.

  2) Compute the gradients [47] $\partial_x(i, j)$ and $\partial_y(i, j)$ at each pixel, (i, j). Depending on computational requirements, the gradient operator may vary from the simple *Sobel* operator to the more complex *Marr-Hildreth* operator [47].

  3) Estimate the local orientation of each block centered at pixel (i, j) using the following equations [47, 50]:

$$\nu_x(i,j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} 2\partial_x(u,v)\partial_y(u,v) \tag{4.8}$$

$$\nu_y(i,j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} (\partial_x^2(u,v) - \partial_y^2(u,v)) \tag{4.9}$$

$$\theta(i,j) = \frac{1}{2}\cot\left(\frac{\nu_y(i,j)}{\nu_x(i,j)}\right) \tag{4.10}$$

where $\theta(i, j)$ is the least square estimate of the local ridge orientation at the block centered at pixel (i, j). Mathematically, it represents the direction that is orthogonal to the dominant direction of the Fourier spectrum of the $w \times w$ window.

4) Due to the presence of information noise, disrupted ridge and valley structures, unclear minutiae, etc., in the input image, the estimated local ridge orientation $\theta(i, j)$, may not be always correct. Since the local ridge orientation varies slowly in a local neighborhood where no singular points appear, a low-pass filter can be used to modify the improper local ridge orientation. In order to perform the low-pass filtering, the orientation image needs to be converted into a continuous vector field, which is defined as follows [50]:

$$\Phi'_x(i,j) = \sum_{u=-\frac{w_\Phi}{2}}^{\frac{w_\Phi}{2}} \sum_{v=-\frac{w_\Phi}{2}}^{\frac{w_\Phi}{2}} h(u,v)\Phi_x(i-uw, j-vw) \tag{4.11}$$

$$\Phi'_y(i,j) = \sum_{u=-\frac{w_\Phi}{2}}^{\frac{w_\Phi}{2}} \sum_{v=-\frac{w_\Phi}{2}}^{\frac{w_\Phi}{2}} h(u,v)\Phi_y(i-uw, j-vw) \tag{4.12}$$

where h is a 2-dimensional low-pass filter with a unit integral and $w_\Phi \times w_\Phi$ specifies the size of the filter. Note that a smoothing operation is performed at the block level. For further information – see [50].

5) Compute the local ridge orientation at (i, j) using

$$\mathrm{O}(i,j) = \frac{1}{2}\cot\left(\frac{\Phi'_y(i,j)}{\Phi'_x(i,j)}\right) \tag{4.13}$$

6) Compute the consistency level of the orientation field in the local neighborhood of a block (i, j) by the following formula:

$$C(i,j) = \frac{1}{n}\sqrt{\sum_{(i',j')\in D} \left|\mathrm{O}(i',j') - \mathrm{O}(i,j)\right|^2} \tag{4.14}$$

$$\left|\mathrm{O}(i',j') - \mathrm{O}(i,j)\right| = \begin{cases} d & \text{if } d < 180° \\ d - 180° & \text{otherwise} \end{cases} \tag{4.15}$$

$$d = (\mathrm{O}(i',j') - \mathrm{O}(i,j) + 360°) \mod 360° \tag{4.16}$$

where D represents a local neighborhood around the block (i, j); n is the number of blocks within D; $\mathrm{O}(i', j')$ and $\mathrm{O}(i, j)$ are local ridge orientations for blocks (i', j') and (i, j), respectively.

7) If $C(i, j)$ is above a certain threshold T_C, then the local orientations in this block are re-estimated at a lower resolution level until $C(i, j)$ is below a certain threshold.

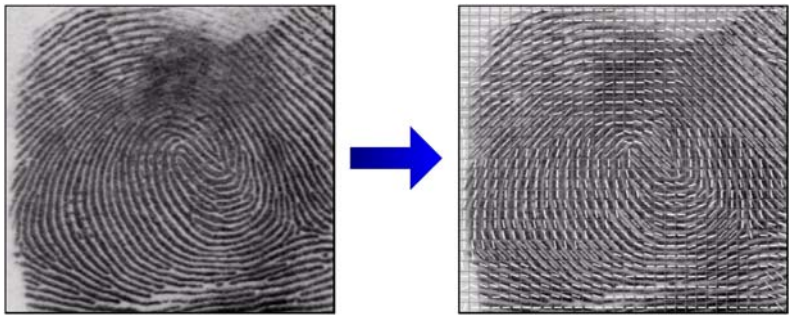


Fig. 4.5: Result of the orientation field computation

After the orientation field of an input fingerprint image is determined, an algorithm for the localization of the region of interest is applied, based on the local certainty level of the orientation field. The result is the located region of interest within the input image. The level of certainty of the orientation field in the block (i, j) is defined as follows:

$$\varepsilon(i,j) = \sqrt{\frac{1}{w \times w} \cdot \frac{(\nu_x(i,j)^2 + \nu_y(i,j)^2)}{\nu_C(i,j)}} \tag{4.17}$$

$$\nu_C(i,j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} \left(\partial_x^2(u,v) + \partial_y^2(u,v)\right) \tag{4.18}$$

If the level of certainty of the orientation field in a block is below a pre-defined threshold T_l,, then all the pixels in such block are marked as background pixels. It is assumed that only one fingerprint is present in the image. Let us call the reduced orientation field O_R. The background pixels, which may disturb following computations, can be deleted.

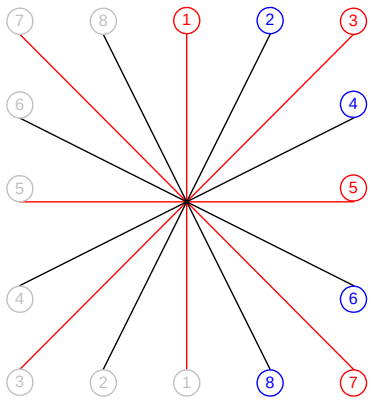


Fig. 4.6: Directions of orientation field pixels

The computation of the centre, based on O_R, consists of following steps:

1. For the estimation of the centre of the fingerprint, some reduction of the number of directions needs to be done. Normally, 8 possible directions are used in each block $w \times w$ [42, 22]. These directions are shown in the Fig. 4.6. The numbers on the left side of this figure are repeated, what means that the direction 0° is the same as 180°; 22.5° is the same as 202.5°; 45° is the same as 225°, etc. The gradients with angles above 180° and below 360° are assigned to those with angles from 0° to 180°.

   The directions in the Figure 4.6 have the following angle values: ①=90°, ②=67,5°, ③=45°, ④=22,5°, ⑤=0°, ⑥= 157,5°, ⑦=135° and ⑧=112,5°. This number of directions is necessary for the classification. But for the fingerprint centre computation, the number of directions could be reduced. In our case, only 4 directions are sufficient, namely ①, ③, ⑤ and ⑦. The directions ① and ⑤ remain without change. The directions ② and ④ are assigned to the direction ③. The directions ⑥ and ⑧ are assigned to the direction ⑦. Now, each direction has the angle resolution of 45°. Let us call the orientation field with only 4 directions O_{4R}.

Fig. 4.7: Block division of the fingerprint image

2. The fingerprint image needs to be divided into four uniform blocks. These blocks can be considered as the particular blocks of the coordinate system, with the centre in the middle of the fingerprint image. This situation is shown in the Figure 4.7, with the following blocks: ❶, ❷, ❸ and ❹.

   Let us define the origin of the image in the upper left corner, i.e. in the position [0, 0]; then the end of the image will be in the lower right corner, i.e. in the position [m, n].

   Let us define the procedure for gravity centre computation as follows (in general):

$$C_{x(h)}^{(r_1+r_2)} = \max\left(\sum_{j=c}^{d} O_{4R}^{(h)}(i,j)\right),\ i = a \ldots b \tag{4.19}$$

$$C_{y(h)}^{(r_1+r_2)} = \max\left(\sum_{j=a}^{b} \mathrm{O}_{4R}^{(h)}(i,j)\right),\ j = c \ldots d \tag{4.20}$$

where $C_{x(h)}^{(r_1+r_2)}$ is the *x* position and $C_{y(h)}^{(r_1+r_2)}$ is the *y* position of the orientation field direction *h* in the image blocks r_1 and r_2. The term $\mathrm{O}_{4R}^{(h)}(i,j)$ denotes the value of the orientation field at the point (*i*, *j*).

To determine the centre of the fingerprint, it is necessary to compute the centers of gravity for the following orientation field directions: ①, ③, ⑤ and ⑦. Two centre points are computed for each direction, each lying in the opposite block(s). These two points for each direction can be considered as the end points of an abscissa. This abscissa could be also described as a continuous line using applicable rules of the mathematical analysis as follows:

$$y_h = k_h \cdot x_h + l_h = \frac{C_{y(h)}^{(t_1+t_2)} - C_{y(h)}^{(t_3+t_4)}}{C_{x(h)}^{(t_1+t_2)} - C_{x(h)}^{(t_3+t_4)}} \cdot x_h + \left(C_{y(h)}^{(t_3+t_4)} - \frac{C_{y(h)}^{(t_1+t_2)} - C_{y(h)}^{(t_3+t_4)}}{C_{x(h)}^{(t_1+t_2)} - C_{x(h)}^{(t_3+t_4)}} \cdot C_{x(h)}^{(t_1+t_2)} \right) \tag{4.21}$$

where *h* is the orientation field direction and t_1, ...,t_4 correspond to the image blocks r_1 and r_2.

a) For *h* = ①, we can use the following settings:

For $r_1(1)$ = ❶ and $r_2(1)$ = ❸ : $a = 0$, $b = \frac{m}{2}$, $c = 0$, $d = n$.

For $r_1(2)$ = ❷ and $r_2(2)$ = ❹ : $a = \frac{m}{2}+1$, $b = m$, $c = 0$, $d = n$.

For y_1 : $t_1 = r_1(2)$ = ❷, $t_2 = r_2(2)$ = ❹, $t_3 = r_1(1)$ = ❶, $t_4 = r_2(1)$ = ❸.

b) For *h* = ⑤, we can use the following settings:

For $r_1(1)$ = ❶ and $r_2(1)$ = ❷ : $a = 0$, $b = m$, $c = 0$, $d = \frac{n}{2}$.

For $r_1(2)$ = ❸ and $r_2(2)$ = ❹ : $a = 0$, $b = m$, $c = \frac{n}{2}+1$, $d = n$.

For y_5 : $t_1 = r_1(2)$ = ❸, $t_2 = r_2(2)$ = ❹, $t_3 = r_1(1)$ = ❶, $t_4 = r_2(1)$ = ❷.

c) For *h* = ③, we can use the following settings:

For $r_1(1)$ = ❶ and no $r_2(1)$: $a = 0$, $b = \frac{m}{2}$, $c = 0$, $d = \frac{n}{2}$.

For $r_1(2)$ = ❹ and no $r_2(2)$: $a = \frac{m}{2}+1$, $b = m$, $c = \frac{n}{2}+1$, $d = n$.

For y_3 : $t_1 = r_1(2)$ = ❹, no t_2, $t_3 = r_1(1)$ = ❶, no t_4.

d) For *h* = ⑦, we can use the following settings:

For $r_1(1)$ = ❸ and no $r_2(1)$: $a = 0$, $b = \frac{m}{2}$, $c = \frac{n}{2}+1$, d = n.

For $r_1(2) =$ ❷ and no $r_2(2)$: $a = \frac{m}{2}+1$, $b = m$, $c = 0$, $d = \frac{n}{2}$.

For y_7 : $t_1 = r_1(2) =$ ❷, no t_2, $t_3 = r_1(1) =$ ❸, no t_4.

If we have appropriate analytical descriptions of the continuous lines y_1, y_3, y_5, y_7, then we can compute their intersections. The points of interest are the intersections between $y_1 + y_5$ and $y_3 + y_7$. The intersection of $y_1 + y_5$ is:

$$\left[M_x^{(1+5)};M_y^{(1+5)}\right]=\left[\frac{I_5-I_1}{k_1-k_5};k_1\cdot\frac{I_5-I_1}{k_1-k_5}+I_1\right] \tag{4.22}$$

and the intersection between y_3 and y_7 is:

$$\left[M_x^{(3+7)};M_y^{(3+7)}\right]=\left[\frac{I_7-I_3}{k_3-k_7};k_3\cdot\frac{I_7-I_3}{k_3-k_7}+I_3\right] \tag{4.23}$$

These two points create an abscissa. If we compute the middle of this abscissa, we receive the centre of the fingerprint. Let us denote this centre point $[C_X;C_Y]$. Its coordinates are:

$$C_X = M_x^{(1+5)} + \frac{\sqrt{\left|\Delta M_y\right|^2+\left|\Delta M_x\right|^2}}{2}\cdot\cos\left(\arccos\left(\frac{\left|\Delta M_y\right|}{\sqrt{\left|\Delta M_y\right|^2+\left|\Delta M_x\right|^2}}\right)\right) \tag{4.24}$$

$$C_Y = M_y^{(1+5)} + \frac{\sqrt{\left|\Delta M_y\right|^2+\left|\Delta M_x\right|^2}}{2}\cdot\sin\left(\arccos\left(\frac{\left|\Delta M_y\right|}{\sqrt{\left|\Delta M_y\right|^2+\left|\Delta M_x\right|^2}}\right)\right) \tag{4.25}$$

where $\Delta M_x = M_x^{(1+5)} - M_x^{(3+7)}$ and $\Delta M_y = M_y^{(1+5)} - M_y^{(3+7)}$.

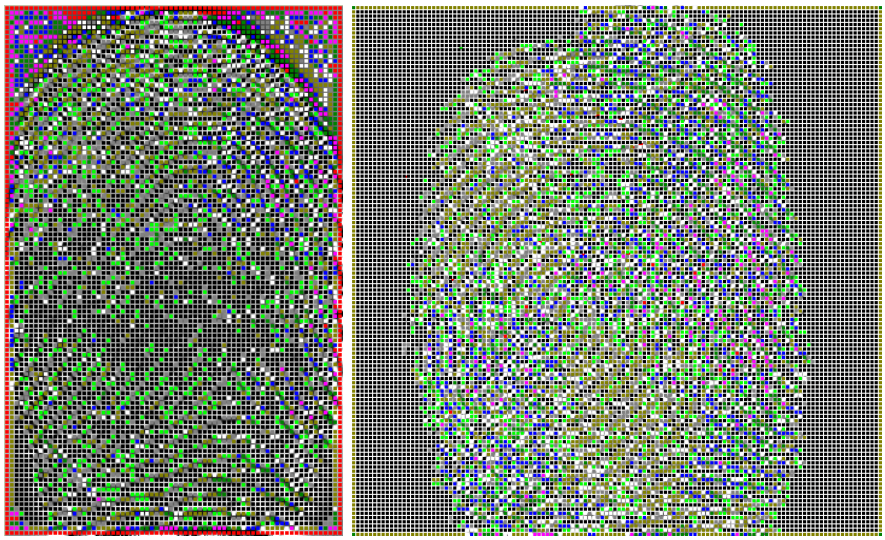


Fig. 4.8: Block orientation fields for Bergdata and dactyloscopic fingerprints

The point $[C_X;C_Y]$ denotes the centre of the fingerprint, not of the image. Thanks to the above mentioned steps, the background information has not been used for the fingerprint centre determination. The Equations (4.17) and (4.18) are responsible for the removal of background directional blocks. The following steps then compute the gravity centers for four directions, always from both respective opposite blocks in the image. The gravity centre points can be connected and expressed as continuous lines. The intersections of two lines with perpendicular directions are computed in the next step. These intersections create a short abscissa. Finally, the middle point of this abscissa denotes the centre point of the fingerprint, $[C_X;C_Y]$.

Assembly of Minutiae Set

The assembling of minutiae set proceeds from the centre of the fingerprint which has been computed as described above. This centre is relevant for further processing of the minutiae set.

After five repetitions of the minutiae set acquirement, some art of comparison of these sets should be made. Only those minutiae μ_i^j which can be identified in all the sets are allowed to be stored in the result set. The minimal threshold $\mu_{\min}$ for respective number of minutiae needs to be defined. The number of minutiae must not be lower than 12 (12 rule, see Chapter 3.3.4). It is reasonable to define this threshold higher, because additional sub-vectors will be generated in the next phase. If we have only 12 minutiae, we are not able to generate sub-vectors – the only one sub-vector is identical with the whole vector including these 12 minutiae. If we need the key with a particular length as the result of the next phase, we should be able to generate the sub-vectors of correspondent length, which means that we need certain number of minutiae to realize this generation; otherwise we will not be able to ensure the adequate key length. On the other side, we should consider the maximal number of minutiae in a fingerprint. This number is 368 (see Chapter 3.3.3 and Equation (3.29) for the matrix of $31\sigma_F \times 46\sigma_F$). Then the condition for the minimal threshold for respective number of minutiae (which is necessary for next phases) can be formulated as follows (n_μ = final number of minutiae present in a fingerprint):

$$12 < \mu_{\min} \leq 368 \text{ and } n_\mu \geq \mu_{\min} \tag{4.26}$$

Appropriate minutiae are included into the final minutiae set on the basis of the following conditions:

- In the first step, all those minutiae shall be included into the final minutiae set which can be found in each minutiae set of all (five) acquired sets at the same position. The additional condition is that not only the positions of such minutiae need to be the same, but their gradients and types, too. The minutiae lying closer to the centre of the fingerprint images $[C_X;C_Y]$ are preferred. If the condition (4.26) is fulfilled – the procedure is terminated, otherwise it is necessary to continue by the next step.

- In the second step, include those minutiae which can be found in each minutiae set of all (five) acquired sets at the same position but either their gradient or their type is different. Apply the majority rule, i.e. three or more information items from five (more as the half) must be same. The difference may exist only in one of the two parameters, i.e. either in their type or gradient but not in both together. The minutiae lying closer to the centre of the fingerprint images $[C_X;C_Y]$ are preferred. If the condition (4.26) is fulfilled – the procedure is terminated, otherwise it is necessary to continue by the next step.
- In the third step, include those minutiae which can be found in each minutiae set of all (five) acquired sets and have the same type and gradient, and which lie at slightly different positions but nevertheless within the tolerance box [22, 36]. The tolerance box is understood to be a close neighborhood in which the compared minutiae still can lie. The criterion for decision is respective distance which can be computed with the use of Eq. (4.2). If the appropriate distance is shorter than the preset threshold, then respective minutia can be considered as the same and shall be accepted. The position of such additionally accepted minutiae is then corrected by accepting the average position of all respective minutiae. The minutiae lying closer to the centre of the fingerprint images $[C_X;C_Y]$ are preferred. If the condition (4.26) is fulfilled – the procedure is terminated, otherwise it is necessary to continue by the last step.
- In this last step, include those minutiae which match in their position or gradient or type, when applying the majority rule, i.e. when still three or more information items from five (more as the half) are the same. Again, the minutiae lying closer to the centre of the fingerprint images $[C_X;C_Y]$ are preferred. If the condition (4.26) is not fulfilled at the end of this step, new acquirement is needed, i.e. new five fingerprints have to be scanned.

After these four steps, we obtain resulting final minutiae set, described as follows:

$$\mu = \left\{ \mu_i \,|\, \mu_i = \left(x_i, y_i, \phi_i, t_i\right), i = 1 \ldots n_\mu \right\} \tag{4.27}$$

where particular components are described by the Eq. (4.1). Only selected minutiae with high probability of repeated finding are in the minutiae set μ and the amount of this minutiae is equal to the number of real minutiae n_μ (see Eq. (4.26)).

4.2.2 Key Generation Phase

The biometric key based on fingerprint minutiae information will be generated in this phase. Some sub-steps are needed – such as estimation of the reference minutia, creation of the oriented closed graph, quantization and sub-graph generation.

Determination of Reference Minutia

From the previous steps, we have obtained the minutiae set μ, which includes already selected and proper minutiae with high quality, lying close to the centre of the fingerprint $[C_X;C_Y]$ (see Chapter 4.2.1). We need to generate an oriented closed graph [9] and therefore we need to decide on the origin of this oriented

graph. The origin can be defined as the starting vertex in the closed oriented graph and let us call it *Reference Minutia* and denote it μ_R.

The main condition for the reference minutia μ_R is its position – the closer is such minutia to the centre of the fingerprint $[C_X;C_Y]$ the higher is the probability that it will be selected as a reference minutia. The following equation can be used for the computation:

$$\mu_R = \left(\mu_m \middle| m \leftarrow \min\left(\sqrt{\left|C_X - x_i\right|^2 + \left|C_Y - y_i\right|^2} \right), i = 1 \ldots n_\mu \right) \tag{4.28}$$

where $[C_X;C_Y]$ is the centre of the fingerprint. The minutia lying closest to the centre of the fingerprint is denoted as the reference minutia μ_R.

Creation of Oriented Closed Graph

Oriented graph *G* is a pair of sets (*V*, *E*), where *V* is a set of vertices, and *E* is a set of edges between the vertices, and this graph does not allow self-loops; then *E* can be expressed as [9]:

$$E = \left\{ (u,v) \middle| u,v \in V, u \neq v \right\} \tag{4.29}$$

In addition to that, the graph is closed which means that the last vertex is connected with the starting vertex or origin (reference minutia μ_R). This makes possible to pass through the graph completely from its origin to its end and again to its origin.

The first task in the creation of the oriented closed graph is to reorder the minutiae. For the reordering of the minutiae, a new coordinate system will be needed. Let us call the old rectangular coordinate system CS_{old}. The origin of CS_{old} has been in the beginning of the image (in the upper left corner – see *Method based on the orientation field* in the Chapter 4.2.1). The new coordinate system, denoted CS_{new}, has the origin in the reference minutia μ_R. The transformation between both coordinate systems can be expressed as follows:

$$T : CS_{old} \rightarrow CS_{new} \left| O_{new}^X = O_{old}^X + x_R \;\&\; O_{new}^Y = O_{old}^Y + y_R \right. \tag{4.30}$$

where $[O_{old}^X ; O_{old}^Y]=[0;0]$ is the origin or starting position of the old coordinate system (the old position is [0;0]), and $[O_{new}^X ; O_{new}^Y]=[x_R; y_R]$ is the origin or starting position of the new coordinate system (the starting position is in the reference minutia μ_R).

On the basis of the transformation *T*, the *x* and *y* coordinates of all minutiae need to be recomputed as follows:

$$\mu^T = \left\{ \mu_i^T \middle| \mu_i^T = \left(x_i - x_R, y_i - y_R, \phi_i, t_i \right), i = 1 \ldots n_\mu \right\}^T \tag{4.31}$$

where μ^T is the new minutiae set using components of the minutiae set μ, which have been transformed using the transformation *T*. The components of the minutiae set μ_i^T are: $\mu_i^T = \left(x_i^T, y_i^T, \phi_i, t_i \right)$ The positional components of μ_i^T can attain negative values; this is the property of each coordinate system. Next steps of the

creation of the oriented closed graph exploit this property of the transformed minutiae set μ_i^T.

The starting position has been set to [0;0] (reason why i=1...n_μ in Eq. (4.31)) because it is more simple to use such coordinate system. That is also the reason, why the starting position of the new coordinate system is set to [0;0] after the transformation T. It means that the position of the reference minutia μ_R is then incorporated into the origin of the new coordinate system. The components of the reference minutia are then (in accordance with Eq. (4.31)): μ_R=(0, 0, -, -), i.e. the x and y coordinates are identical with those of the origin of CS_{new} and the information about the gradient and the type is no more important. The reference minutia μ_R is set as a starting vertex of the oriented closed graph G.

Now, the reordering of remaining minutiae points can be realized. The distances among reference minutia and other minutiae are computed first:

$$d_i^T = \sqrt{\left|x_R - x_i^T\right|^2 + \left|y_R - y_i^T\right|^2}, \quad i = 2\ldots n_\mu \tag{4.32}$$

The minutiae μ_i^T are reordered on the basis of the distance and will get only new indexes:

$$\mu^T{}' = \left\{\mu_i^T{}' \middle| d_i \le d_{i+1},\, i = 2\ldots n_\mu\right\} \tag{4.33}$$

The first minutia in $\mu^T{}'$ is the reference minutia, the next one is the first closest minutia to the reference minutia, the second one is the second closest, etc.

Now, the step for the rotational modification (the elimination of rotation) comes in consideration and this step consists in the rotation of all minutiae in the set $\mu^T{}'$. For the rotation, the angle α is used, which can be computed as follows:

$$\alpha = \frac{\pi}{2} - \arccos\left(\frac{\left|x_2^T{}'\right|}{\sqrt{\left|x_2^T{}'\right|^2 + \left|y_2^T{}'\right|^2}}\right) \tag{4.34}$$

where $x_2^T{}'$ and $y_2^T{}'$ are the coordinates of the second minutia. The angle α is the angle between the x-axis and the abscissa from the reference minutia to the second minutia $\mu_2^T{}'$. Using the angle α, we can rotate all components of $\mu^T{}'$:

$$\mu^T{}'' = \left\{\mu_i^T{}'' \middle| \mu_i^T{}'' = \left(x_i^T{}'', y_i^T{}'', \phi_i^T{}'', t_i\right),\, i = 2\ldots n_\mu\right\}, \text{ where} \tag{4.35}$$

$$x_i^T{}'' = x_i^T{}' \cdot \cos(\alpha) - y_i^T{}' \cdot \sin(\alpha)$$

$$y_i^T{}'' = x_i^T{}' \cdot \sin(\alpha) + y_i^T{}' \cdot \cos(\alpha)$$

$$\phi_i^T{}'' = (\alpha + \phi_i) \bmod 2\pi$$

All minutiae are translated, reordered (according to the distance from the first component in $\mu^T{}''$, i.e. the reference minutia) and rotated. Now we can write:

$$G = \left\{(\mu_i^T{}'', \mu_{i+1}^T{}'') \middle| \mu_i^T{}'', \mu_{i+1}^T{}'' \in \mu^T{}'' \;\&\; \mu_i^T{}'' \neq \mu_{i+1}^T{}'',\, i = 1\ldots n_\mu\right\} \approx \mu^T{}'' \tag{4.36}$$

By this procedure, an oriented closed graph has been created with the beginning and ending vertex in the reference minutia. Its components, i.e. the edges E, represent vectors starting in the actual minutia and pointing to the next minutia in the

set $\mu^{T''}$. The graph *G* can be considered as the full vector path, including all vertices (minutiae).

Quantization

The positions of $\mu^{T''}$ are prone to change by a very small distance. This change could be induced from stronger finger pressure on the sensor surface, humidity or dryness of the finger, maturing, and other factors. These factors change the distances among the papillary lines or reduce the quality of the image in general, with the resulting effect that the minutiae positions can vary. Therefore it is reasonable to reduce the precision of specification of minutiae positions. This step is called *Quantization*, concrete quantization of minutiae positions. The influence of quantization step on the strength of fingerprint information is described in the Chapter 3.3.5. Using Equation (3.35), it is possible to compute the topological quantization. Reduced strength of fingerprint information is shown in the Tab. 3.5, Fig. 3.13 and Fig. 3.14.

The method of quantization is schematically shown in the Fig. 3.12. The original pixel matrix has 230×350 points. For the key generation, it is not necessary to have so big amount of positions. This is the reason for the application of quantization. The quantization should be understood as the application of rougher matrix mask on the original matrix (230×350 points). There are 80.500 possible positions in the original matrix, where the minutiae can be placed. When we use a mask matrix with the cell dimensions of κ_x and κ_y, the number of original positions decreases as follows:

$$\left(\frac{Width}{\kappa_x} \times \frac{Height}{\kappa_y} \right) = p_\kappa \tag{4.37}$$

where p_κ is the number of positions in the rough matrix mask. When deciding on the dimensions κ_x and κ_y, we should take in consideration the condition that no two minutiae can lie in the same cell! For this purpose, we can apply the theorem of minutia and antiminutia as described in the Chapter 3.3.3. This theorem defines that an antiminutia should lie between two minutiae and if the resolution for minutiae is σ_F and the resolution of sensor σ_S, then the maximal value for κ_x and κ_y is 7.

New places in the rough matrix mask are generated, using the principle shown in the Fig. 4.9. The first cell is placed in the block I., the second in the block II., the next in the block III. and the last in the block IV. Then the fifth cell is placed again in the block I., right of the previous cell in the same block. Cells are inserted in the rough matrix mask according to spiral principle. The generation of this spiral rough matrix mask runs until the last minutia is positioned in some cell. Then, it is not necessary to continue in generating further cells. The maximal number of cells can reach the value *Width*×*Height*, when $\kappa_x = \kappa_y = 1$.

The minutiae from the set $\mu^{T''}$ are then positioned to the particular cells of the rough matrix mask. The structure of the minutiae set change as follows:

$$\zeta = \left(n_i^\zeta, \phi_i^T, t_i \right),\ i = 1 \ldots n_\mu \tag{4.38}$$

where n_i^{ζ} is the cell number in the rough matrix mask. The *x* and *y* coordinates are not saved any more. The application of rough matrix mask modifies the granularity of the minutiae position system (it will be modified only in the interval 1 to 7). Resulting influence on the strength of minutiae information has been discussed in the Chapter 3.3.5.

The structure of the graph *G* needs to be modified accordingly. The vertices are only the cell numbers in the rough matrix mask now, but the order of minutiae remains the same. The first minutia in the set ζ is the reference minutia, the second is the nearest minutia to the reference minutia, etc. The graph is therefore modified as follows, using the Eq. (4.29):

$$E = \left\{(u,v) \middle| u,v \in V \,\&\, u \neq v\right\} = \left\{(n_i^{\zeta}, n_{i+1}^{\zeta}) \middle| n_i^{\zeta}, n_{i+1}^{\zeta} \in \zeta, n_i^{\zeta} \neq n_{i+1}^{\zeta}\right\} \tag{4.39}$$

where $V = \zeta$, $i = 1 \ldots n_{\mu}$ and $G = (V, E)$.

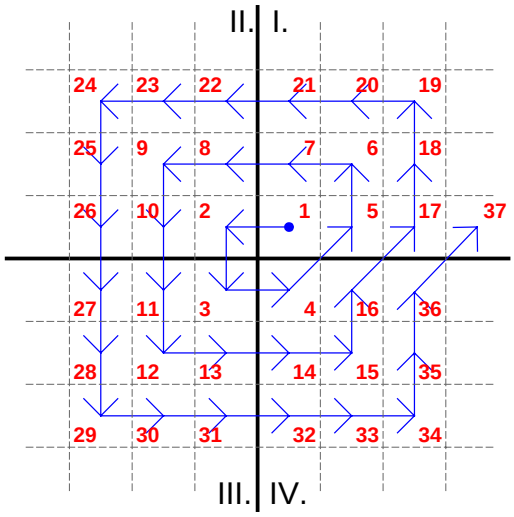


Fig. 4.9: Principle of creation of rough matrix mask

Sub-Graph Generation

At the beginning of the sub-graph generation, we have a complete oriented closed graph $G = (V, E)$, see Eq. (4.39). This graph has been generated on the basis of the minutiae set ζ. It is difficult to repeat the full path in the complete graph (due to the variations in each fingerprint scan). Therefore it is recommended to build or create sub-graphs, which include sub-vectors. These sub-graphs are also oriented and have a closed path (inherited properties of the parent graph). The sub-graphs should not be too small, because the possibility that they can be found in another graph based on totally different fingerprint would be otherwise high. On the other side, the vectors should be not too large, because the probability that they may not be found in the fingerprint of the same finger next time would be again very high. Of course, this should be taken into consideration with regard to the number of all possible sub-graphs which can be generated from the parent graph. The number of these sub-graphs could be computed as the amount of combinations without repeating [44, 43], using following equation:

$$Nr_{Combinations} = \binom{m}{k} \tag{4.40}$$

where m is the number of all minutiae of the complete graph ($m = n_\mu$) and k is the number of minutiae in the sub-graph. The sub-graphs of the parent graph need to cover all possible combinations. It is clear that the amount of sub-graphs can reach very high values; hence the number of sub-graphs should be somehow limited, considering time consumption and data volume.

The sub-graph can be mathematically described by applying the following expressions:

$$G^r_{Sub} = (V^r_{Sub}, E^r_{Sub}),\ G^r_{Sub} \subset G \tag{4.41}$$

where

$$V^r_{Sub} = \{\zeta_1, \ldots, \zeta_k\}\ \&\ E^r_{Sub} = \left\{ \left(n^\zeta_i, n^\zeta_{i+1}\right) \middle|\ n^\zeta_i, n^\zeta_{i+1} \in V^r_{Sub}\ \&\ n^\zeta_i \neq n^\zeta_{i+1} \right\} \tag{4.42}$$

where r denotes one of the combinations, i.e. $r = 1 \ldots Nr_{Combinations}$, and k is the number of used sub-vertices for the sub-graph (see Eq. (4.40)).

These sub-graphs represent the individual biometric keys. Each sub-graph of G is a set of components from ζ. Then the biometric key set will be as follows:

$$K_r = \left\{ \left(n^\zeta_l, \phi^T_l, t_l\right)_r, \ldots, \left(n^\zeta_o, \phi^T_o, t_o\right)_r \right\} \text{ and } K = \left\{ K_r \middle|\ r = 1 \ldots Nr_{Combinations} \right\} \tag{4.43}$$

where K is the set of all biometric keys, K_r is the r^{th} combination and l & o assume the values from the interval <1, n_μ> ($l < o$), whereas the number of components is k (see Eq. (4.40)). Each biometric key (or sub-graph G^r_{Sub}) contains ordered (relation is lower as (<)) k items from the graph G and the number of biometric keys is $Nr_{Combinations}$, whereas r runs from 1 to $Nr_{Combinations}$.

4.2.3 Cryptomodule Phase

Hash Computation

Hash functions take a message as an input and produce an output referred to as a *hash*. A *hash function*, h, is a function which has at least the following two properties [77]:

- Ability to compress – h transforms an input x with an arbitrary finite bit length, to an output $h(x)$ with a fixed bit length n.
- Ease of computation – given h and an input x, $h(x)$ is easy to compute.

There are two types of hash functions – *Modification Detection Codes* (MDCs) and *Message Authentication Codes* (MACs). In our case, the MDCs are important. The purpose of an MDC is to provide a representative image of a message (data integrity assurance), while satisfying additional properties (for inputs x, x' and outputs y, y') [77]:

- Pre-image resistance – for essentially all pre-specified outputs, it is computationally infeasible to find any input which would be transferred to that output, i.e. to find any pre-image x' such that $h(x')=y$, i.e. for any given y no corresponding input can be found.

- Second pre-image resistance – it is computationally infeasible to find any second input, which has the same output as any specified input, i.e. for any given x, it is impossible to find a second pre-image $x' \neq x$ such that $h(x) = h(x')$.
- Collision resistance – it is computationally infeasible to find any two distinct inputs x, x', which would be transferred to the same output, i.e. such that $h(x) = h(x')$.

The subclass of MDCs is the *one-way hash function* which is a hash function h as defined above with additional properties (pre-image resistance and second pre-image resistance). Additional properties of one-way hash functions, as non-correlation, near-collision resistance and partial pre-image resistance, can be found in [77]. This type of hash functions will be used for further computations. The following four MDC hash functions can be mentioned as examples (the number of output bits of hash value is in brackets) [77]: MD4 (128), MD5 (128), RIPEMD-128 (128), SHA-1 or RIPEMD-160 (160).

Individual components from the biometric keys set K_r are taken as the input of the hash function h. After the computation of all hashes of all components from K_r, the output set is defined:

$$H = \left\{ h(K_r) \,\middle|\, K_r \in K \;\&\; r = 1 \ldots Nr_{Combinations} \right\} \tag{4.44}$$

where H is the whole set of hash values from all biometric key items from K. In H, only hash values of respective biometric key are saved. The positions of items in H correspond to the positions of items in K, i.e. the hash value in H on the i^{th} position can be repeatedly computed from the i^{th} item in K. The set H has again $Nr_{Combinations}$ components.

Secret Encryption

At the beginning of this section, some definitions are needed. A *product cipher* combines two or more transformations in such a manner that the resulting cipher is more secure than its individual components [77]. A *substitution-permutation* is a product cipher composed of certain number of stages, each involving substitutions and permutations [77]. An *iterated block cipher* is a block cipher involving the sequential repetition of an internal function called a round function. Parameters include the number of rounds, block bit size, and bit size of input key from which round keys are derived. A *Feistel cipher* is an iterated cipher transferring a $2t$-bit plaintext (L_0, R_0), for t-bit blocks L_0 and R_0, to a ciphertext (R_r, L_r), through an r-round process, where $r \geq 1$. For $1 \leq i \leq r$, round i transfers $(L_{i-1}, R_{i-1}) \rightarrow (L_i, R_i)$ as follows: $L_i = R_{i-1}$, $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$, where each sub-key K_i is derived from the cipher key K. For example, DES (*Data Encryption Standard*) cipher is a Feistel cipher which processes plaintext blocks of 64 bits, producing 64-bit ciphertext blocks. The effective size of the secret key K is 56 bits. Further description of the DES algorithm can be found in [77]. Another Feistel cipher is FEAL (*Fast Data Encipherment Algorithm*), which transfers 64-bit plaintext to 64-bit ciphertext blocks under a 64-bit secret key [77].

In our case, we use some information which should be protected as a plaintext. Two examples of conceivable plaintext are: a private key from another application

(e.g. asymmetrical cryptography) or photography of the user. The bit length of input information is unimportant, because all block algorithms can work with every piece of information of finite length. One selected key from the set *K* is used as a secret key for the cipher-plaintext-processing. After respective encipherment, the output (a ciphertext) has the same length as the plaintext.

If we call the plaintext (or open secret) as *P*, and the ciphertext (or enciphered secret) as *C*, then we can define the encryption process *g* as the transformation:

$$g : P \overset{K_r}{\rightarrow} C, C = g_{K_r}(P) \tag{4.45}$$

where K_r is the particular key from the set *K*, $r = 1...Nr_{Combinations}$. Using Eq. (4.44), we can generate the following set:

$$S = \left\{ g_{K_r}(P) \middle| K_r \in K, r = 1...Nr_{Combinations} \right\} \tag{4.46}$$

This set *S* contains $Nr_{Combinations}$ of enciphered versions of the plaintext *P*, always using another key K_r. The positions of items in *S* correspond to the positions of items in *H* and *K*, i.e. the ciphertext in *S* on the i^{th} position can be repeatedly decrypted using the i^{th} item in *K*. The set *S* has again $Nr_{Combinations}$ components.

Certificate Creation

The first version of X.509 was defined in 1988 and further extended and improved. The basic structure of the X.509 certificate is shown in the Figure 4.10:

X.509 Version 1	Version Serial Number Signature Issuer Validity Subject Subject Public Key Info
X.509 Version 2	Issuer Unique ID Subject Unique ID
X.509 Version 3	Extensions

Fig. 4.10: Basic structure of X.509 certificate [105]

The basic structure of X.509 certificate is used for the hash computation. The hash is then signed and attached to the basic structure; these two parts together form the certificate. The last part, called “Extensions”, improves the flexibility of the certificate. It makes possible to add some information which is not included in the basic structure of this certificate standard. Each extension must have a unique object identifier, part attribute called “critical” and a bit string (the content). The flag “critical” is important, because the client, who does not know the extension denoted as “critical”, must see the certificate as invalid and must reject it. On the other hand, any non-critical extension can the client see only as a note and such note can be completely ignored. The certificate X.509 is recognized as a standard. Further information on this certificate can be found in [105, 1, 94] or in other RFC descriptions.

We use the basic structure of the X.509 certificate as the fundamental structure and the part of extensions is used for the storage of our data. The certificate is slightly different in our case, as the certificates are normally stored on the smart card or some token with small data capacity. But we have generated $Nr_{Combinations}$ of couples (H_r, S_r). This is very big amount of data – when considering e.g. a photography as a plaintext, then the data size can be tens of megabytes. The main idea is to have all data signed by a certification authority [105]; thus confirming the confidentiality of the data stored in the certificate. Very important condition for the whole process of certificate creation is that the whole process must to be realized by an administrator or a certification authority in order to ensure the confidentiality of signed data.

The part "Extensions" of the certificate looks as follows:

$$Ext = \left\{id, false, \left\{(H_r, S_r) \middle| H_r \in H, S_r \in S, r = 1 \ldots Nr_{Combinations}\right\}\right\} \tag{4.47}$$

The whole basic structure as shown in the Figure 4.10, including the part "Extensions" according to Eq. (4.47), forms the basis for the computation of the hash value which is then signed by the certification authority (an administrator) and appended to the basic structure thus completing the real certificate.

4.3 Certificate Usage Concept

The concept describing the certificate usage concerns the daily usage of this certificate. The phase of usage can be applied repeatedly; it requires no action of the certification authority (or the administrator). The only purpose is, if the confidentiality of the certificate can be proved, to present the public key of the certification authority (or the administrator), which (or who) has signed the certificate. The schema of this concept is shown in the Figure 4.11.

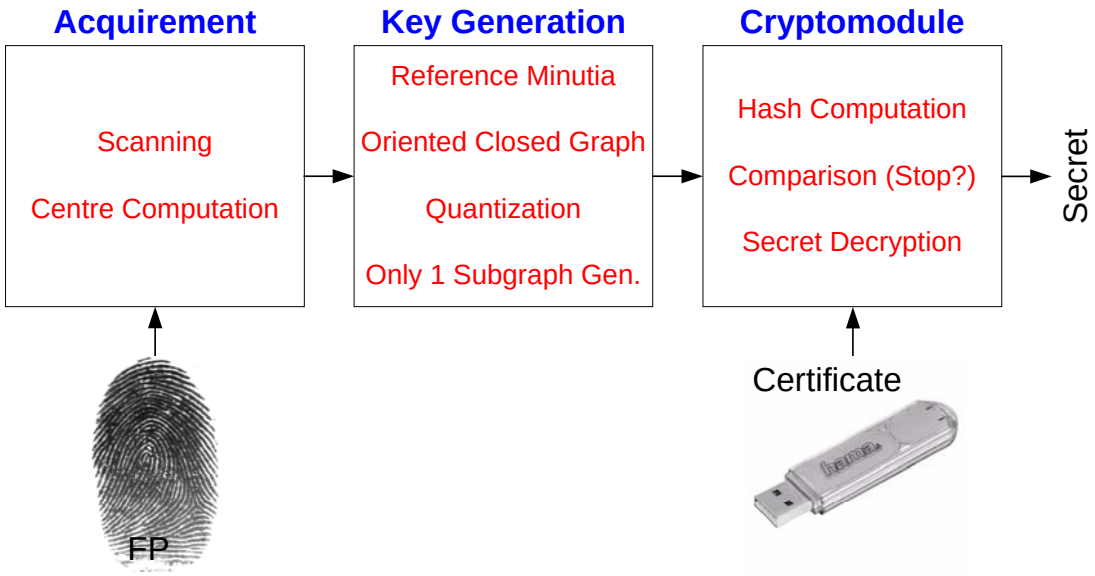


Fig. 4.11: Certificate Usage Concept

4.3.1 Acquirement Phase

The acquirement phase is very similar to the acquirement phase of the concept for the Certificate Creation. The only difference is that only one fingerprint is scanned and appropriate minutiae are extracted. Of course, the same algorithms for fingerprint acquirement, image enhancement and processing + minutiae extraction are required.

The minutiae are extracted from the fingerprint and then the centre of the fingerprint needs to be computed. Any of the previously discussed methods (Method based on the minutiae gravity centre, Method based on the orientation field or Method based on ridge count) can be used for computing the centre. The method based on the minutiae gravity centre uses all the extracted minutiae as they are. And the methods based on the orientation field and ridge count need only the fingerprint image with 256 gray-levels to determine the centre of fingerprint. No minutiae are needed for the second and third method. Let the centre of fingerprint has the coordinates $[C_X;C_Y]$. No minutiae set assembly is made here – we have only one set of minutiae from a single fingerprint.

The set of minutiae from the acquirement phase, when proceeding from the Equation (4.27), can be expressed as:

$$\omega = \left\{ \omega_i \mid \omega_i = \left(x_i, y_i, \phi_i, t_i \right) \middle| \, i = 1 \ldots n_\omega \right\} \tag{4.48}$$

where n_ω is the number of all extracted minutiae.

4.3.2 Key Generation Phase

Similarly as in the Chapter 4.2.2, the reference minutia needs to be found. For the computation, the Equation (4.28) can be used:

$$\omega_R = \left(\omega_m \middle| \, m \leftarrow \min\left(\sqrt{\left| C_X - x_i \right|^2 + \left| C_Y - y_i \right|^2} \right), i = 1 \ldots n_\omega \right) \tag{4.49}$$

where $[C_X;C_Y]$ is the centre of respective fingerprint, n_ω is the number of extracted minutiae and ω_R is the reference minutia, $\omega_R \in \omega$.

The principle of creation of the oriented closed graph is again the same as in the Chapter 4.2.2. The whole computational process will not be repeated; only the result is presented here, using Equation (4.36):

$$G' = \omega' = \left\{ \left(\omega'_i, \omega'_{i+1} \right) \middle| \, \omega'_i, \omega'_{i+1} \in \omega' \;\&\; \omega'_i \neq \omega'_{i+1} \right\}, \; i = 1 \ldots n_\omega \tag{4.50}$$

where ω' is the original set (ω) after the transformation and reordering of coordinates (see Eq. (4.33) and (4.35)), and n_ω is the number of extracted minutiae.

Of course, the quantization needs to be done even in this concept. The same principles for quantization is used (see Chapter 4.2.2, section Quantization). The only condition is that the rough matrix mask must have the same cell size, i.e. the number of cells in both cases must be the same! After the quantization, we get a similar set as in Eq. (4.39), except for that only the components from ω' are used,

whereas the positions *x* and *y* are transferred into the cell identificator, as in Equation (4.38).

The first difference in this concept is the sub-graph generation. We have the graph *G'* (Equation (4.50)). The difference in relation to the Certificate Creation concept is that only one sub-graph is generated. One possibility of combinations from the whole graph is absolutely randomly selected and the corresponding sub-graph is extracted. The dimension of the sub-graph must be the same as the dimension of all sub-graphs from the certificate creation concept. That means, the sub-graph, G'_{Sub}, of G' is generated similarly as in Equations (4.41) and (4.42), using the minutiae ω', instead of ζ. This sub-graph represents one of all possibilities $Nr'_{Combinations}$ (using Eq. (4.40) and $m = n_{\omega}$ and k has been left the same). Then we can consider this sub-graph G'_{Sub}, as a biometric key:

$$K' = \left\{(n_l^{\omega'}, \phi_l^{\omega'}, t_l^{\omega'}), \ldots, (n_o^{\omega'}, \phi_o^{\omega'}, t_o^{\omega'})\right\} \tag{4.51}$$

where *l* is the starting position and *o* is the end position of the sub-graph in relation to the graph *G'*. In this case, only one biometric key *K'* exists and no other keys are generated. The size of *K'* must be same as the size of the key K_r in the Equation (4.43).

4.3.3 Cryptomodule Phase

The cryptomodule phase is also slightly different when compared with the same phase of the Certificate Creation concept.

Hash Computation and Comparison

First of all, the hash value of the key *K'* is computed, using the definition and properties of the hash function, as mentioned in the Chapter 4.2.3. The hash value is defined by the expression:

$$H' = h(K') \tag{4.52}$$

where *h* is the hash function.

The following task – the comparison – is new. We have the hash value *H'* and the hash values *H* (Eq. (4.44)) have been stored in the certificate. The first step is to check the validity of the certificate. It could be done using the public key of the certification authority (the administrator). The validity test is done in a normal way; see [105, 1, 94]. If the validity cannot be confirmed, then the certificate is not confidential and no further steps are done. But if the validity of the certificate has been proven, then the comparison can be done. The comparison consists in searching in the set *H* (Equation (4.44)) for the occurrence of the value *H'*. We can obtain two possible results:

- A match has been found. It means that the hash value *H'* has been found in *H* (in the certificate). This result informs us that the key *K'* can be used for the decryption.
- A match has not been found. The occurrence of *H'* in *H* has not been confirmed. It means only that this sub-graph is not the right one. But we have in

total $Nr'_{Combinations}$ possibilities. Then the sub-graph generation step of this concept needs to be done repeatedly. In the next call of the sub-graph generation, another sub-graph needs to be generated. The repetition can be done only $Nr'_{Combinations}$ times. If we run out of all the $Nr'_{Combinations}$, then no sub-graph has built the right key and the decryption step cannot be processed or completed.

Secret Decryption

If a match has been found, i.e. the first condition of the previous list has been fulfilled and then the decryption can start. Since not only the hash value has been saved in the certificate, but there are pairs – hash value of the biometric key and the encrypted secret using the biometric key, then we can decrypt the corresponding secret, using the key *K'* as follows:

$$q : C \overset{K'}{\rightarrow} P, P = q_{K'}(C) \tag{4.53}$$

where *C* is the encrypted secret (ciphertext) and *P* is the plaintext. The decryption step needs to use the same cryptographic algorithm as it has been already used for the encryption (Equation (4.46)). If the decryption process is successful, we obtain the original plaintext (private key or photography), which has been stored in the certificate.

4.4 Proposal of Practical Usage

Two practical possibilities of usage of the biometric certificate will be described in this section. Two possible proposals are as follows:

- *Private key protection*. The first possibility is the protection of some private key. Let us assume that some application based on PKI (Public Key Infrastructure) has generated a key pair. The public key is publicly accessible, but the private key needs to be protected. Nowadays, the directions of PKI for storage of private key are used. One condition defines that the private key cannot be saved in an open form and should not be readable by unauthorized person. These conditions are guaranteed in our system, when the secret part (in this case the private key) is encrypted and saved only in the encrypted form. The pairs – hash value from the biometric key and the encrypted private key (using the corresponding biometric key) – are saved. Thus the certificate has been created. When the private key is requested by another application, the fingerprint needs to be scanned and the biometric key is generated. The comparison of the hash value from this key with the saved hash values in the certificate is made. If a match is found, then the biometric key corresponding to the hash value can be used to decrypt the private key, which can be then delivered to the requesting application.

- *Personal document extension*. The second proposed possibility is important for new extension of the information in the personal documents, with regard to biometrics. Of course, the image of biometric attribute (e.g. a fingerprint, face image, etc.) could be stored on the chip in the personal document, but we can

use some art of certificate as described in the above parts. The personal document can include a smart chip on which all data can be stored (the capacity of this chip medium is not important in this consideration – nowadays there are already available smart cards or tokens with adequate data capacity). The certificate can be stored on this chip including all data as described in the certificate creation concept, i.e. the personal information data and hash values of the sub-graphs generated from the fingerprint of the user. Let us consider the photography of the user as the secret part (plaintext). This photography can be encrypted using all keys and saved in the certificate. This certificate creation phase must be done by a certification authority (the best option is – a state certification authority). Then the user obtains his or her personal document. Let us consider the passport control as an example of usage. The user passes his passport to an officer who checks the data in the paper form. But moreover, the data stored in the chip are read during the control (data are scanned and shown on the monitor). The user is requested to authenticate such data – verify his identity with the identity information on the chip. He must let his finger to be scanned. This biometric information is then processed and the sub-graph generated. If the match in the hash values set is found, the photography of the user can be deciphered using this biometric key and compared visually with the photography scanned from the paper form and with the real visage of the user. If some difference is found, the user would not pass through and would be then checked more strictly.

As shown in the Fig. 4.12, the cryptomodule can combine two or more additional biometric attributes. The encryption / decryption needs to be done in a cascade. The conjunction of all keys together and their usage can bring difficulties [27, 28]. For example, if we combine all keys together, then a very big number of combinations can arise and this can make this method impracticable. Of course, each art of extension of the encryption process can lead to the refusal of the right encryption of the secret *S*. These problems are discussed further in [27, 28, 31, 35].

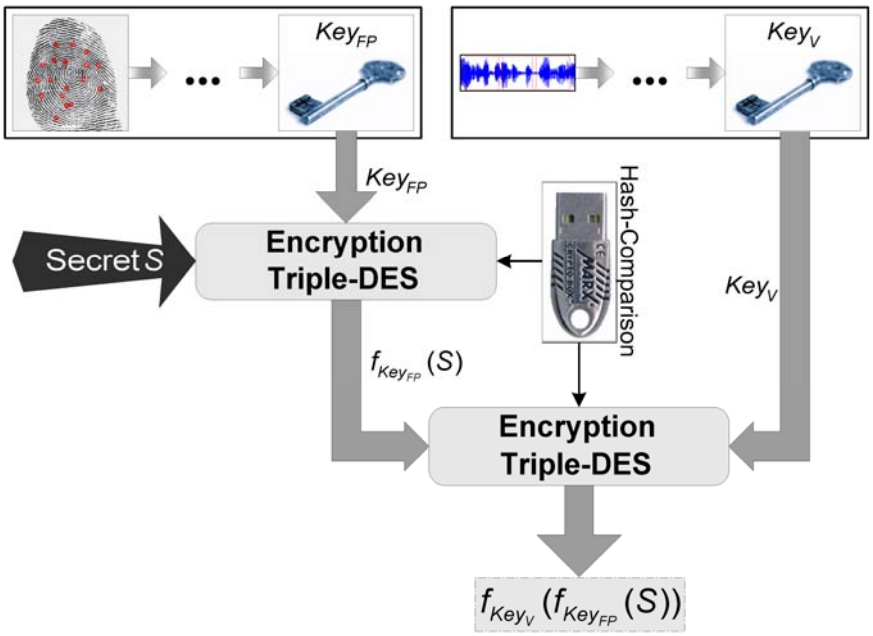


Fig. 4.12: Cryptomodule using more biometric attributes (fingerprint + voice)

5. Practical Results and Summary

This final chapter describes the acquirement of the fingerprint database, the reliability testing using some industrial algorithms and testing of my own applications applied to this fingerprint database.

5.1 Fingerprint Database

Three sensors (from companies Bergdata, SecuGen and Veridicom) have been applied for the acquirement of the fingerprint database. Dactyloscopic cards (with their own format for rolled fingerprints) were used as the fourth input for the database. A brief description of the sensors used for the acquirement follows (see also Fig. 5.1):

- **Bergdata FCAT-100**[1]
  - Resolution: 500 *dpi*
  - Sensor technology: thermal – sweep (Atmel FingerChip™)
  - Output image: 280 × 440 pixels, 256 gray-scales, BMP-Format

- **SecuGen Hamster™ II**[2]
  - Resolution: 500 *dpi*
  - Sensor technology: optical (SecuGen FDU02)
  - Output image: 260 × 300 pixels, 256 gray-scales, screen shot

- **Veridicom 5th Sense PRL**[3]
  - Resolution: 500 *dpi*
  - Sensor technology: capacitive (Veridicom)
  - Output image: 300 × 300 pixels, 256 gray-scales, RAW-Format

Each sensor has been delivered with corresponding software. The sensors Bergdata (FP-SDK-HDW-FC) and Veridicom (SDK Version 2.7) have included SDKs. Both SDKs have been used for the implementation of my own acquirement applications (see the file "*Applications.pdf*"). The Bergdata application stores the fingerprints as BMP images. The Veridicom application allows only the storage in RAW format. The delivery of SecuGen sensor has included only a demo program which does not allow the direct storage of the fingerprint image. The images have been cut out from the screen shots and then saved in an arbitrary format. The dactyloscopic cards were scanned in the resolution of 600 *dpi*. Appropriate images have been manually cut out and saved as gray-scale images in TIFF format.

[1] http://www.bergdata.com/
[2] http://www.secugen.com/
[3] http://www.veridicom.com/

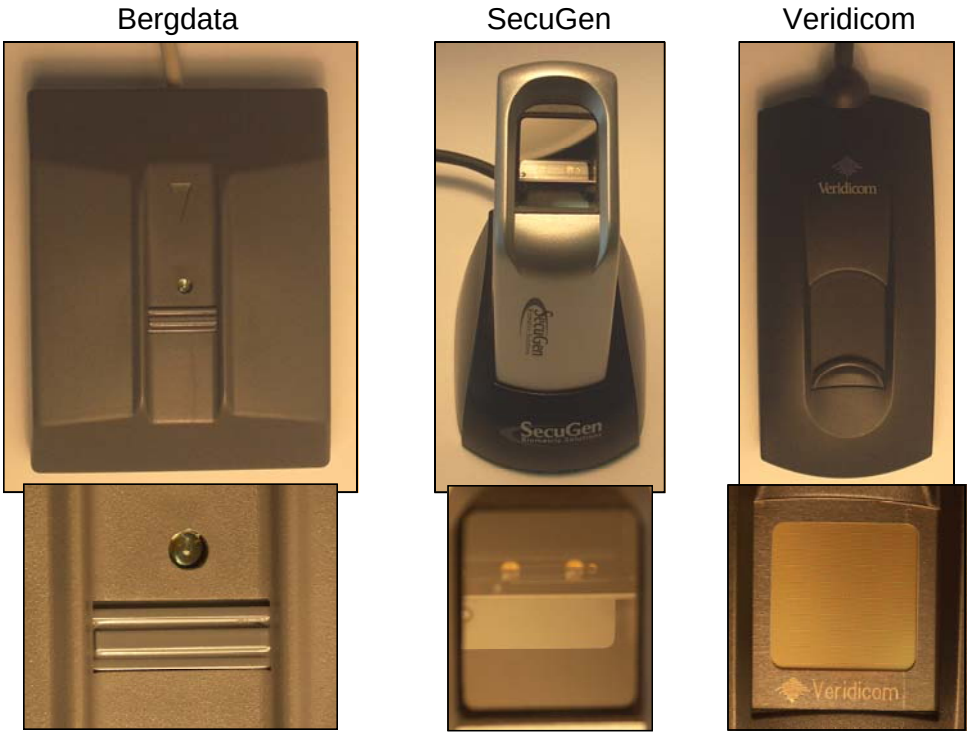


Fig. 5.1: Applied fingerprint sensors

Examples of fingerprint images from all database inputs are shown in Fig. 5.2. Each fingerprint file has different image height and width, namely: Bergdata (440 × 280 pixels, 123kB), dactyloscopic (512 × 512 pixels, 262kB), SecuGen (300 × 260 pixels, 78kB) and Veridicom (300 × 300 pixels, 90kB). All images use 256 gray-scales. The images from the Bergdata sensor are rotated by 180° (the top of the fingerprint is in the bottom of the image) – this attribute has been left without change, while all Bergdata images are rotated – therefore unimportant attribute. The rolled fingerprint method (see Chapter 2.1.1) was used nearly in all cases for dactyloscopic fingerprints acquirement what means that more delta points (see Chapter 2.1.2) can be found in these fingerprints. The probability to find the deltas in the images from other three sensors is lower, but not zero. The area occupied by fingerprints in dactyloscopic images is larger than the corresponding area occupied by images from common sensors; this also means that the former fingerprint images offer more information (more papillary lines).

Fig. 5.2: Fingerprint images (Bergdata, dactyloscopic, SecuGen, Veridicom)

The fingerprint database includes N_{User} = 10 users. Two fingerprint sensors (Bergdata, Veridicom) have been applied by each user and the fingers of the users have been impressed on dactyloscopic cards (see Fig. 5.3)(N_{Sensor} = 3). The fingerprints from the SecuGen sensor were stored only by one user (because the storage of images has not been supported). The record of each user in the database contains both hands (N_{Hand} = 2) and N_{Finger} = 4 fingers (index finger, middle finger, ring finger and thumb). Small fingers are not considered [Arn04]. In each session (1 user and 1 sensor), total N_A = 65 fingerprints of each finger were stored. The number of all acquired fingerprints is:

$$N_{AFP} = N_{User} \cdot N_{Sensor} \cdot N_{Hand} \cdot N_{Finger} \cdot N_A = 10 \cdot 3 \cdot 2 \cdot 4 \cdot 65 = 15.600 \quad (5.1)$$

This number (N_{AFP}) should be increased by the amount of fingerprints from the SecuGen sensor (inclusive small finger). The whole amount of all acquired fingerprints is then:

$$N_{AFP} = 15.600 + (1 \cdot 1 \cdot 2 \cdot 5 \cdot 65) = 16.250 \quad (5.2)$$

In total, 16.250 fingerprints were acquired and stored (the database size is ~2,5GB (this size relates only to images without templates – see Chapter 1.2)).

Fig. 5.3: Example of dactyloscopic card

As the users were not used to work with the sensors (in particular, sweeping was difficult) and the rolled method for acquirement of dactyloscopic fingerprints presented also certain problems, some fingerprints are of very low quality. Hence, it was necessary to reduce the database size. The final number of fingerprints per finger was set to N_N = 50 (instead of 65). This reduction step was done manually (images were examined and those of low quality deleted) and this means that the final number of all fingerprints in the database is:

$$N_{FP} = N_{User} \cdot N_{Sensor} \cdot N_{Hand} \cdot N_{Finger} \cdot N_N = 10 \cdot 3 \cdot 2 \cdot 4 \cdot 50 = 12.000 \quad (5.3)$$

When considering 500 fingerprints from the SecuGen sensor, the total final amount of fingerprints is $\boldsymbol{N_{FP}}$ **= 12.500**.

5.2 Database Enrollment and Matching (Industrial Algorithms)

Two industrial algorithms (Siemens and Veridicom) have been used for database quality testing. Both of them have been used for the enrollment of fingerprints (creation of templates) and matching (comparison of templates) – see Chapter 1.2. The analysis of applicable rates for both algorithms is included further in this chapter.

Failure to Acquire Rate (**FTA**)

The **FTA** rate is related to the frequency of a failure to acquire and means the failure of biometric sensor to capture the biometric data (in our case fingerprints). The brief description of this rate can be found in Chapter 1.3. This rate has no relation to respective industrial algorithms because it is computed only in the acquirement phase where no enrollment and no matching are done. The numbers of refused acquirement attempts for each user are shown in the Tab. 5.1. The computation of **FTA** has been done as follows:

$$FTA_{Sensor} = \frac{\sum refused\ acquirement\ attempts}{\sum all\ acquirement\ attempts} \tag{5.4}$$

Tab. 5.1: **FTA** rates for respective sensors

FTA [%]	Bergdata	SecuGen	Veridicom
User_01	2	-	1
User_02	0	-	0
User_03	1	-	0
User_04	0	-	1
User_05	0	-	0
User_06	1	-	1
User_07	0	0	0
User_08	0	-	0
User_09	0	-	0
User_10	1	-	0
FTA_{Sensor}	**0,77%**	**0,00%**	**0,46%**

Data for dactyloscopic fingerprints are missing in the Table 5.1. The images of fingerprints, in dactyloscopic case, are only impressions on paper and therefore we cannot discuss the cases of failure to acquire a fingerprint. That's why no **FTA** rate for dactyloscopic fingerprints exists. All acquirement refusals mean the inability of the sensor (hardware item) to deliver the output data. No software control of the image quality or fingerprint's area has been done because such control is reserved for another rate – **FTE**.

Failure to Enroll Rate (**FTE**)

The **FTE** rate is related to the frequency of a failure to enroll and means the inability of the system to extract the biometric data (in our case fingerprint's features) for biometric record keeping (creation of a template). **FTE** depends strongly on the quality of image. Some images consist only of latent fingerprints left on the sensor surface, other fingerprints have been deformed (quite frequent case for sweep sensors) or the area of the fingerprint is too small (only very limited part of papillary structures was scanned).

The enrollment phase has been applied on the final database (50 fingerprints per finger). The threshold of minimum quality for the enrollment by the Veridicom software has been set to 20 (the maximum value is 100). The quality has been computed internally in the Veridicom's SDK kernel. Two thresholds have been set in the application of Bergdata – the minimum fingerprint quality threshold to 60 (the maximum value is 100) and the image area threshold to 100 (the maximum value is 300). The concrete settings of the Siemens algorithm for enrollment are unknown, but special configuration files for corresponding sensors have been used.

The **FTE** rates for corresponding sensors and algorithms are shown in the Tab. 5.2. The computation of **FTE** has been done as follows:

$$FTE = \frac{\sum refused\ enrollment\ attempts}{\sum all\ enrollment\ attempts} \tag{5.5}$$

Tab. 5.2: **FTE** rates for respective sensors and algorithms

	Bergdata			Dactyloscopic		SecuGen		Veridicom	
FTE [%]	**FTE_S**	**FTE_B**	**FTE_V**	**FTE_S**	**FTE_V**	**FTE_S**	**FTE_V**	**FTE_S**	**FTE_V**
User_01	0,00	10,00	0,00	0,00	0,75	-	-	5,75	0,00
User_02	1,00	2,00	0,00	0,00	13,75	-	-	47,25	0,00
User_03	9,75	8,25	0,00	0,00	0,00	-	-	0,00	0,00
User_04	0,00	1,25	0,00	0,00	11,00	-	-	5,50	0,00
User_05	11,25	2,00	0,00	0,00	1,25	-	-	17,50	0,00
User_06	14,25	4,75	0,00	0,00	0,00	-	-	4,50	0,00
User_07	0,00	0,75	0,00	0,00	0,00	0,00	0,00	0,00	0,00
User_08	0,00	0,18	0,00	0,00	0,00	-	-	21,75	0,00
User_09	5,25	2,75	0,00	0,00	0,00	-	-	0,00	0,00
User_10	0,00	7,75	0,00	0,00	2,00	-	-	15,25	0,00
FTE_{Total}	**4,15**	**3,97**	**0,00**	**0,00**	**2,88**	**0,00**	**0,00**	**11,75**	**0,00**

When we compute the **FTE** rates in relation to a sensor or to an algorithm, we obtain following results:

- Sensor Bergdata: **FTE_{SB}** = 2,71 %
- Sensor SecuGen: **FTE_{SG}** = 0,00 %

- Sensor Veridicom: $\mathbf{FTE}_{SV}$ = 5,88 %
- Algorithm Bergdata: $\mathbf{FTE}_{AB}$ = 3,97 %
- Algorithm Siemens: $\mathbf{FTE}_{AS}$ = 3,98 %
- Algorithm Veridicom: $\mathbf{FTE}_{AV}$ = 0,72 %

These are only informative results indicating how many fingerprints have not been enrolled.

The next analyzed rate should be the Failure to Match (**FTM**) rate, but this rate is not so important in this case because all the fingerprints in the database which couldn't be enrolled couldn't be also matched. All other fingerprints could be matched and the situation can be better described by **FMR** / **FNMR** distributions and Receiver Operating Curves.

Receiver Operating Curve (**ROC**)

ROC is the graphical representation of the False Non-Match Rate (**FNMR**, eventually **FRR**) in relation to the False Match Rate (**FMR**, eventually **FAR**) – see Chapter 1.3. First the **FMR** and **FNMR** areas have to be computed. Genuine (comparisons of fingerprints from the same finger) and Impostor (comparisons of fingerprints from different fingers) distributions and corresponding **FMR** and **FNMR** rates can be seen in the Figures 5.4 to 5.7. The distributions for the Siemens algorithm and different sensors are shown in the Fig. 5.4. The Fig. 5.5 shows the final genuine and impostor distributions for the Siemens algorithm and all sensors. The distributions for the Veridicom algorithm and different sensors are shown in the Fig. 5.6. And finally, the Fig. 5.7 shows the final distributions for the Veridicom algorithm and all sensors. It should be noted that the *x*-axis scale is linear and the *y*-axis scale is logarithmic in all these figures. The reason is the density function – the peaks were too small to be seen in some areas. The *x*-axis displays the distribution of Matching Scores – the meaning is: 0 = no match (0% equality), 100 = total match (100% equality). And the values on the *y*-axis represent the occurrence density of the corresponding Matching Score.

The best Genuine distribution should have a full density at the Matching Score 100 and the best Impostor distribution should have a zero density at the Matching Score 0. Of course, such distributions are hardly possible because each fingerprint has "something" similar with any other fingerprint. Other factors influencing the results are: translation, rotation, pressure, dryness or wetness of fingerprint, etc. These factors affect the quality of fingerprints and when considering the influence of these factors, we can arrive at the conclusion that some similarity can be found in the fingerprints which do not belong to each other, or vice-versa – some fingerprints from the same finger can be recognized as from different fingers.

The following figures, namely Fig. 5.8 to Fig. 5.11, display the Receiver Operating Curves. The *x*-axis represents the values of False Match Rates (**FMR**) and the *y*-axis the corresponding False Non-Match Rates (**FNMR**) for all values of the threshold *T* within the interval from 0 to 100 on the Matching Score axis.

The better Receiver Operating Curve is, the closer it lies to the point of origin (0,0) and to the axes. The best possibility is no point of intersection of the Genuine and Impostor distributions and therefore the conclusion is that no real values of **FMR** and **FNMR** exist, or more precisely, such values are then equal to zero.

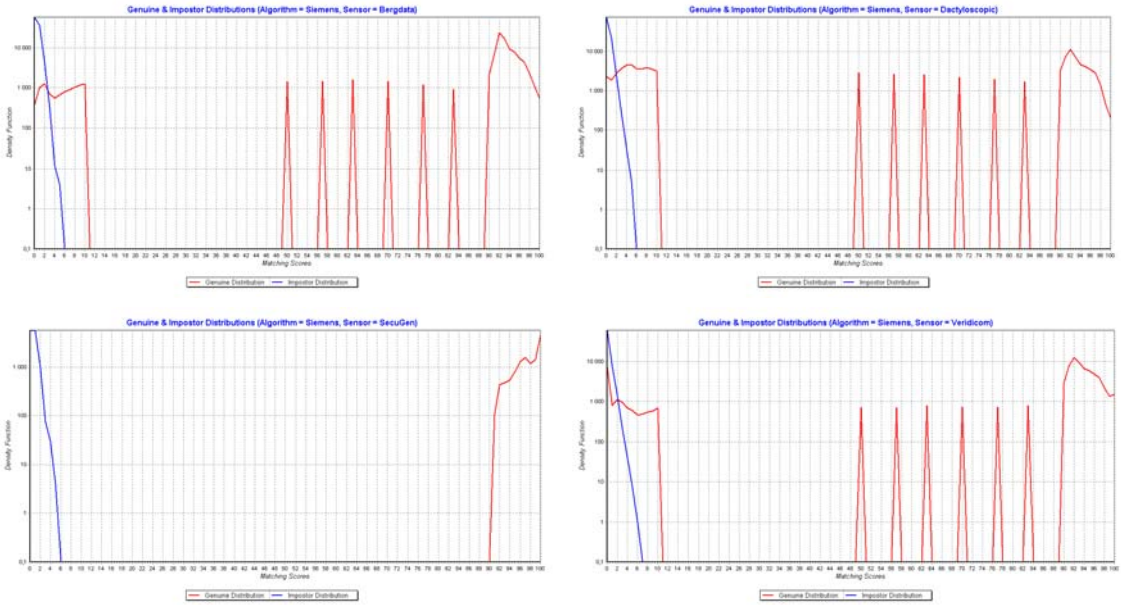


Fig. 5.4: Distributions for the Siemens algorithm and different sensors

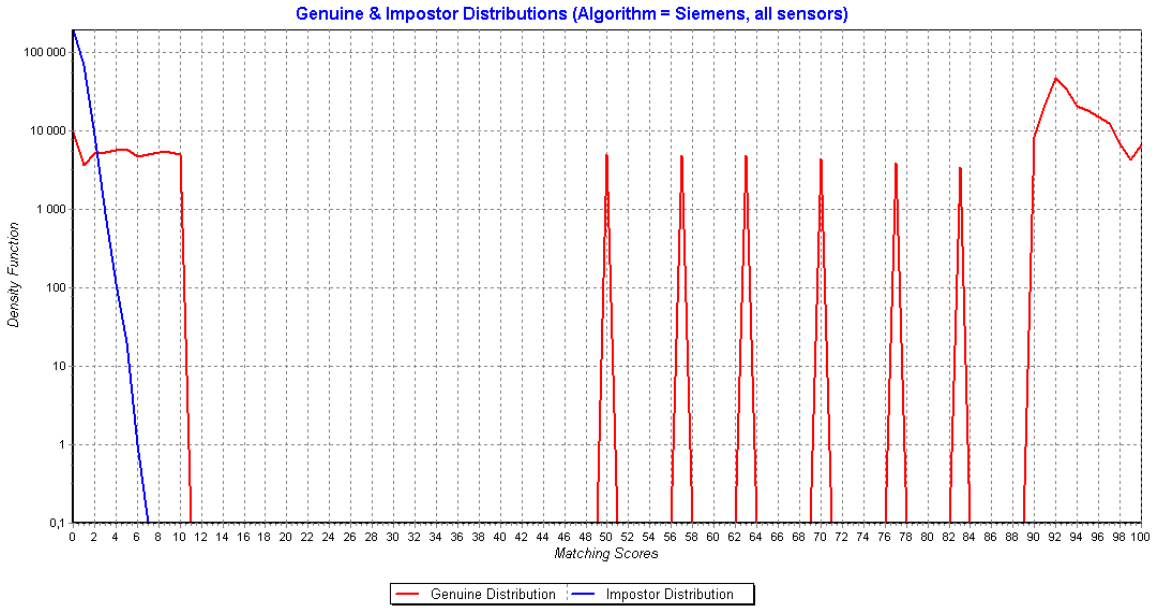


Fig. 5.5: Final Genuine and Impostor distributions for the Siemens algorithm

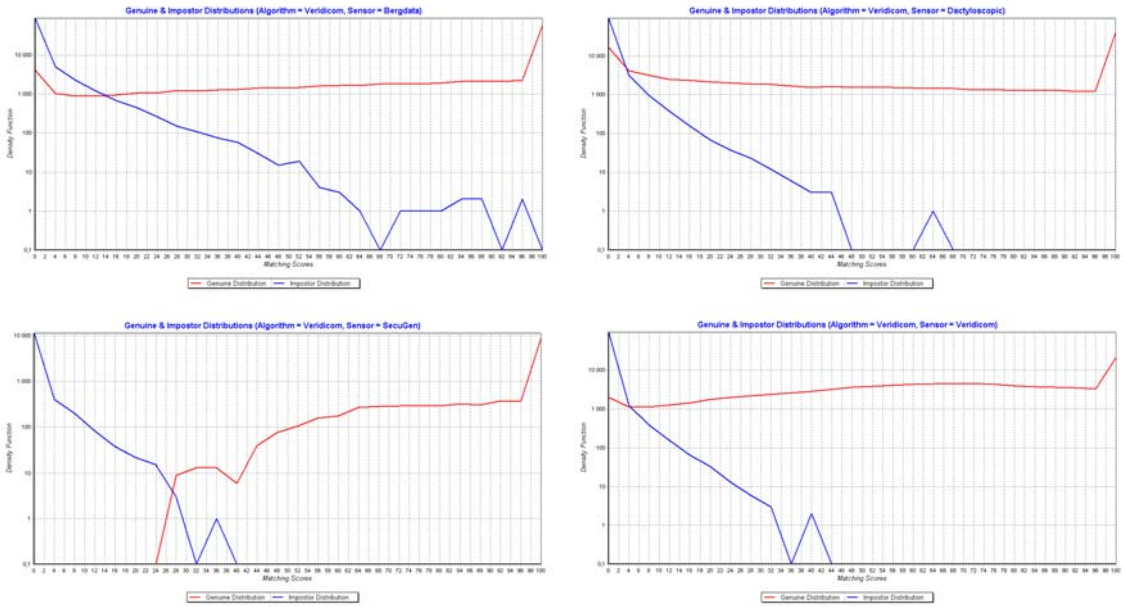


Fig. 5.6: Distributions for the Veridicom algorithm and different sensors

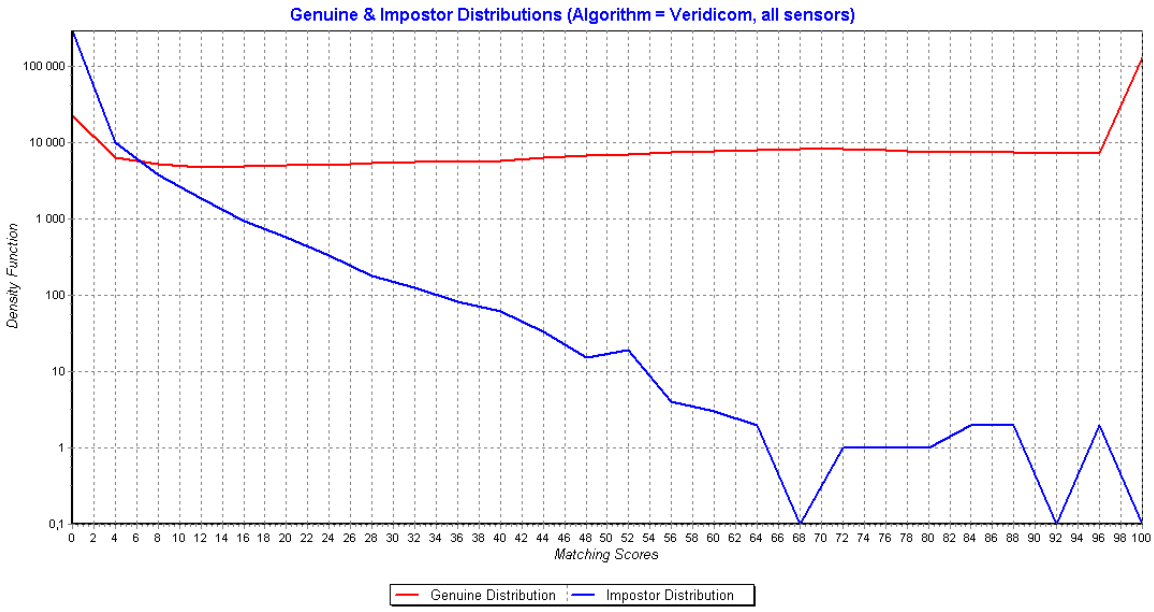


Fig. 5.7: Final Genuine and Impostor distributions for the Veridicom algorithm

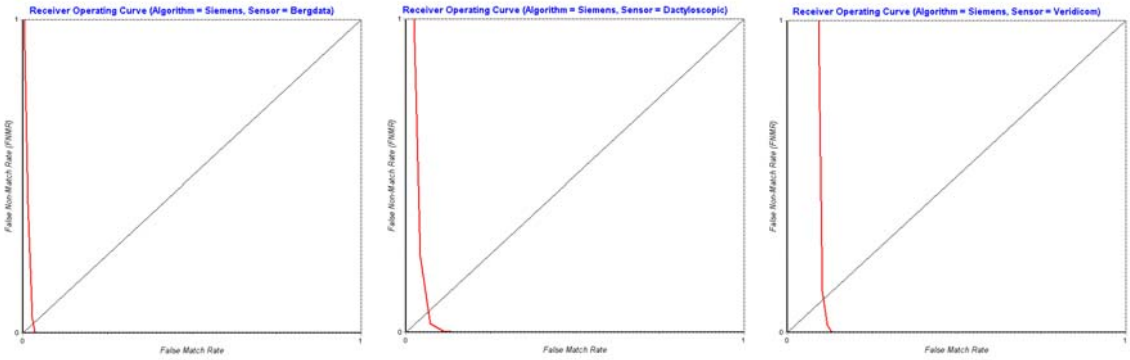


Fig. 5.8: **ROC** for the Siemens algorithm and different sensors

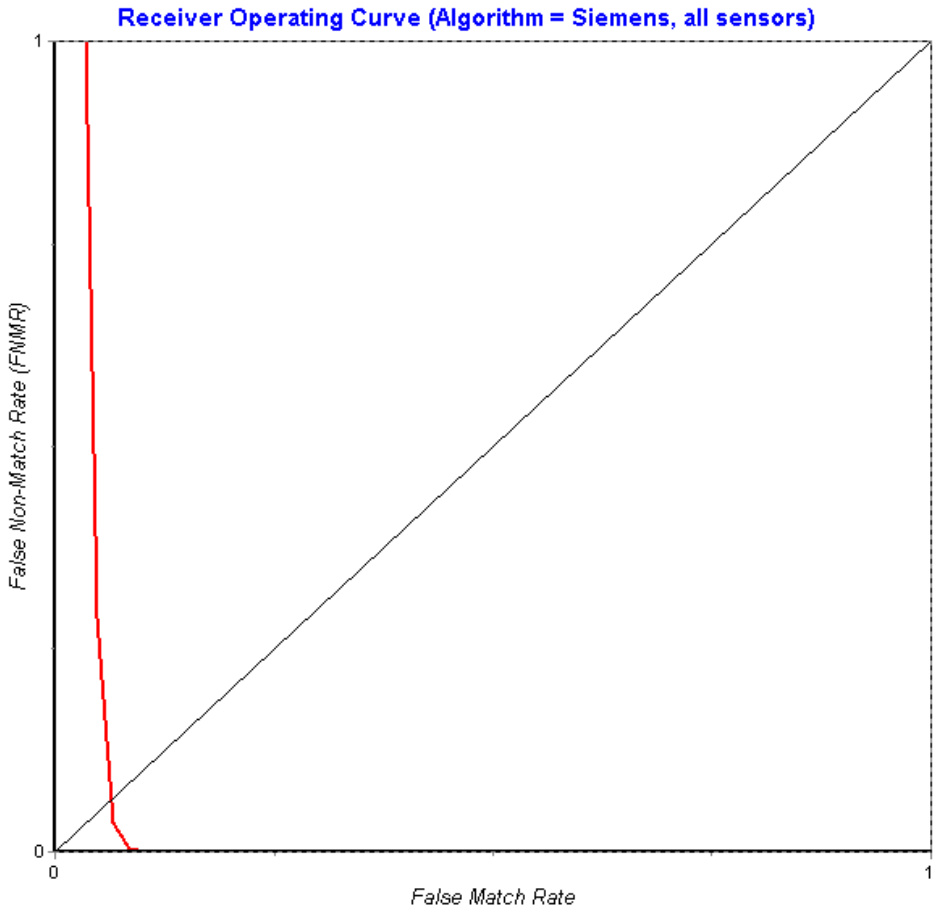


Fig. 5.9: Final **ROC** for the Siemens algorithm

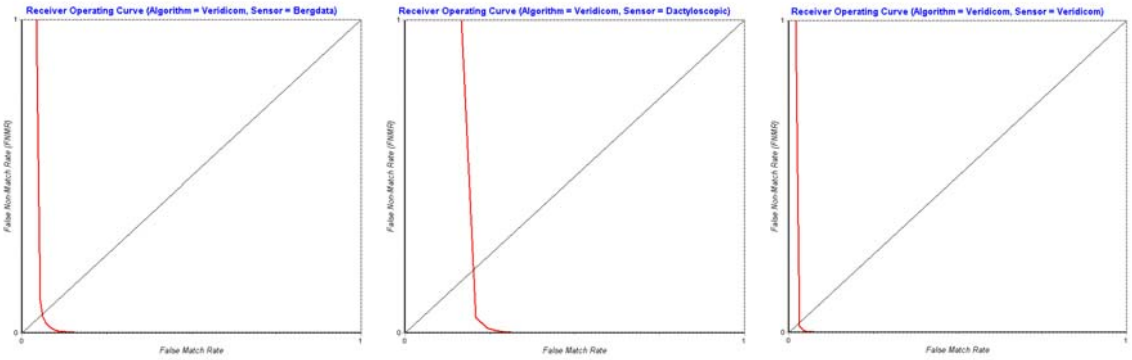


Fig. 5.10: **ROC** for the Veridicom algorithm and different sensors

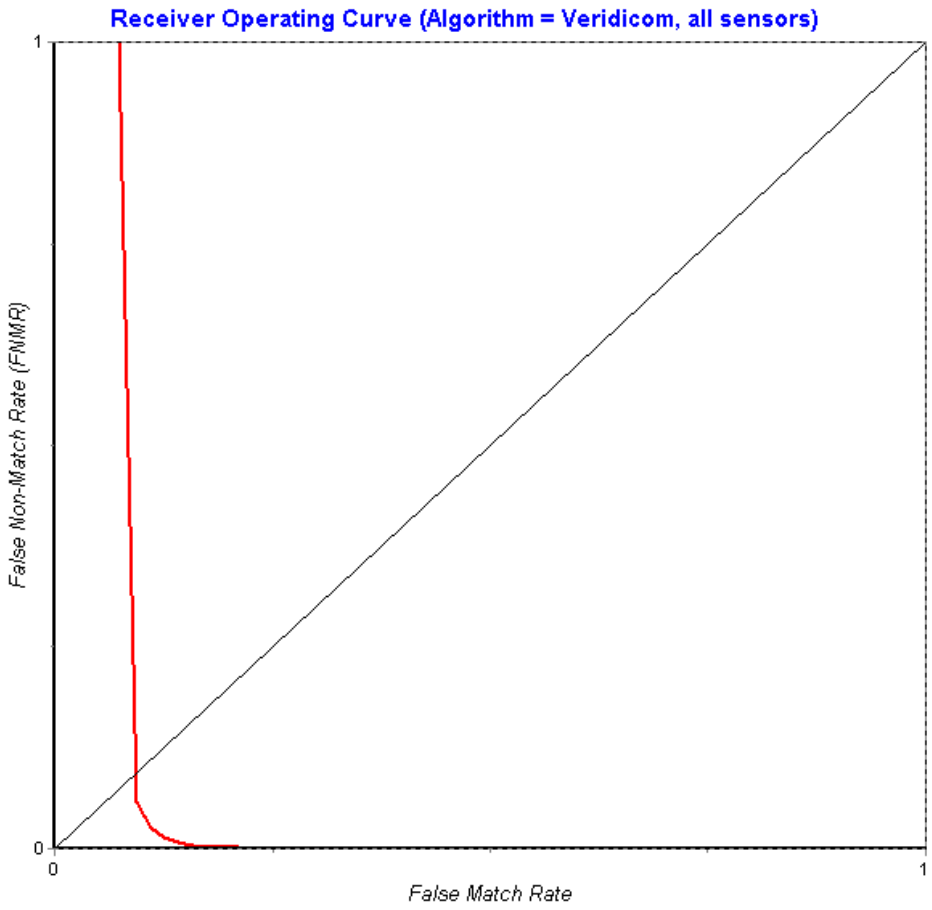


Fig. 5.11: Final **ROC** for the Veridicom algorithm

5.3 Key Generation

This chapter contains the results of testing of my own applications for the key generation from fingerprints. The algorithms have been described in the Chapter 4 and all the applications are presented in the file "*Applications.pdf*".

Computation of centers

There are three proposed methods for center computation. In the application for centers computation, the first method is based on the minutiae gravity center, the second one on the orientation field, and the last one is based on the ridge count maximums in horizontal and vertical directions (description of all three methods can be found in the Chapter 4.2.1). A fingerprint in digital imaging is not stable and more then one concrete position need to be considered for all computing operations. When acquiring fingerprint images, their translation, rotation or even stretching (by Bergdata sensor) can occur, and therefore we need to set some reference points in the fingerprint, which would enable us to compute the stability of all three methods. All fingerprints were examined and their cores (if present) and some other reference points (often delta points, especially at dactyloscopic fingerprints) have been determined. These two point types have been used for the computation of stability of respective methods. One aspect seems to be very important – the proportional variation of the distance between the core and the reference point. Such variation of distance has been computed as the maximal distance between the core and the reference point, minus the average distance between the core and the reference point, divided by the maximum length in the image (image diagonal), multiplied by 100 (to express it in %). The results of computation of average proportional distances between the cores and reference points for all users and all sensors can be seen in the Fig. 5.12. The *x*-axis refers to the ID number of users and the *y*-axis refers to the proportional variation of the distance between the core and reference point (in %). In this case, the max. distance variation can be used, as both points were set carefully (to keep variation small). Therefore no Gaussian distribution and deletion of maximal or minimal values is necessary.

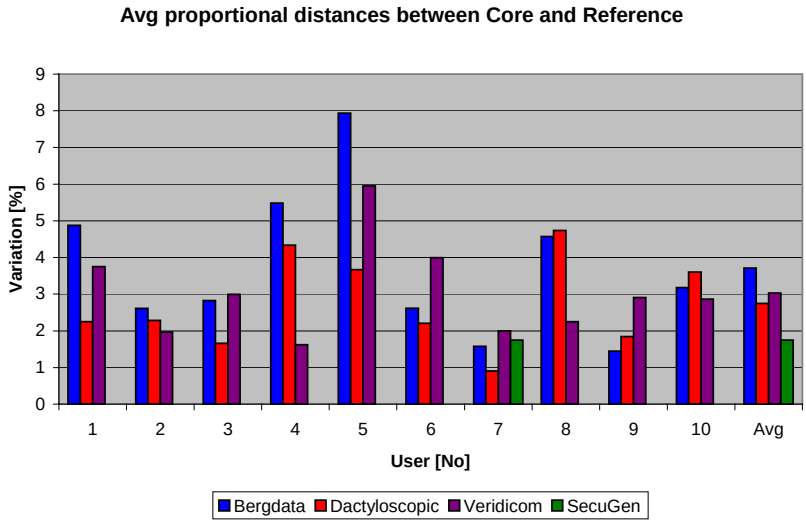


Fig. 5.12: Average proportional distances between Core and Reference

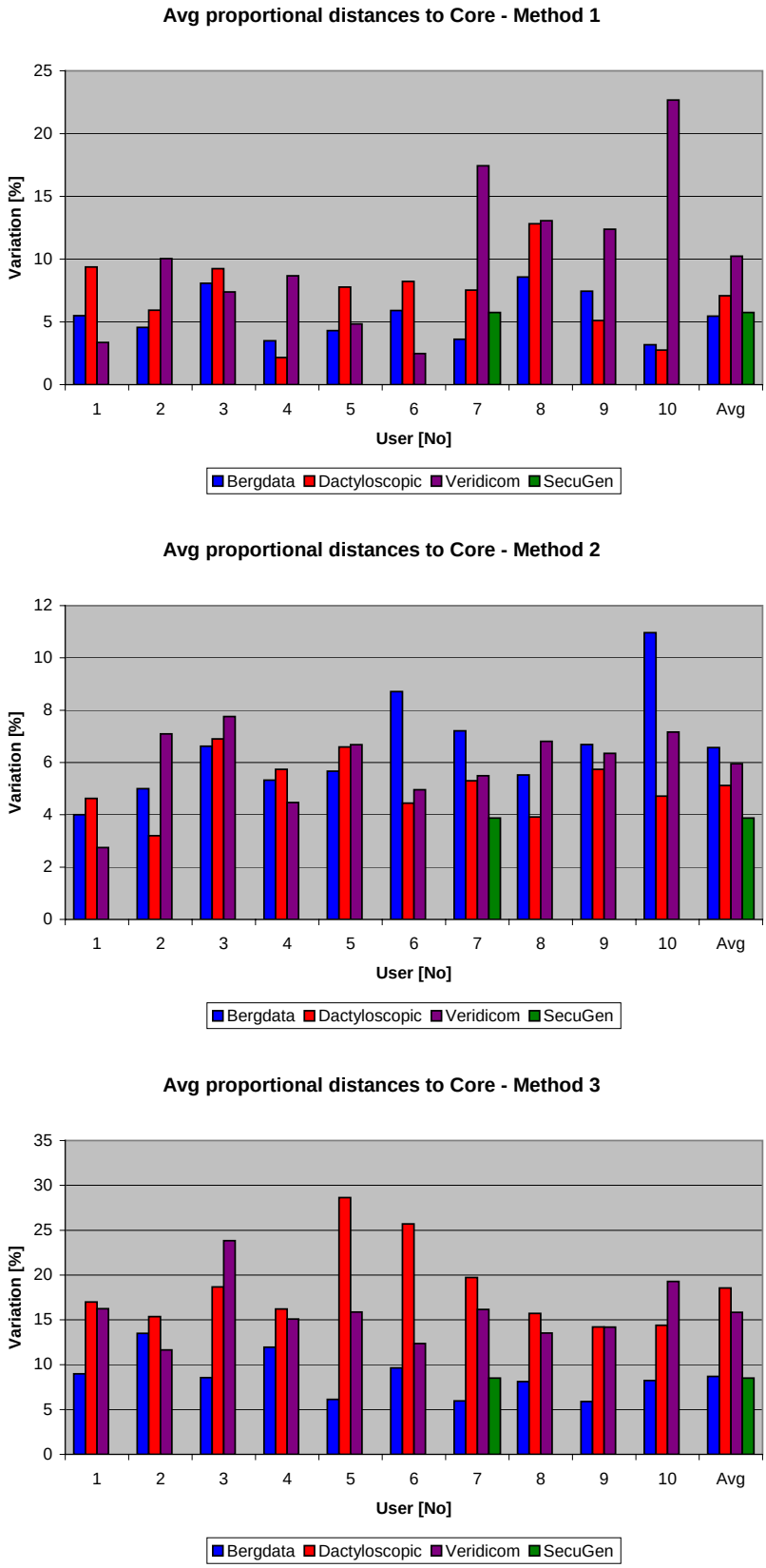


Fig. 5.13: Avg proportional distances between Core and corresponding centers

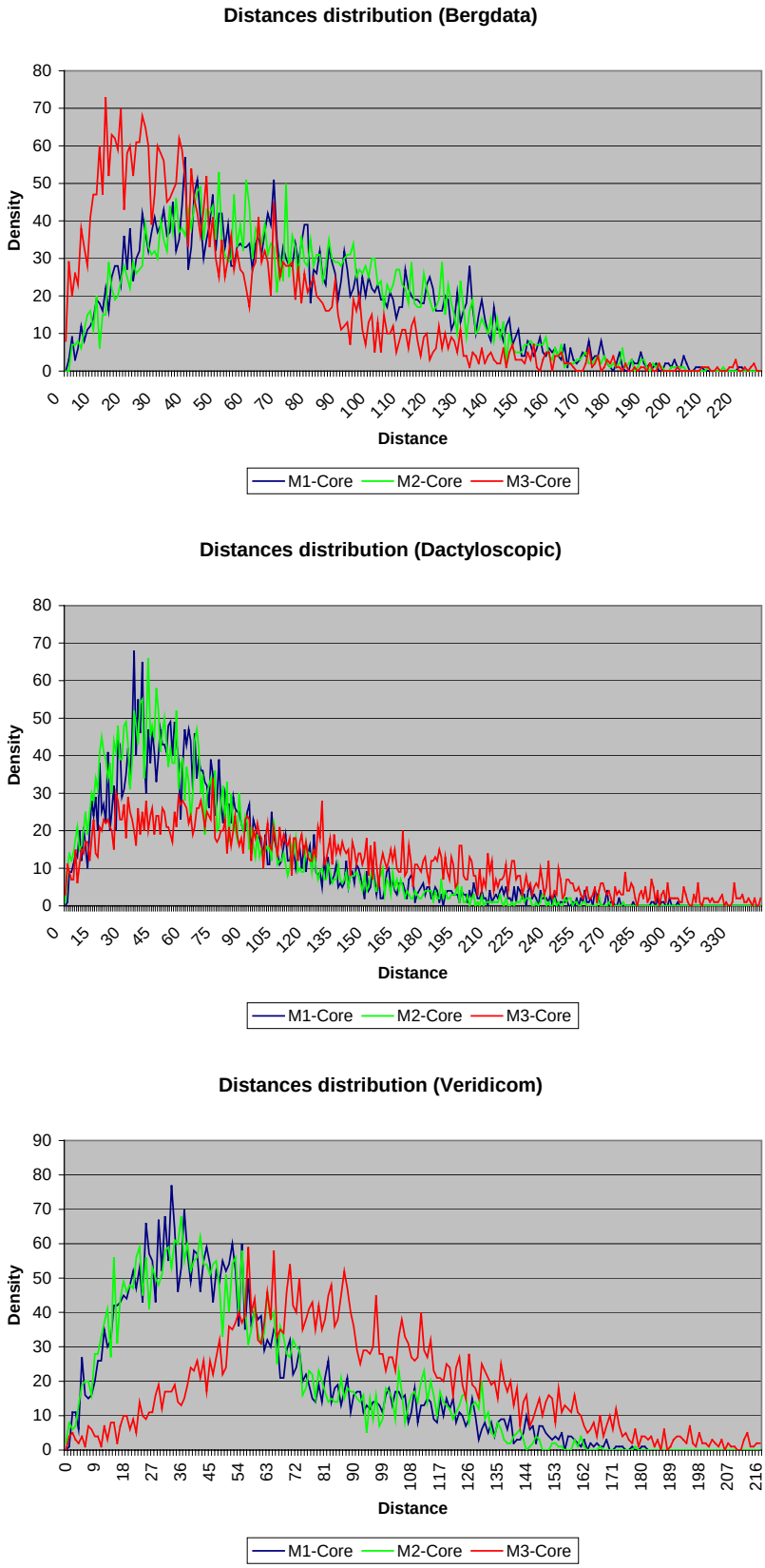


Fig. 5.14: Distributions of distances between Core and corresponding centers

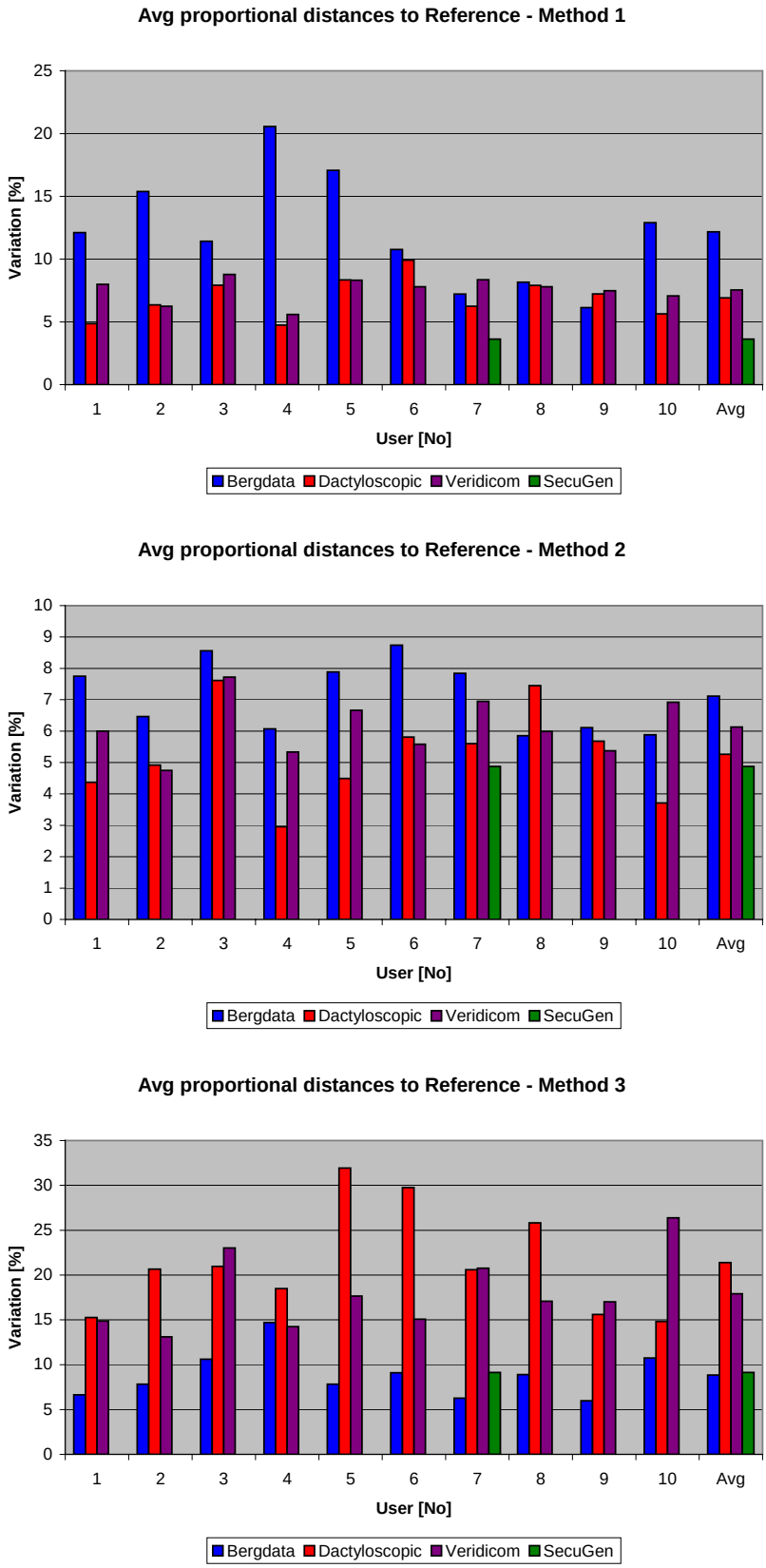


Fig. 5.15: Avg proportional distances between Reference and correspond. centers

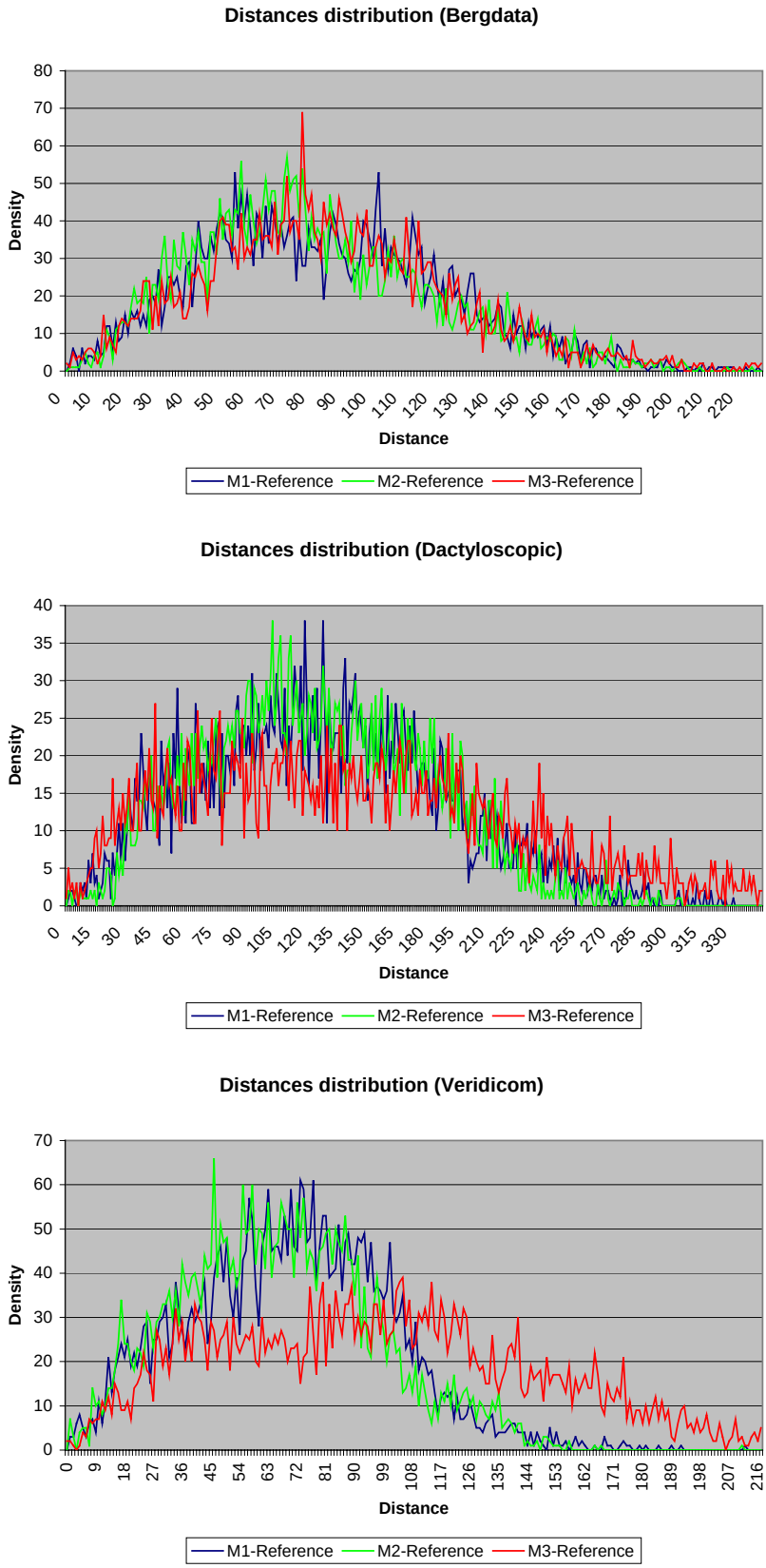


Fig. 5.16: Distributions of distances between Reference and correspond. centers

The average proportional distance variations between Core and corresponding center resulting of each method (see Tab. 5.3, Fig. 5.13 and Fig. 5.14), and between Reference and corresponding center resulting of each method (see Table 5.3, Fig. 5.15 and Fig. 5.16), have been computed for the comparison of stability of all three methods. The *x*-axis in Fig. 5.13 and Fig. 5.15 represents the user ID number and *y*-axis the distance variation (in %). The *x*-axis in Fig. 5.14 and Fig. 5.16 represents the distance (e.g. between Reference and the center of the Method_2) and the *y*-axis the density, i.e. how many times the corresponding distance (on the *x*-axis) has been found.

Tab. 5.3: Proportional distance variations for Core/Reference to the Center_M_x

Sensor / Center M_x	Core (data in [%])			Reference (data in [%])		
	Method_1	Method_2	Method_3	Method_1	Method_2	Method_3
Bergdata	5,47	6,57	8,69	12,17	7,12	8,86
Dactyloscopic	7,09	5,12	18,57	6,92	5,26	21,39
Veridicom	10,23	5,96	15,83	7,54	6,13	17,92
SecuGen	5,75	3,88	8,50	3,63	4,88	9,13
Average	**7,14**	**5,38**	**12,90**	**7,57**	**5,85**	**14,33**

The problem with proportional distance variations is that the positions of center points of corresponding methods can vary in a very small distance intervals, but at the same time, the center points can be scattered on all sides of the Core / Reference point (for better idea see Fig. 5.17). This figure shows some singular points (Δ = delta, ★ = core and ○ = centers of some method). The core is connected with the center points – the average distance in the left image is definitely greater than the average distance in the right image. But the scattering of the centers in the left image is smaller than in the right one. Hence, the circumference of area of occurrence of all centers can be taken as an additional metric for the quality of the set of centers. An alternative metric for such quality can be the area of the polygon with the center points as vertices.

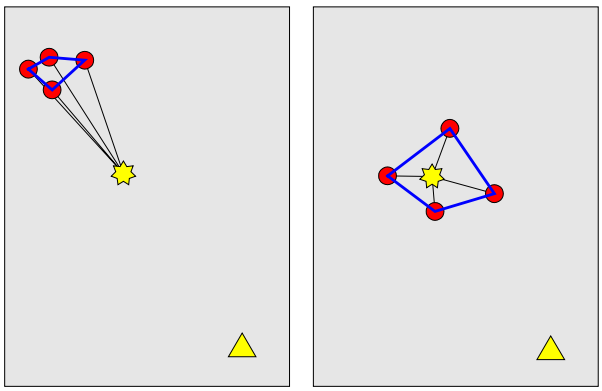
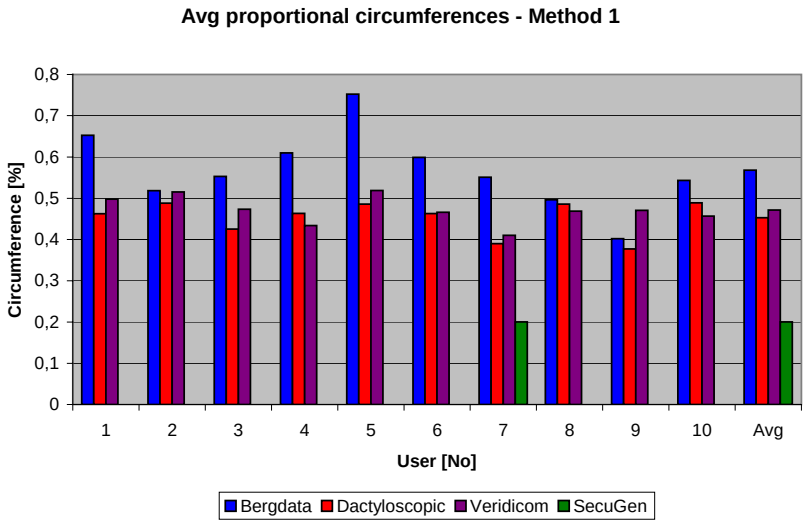
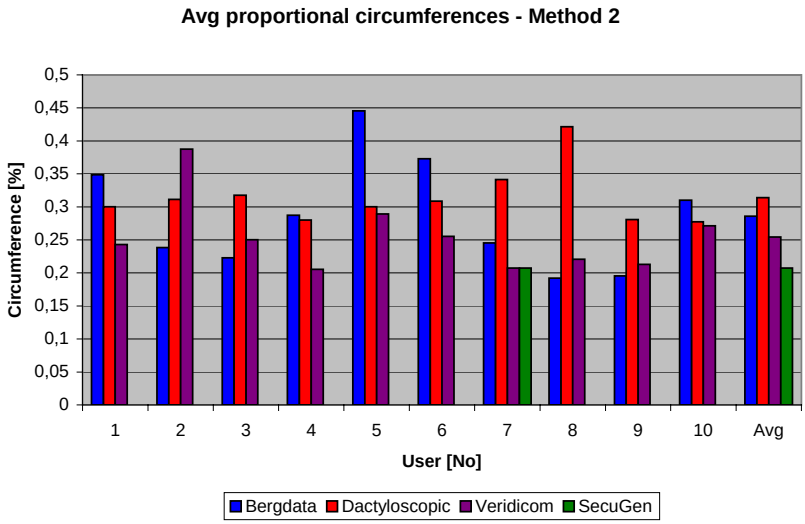


Fig. 5.17: Relation between distances and circumferences





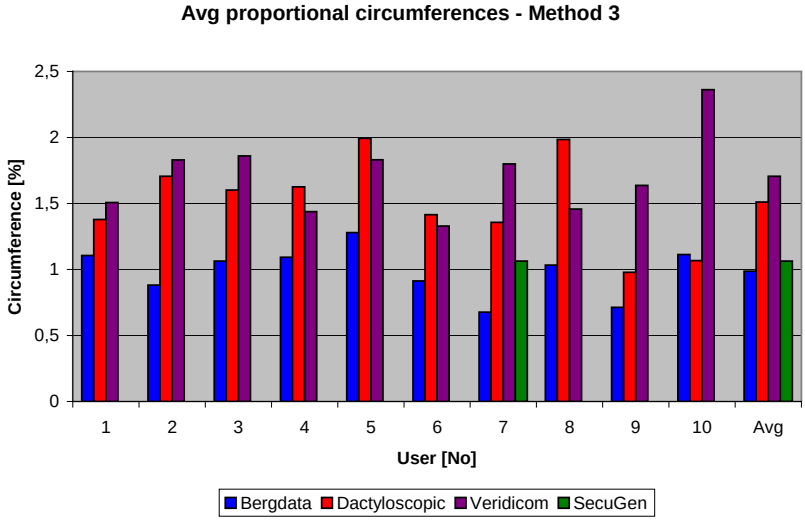


Fig. 5.18: Average proportional circumferences

Fig. 5.18 shows the average circumference variations. The data on the *y*-axis represent the percentage ratio related to maximal circumference in the corresponding image (image height and width is different for all three sensor images). Total 50 (N_N) maximal distances (≈ image diagonal – newly computed for each image) have been used as the reference for the maximal circumference. Tab. 5.4 contains the average circumferences for corresponding methods and sensors.

Tab. 5.4: Average circumferences for different methods and sensors

Sensor / Circumference	Method_1	Method_2	Method_3	Avg
Bergdata	0,57	0,29	0,99	**0,62**
Dactyloscopic	0,45	0,31	1,51	**0,76**
Veridicom	0,47	0,25	1,71	**0,81**
SecuGen	0,20	0,21	1,06	**0,49**
Average	**0,42**	**0,27**	**1,32**	-

Certain cells in Tab. 5.3 and Tab. 5.4 are highlighted – black background means the best method for corresponding sensor, the gray one represents the medium-quality method and the white background the worst one. After considering all winners and losers, the best method is Method_2 (Orientation Field), the second best method is Method_1 (Minutiae Gravity Center) and the last (worst) method is Method_3 (Ridge Count). Therefore the Method_2 has been used in the following computations. The centers determined by other methods are not considered – to save the disk capacity (data volume increases rapidly with the sub-combinations of the files). The stability of all three methods has been tested for all four sensors. It is necessary to note that the SecuGen sensor has been excluded from further considerations as the fingerprints only from one user of the SecuGen sensor are available. The results of SecuGen cannot be compared with those of other sensors because of very small amount of available data from this sensor. That is also the reason why the SecuGen results cannot be found in the following sections.

Reduction of minutiae amount

The amount of minutiae in fingerprints can vary very much. Fig. 5.19 shows the maximal, minimal and average amounts of minutiae for three sensors. The *x*-axis includes the ID numbers of corresponding users and the *y*-axis includes the scale for amount of minutiae. The dactyloscopic fingerprints have greater amount of minutiae due to their greater area. Tab. 5.5 contains the maximum, minimum, average and average percentage values relating to the number of refused minutiae files for all three sensors. Fig. 5.20 displays graphs showing the numbers of refused fingerprint files and indicates also the thresholds for minutiae amounts on the graph's right side (or legend). The following values were defined as thresholds: 10, 15, 20, 25, 30 and 35. The *x*-axis includes the ID numbers of corresponding users, the *y*-axis includes the scale for percentage amount of refused files (related to the maximal number of minutiae files for respective user) and the *z*-axis is the row for one of five thresholds.

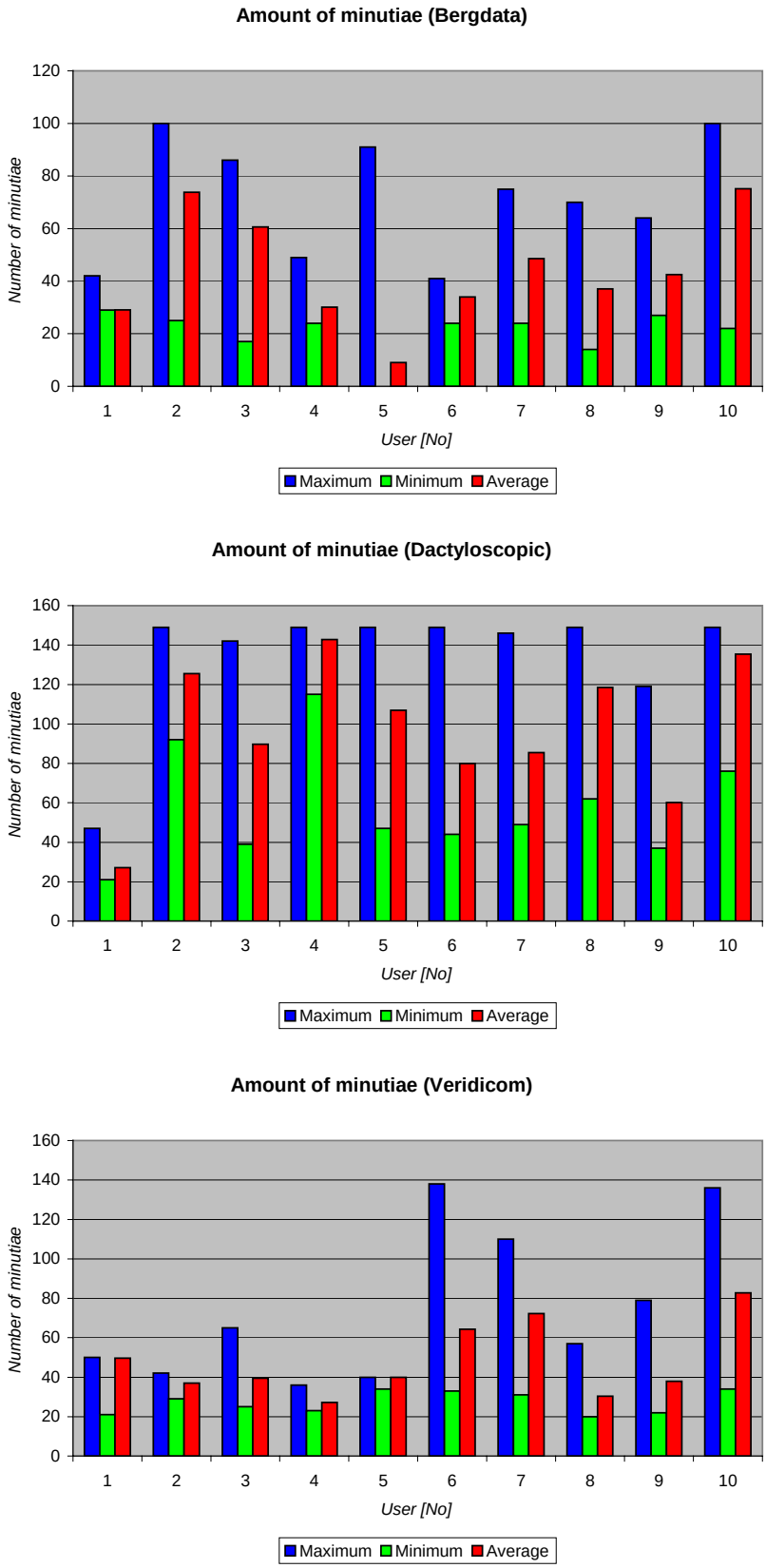
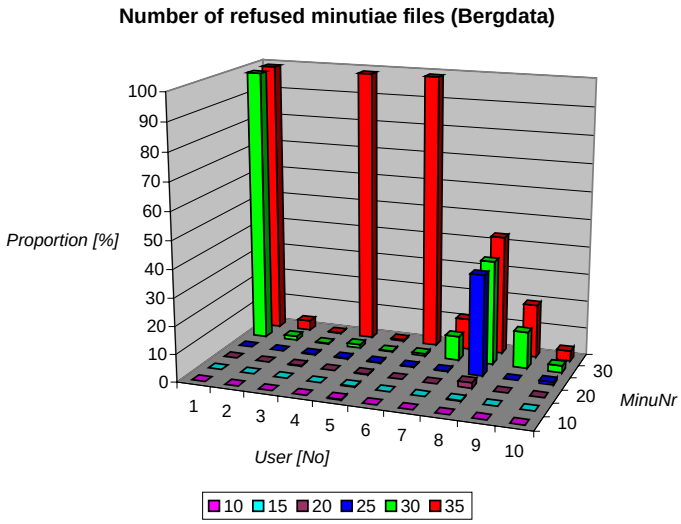
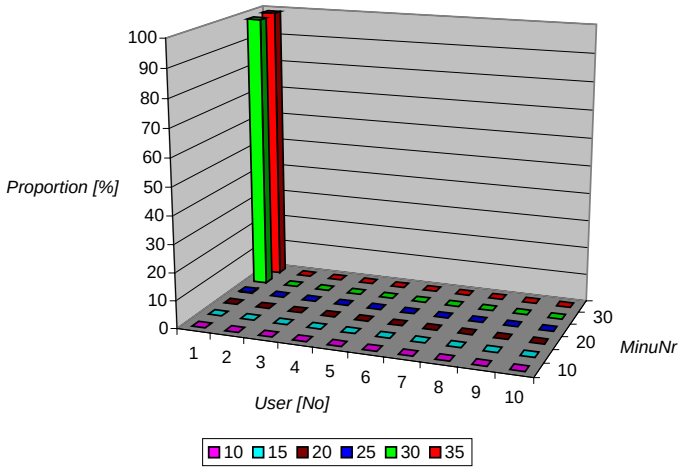


Fig. 5.19: Amount of minutiae for different sensors





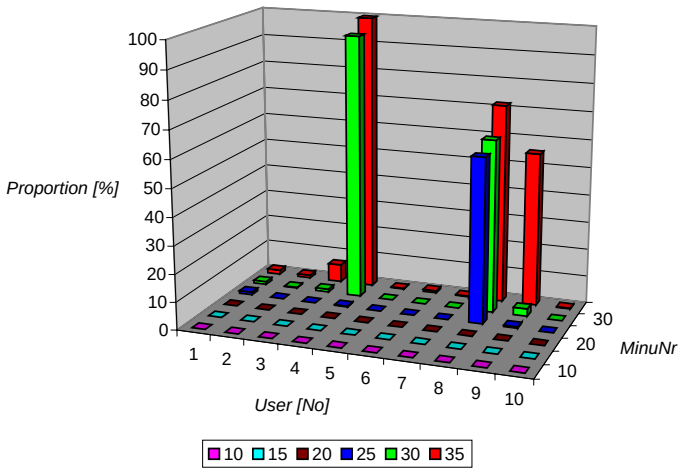


Fig. 5.20: Number of refused minutiae files

Tab. 5.5: Minutiae amounts and percentage amounts of refused files

	Sensors		
	Bergdata	Dactyloscopic	Veridicom
Minutiae – Maximum	71,80	134,80	75,30
Minutiae – Average	44,01	58,20	48,09
Minutiae – Minimum	20,60	97,14	27,20
Threshold = 10 [%]	0,025	0,000	0,000
Threshold = 15 [%]	0,050	0,000	0,000
Threshold = 20 [%]	0,250	0,000	0,000
Threshold = 25 [%]	3,850	0,025	6,100
Threshold = 30 [%]	16,450	9,850	16,250
Threshold = 35 [%]	37,775	9,900	23,750

When we analyze the results, it's clear that the thresholds of 35 and 30 are not suitable – there are too many refused files. The threshold of 10 can be rejected immediately, as the number 10 is lower than the acceptable lowest limit of the number of minutiae (see the guideline in the Chapter 3.3.4). When considering remaining three thresholds, it can be stated that the number of refused files for the threshold of 25 is still too high. At the end – only the thresholds of 15 and 20 are significant. It means that only the files with the thresholds of minutiae amounts 15 and 20 are used for the following computations.

Reordering, translation and rotation

The next sub-step is reordering of minutiae. For the reordering of minutiae, the center of the Method_2 (winner) is used. The distances to this center have been computed for each minutia (from maximum of 15 / 20). The closest minutia to the center of the Method_2 has been selected as the Reference minutia and the following minutiae are reordered depending on the distance (relative criterion is smaller or equal). At the end of this sub-step, all the minutiae are reordered.

The next sub-step is the translation. The Reference minutia is translated to the origin of the coordinate system and other minutiae are translated in the same direction (the *x*- and *y*-coordinates of the corresponding minutia are adjusted with regard to the translation of the Reference minutia). At the end of this sub-step, all the minutiae are translated.

The last sub-step is the rotation. For the rotation, the angle between the Reference minutia and the next minutia (after reordering) is computed. The average rotation angle values are shown in Fig. 5.21. For each fingerprint, the rotation angle values are indeed different and that is the reason why we cannot analyze such angle values or make any conclusions of them. The *x*-axis (Fig. 5.21) includes the ID numbers of corresponding users and respective average values (AVG), the *y*-axis includes the scale for average angle values for corresponding users and sensors. The rows on the *z*-axis are all three sensors. The average rotation angle for

Bergdata is 177,11°; for dactyloscopic fingerprints 196,40° and for Veridicom 114,99°.

As no conclusion with regard to the rotation angle could be made, both (rotated and non-rotated) minutiae files are used in the following sub-sections.

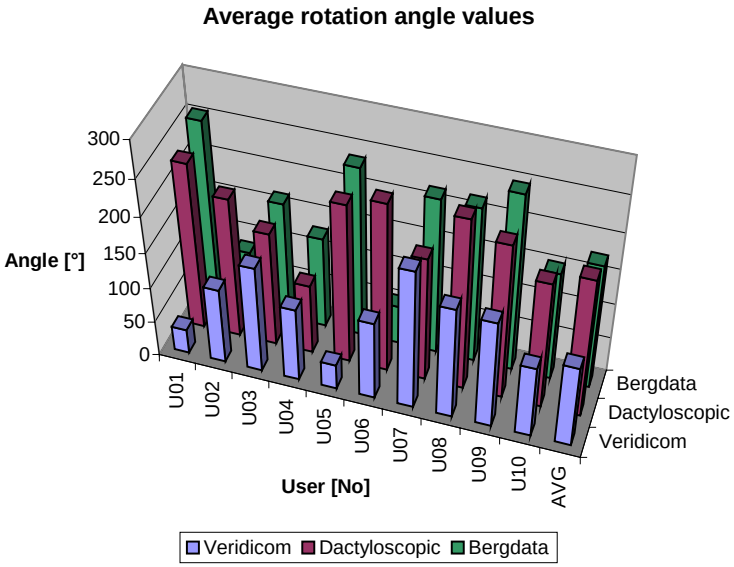


Fig. 5.21: Average rotation angle values

Quantization

The quantization is the following step. This step includes putting of some art of rough mask on the image, with the beginning in the Reference minutiae (position [0,0]) – see Fig. 4.9 and Chapter 4.2.2 - Quantization (formation of rough mask). The minutiae (15 or 20, rotated or non-rotated) are placed into respective rough matrix cells. The matrix cells have been defined as rectangle cells with the dimensions of *Factor* × σ_S. The following four values were used as factors: 3, 5, 10 and 20. If a cell already contains a minutia and some other minutia is to be placed into the same cell, a collision is announced. The numbers of collisions for different factors and sensors can be found in Fig. 5.22. The *x*-axis denotation means: U*m* – user ID number *m* (01 - 10), C*nn* – number of minutiae is *nn* (15/20), N/R – non-rotated/rotated. The average percentages of collisions for different factors and sensors are indicated in Tab. 5.6 (rotated and non-rotated files are considered as the same – the rotation changes only the position of minutiae but it does not change the distances among them).

When we analyze the values in the Tab. 5.6, we can see that the factors 10 and 20 are too high and therefore not suitable for the quantization. This result had been awaited due to the computations discussed in the Chapter 3.3.5. On the basis of these results, only the files with factors 3 and 5 have been used for further considerations.

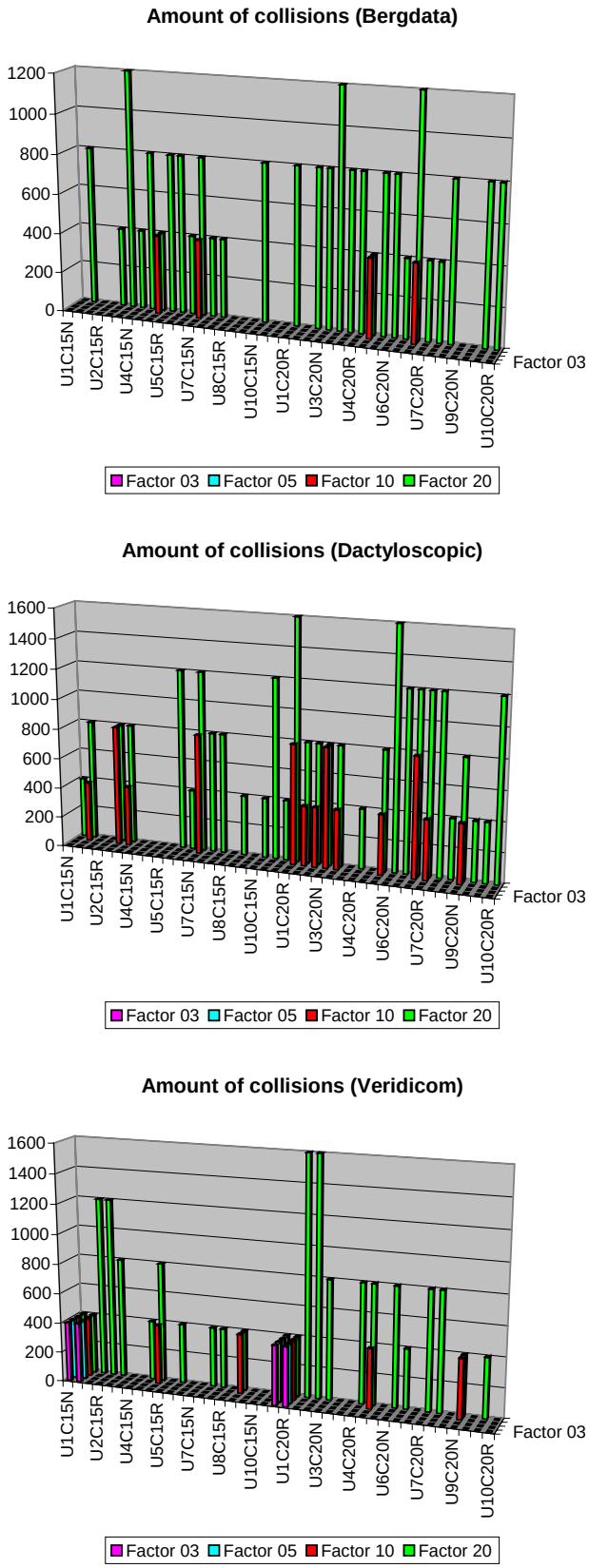


Fig. 5.22: Amount of collisions for different sensors

Tab. 5.6: Collisions for different quantization factors

Sensor	MinuNr	Factor = 3	Factor = 5	Factor = 10	Factor = 20
Bergdata	15	0,00 %	0,00 %	0,67 %	7,00 %
	20	0,00 %	0,00 %	0,50 %	7,50 %
Dactyloscopic	15	0,00 %	0,00 %	2,00 %	7,67 %
	20	0,00 %	0,00 %	3,00 %	10,00 %
Veridicom	15	0,67 %	0,67 %	1,33 %	5,67 %
	20	0,67 %	0,67 %	1,00 %	6,25 %

Computation of sub-graphs

After the quantization step, all non-repeating, ordered combinations of sub-vectors (sub-graphs) are computed. We have two file types – one with 15 and the second with 20 minutiae pro file. In consequence, the sub-vectors with lengths 12 and 17, respectively, have been considered. It means that the numbers of possible combinations of files with 15 and 20 minutiae, respectively, can be expressed as follows:

$$\binom{15}{12} = \frac{15!}{12! \cdot 3!} = 455 \text{ and } \binom{20}{17} = \frac{20!}{17! \cdot 3!} = 1140 \qquad (5.6)$$

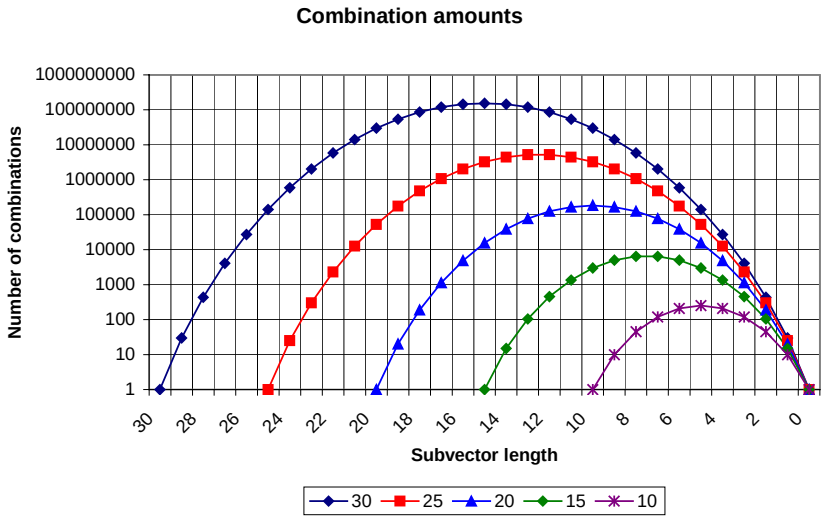


Fig. 5.23: Increasing amount of data for different sub-vector lengths

For each file all (455/1140) sub-vector combinations, i.e. 12 from 15 and 17 from 20, were generated. The reason for the numbers of 12 and 17 was the data amount. The corresponding increase of data volume can be seen in Fig. 5.23. The *x*-axis denotes the length of the sub-vectors; the *y*-axis the number of all possible combinations for corresponding maximal number of minutiae (curve) and sub-vector length. The following Fig. 5.24 shows the data volumes for different maximal numbers of minutiae (curves) and corresponding sub-vector lengths. It is clear

from this figure that data volumes increase very rapidly and e.g. for the amount of minutiae equal to 30 and length of sub-vector equal to 15, the corresponding database volume would be 279TB!! When we compute the database volumes for our values, then for the maximum amount of minutiae equal to 20 and length of sub-vector equal to 17 we obtain the database volume 2,3GB and for the maximum amount of minutiae equal to 15 and length of sub-vector equal to 12 we obtain the database volume 655MB. It needs to be mentioned that for each sub-vector length further combinations exist (rotated / non-rotated and factor 3 / 5), and therefore the whole database volume after the computation of sub-vectors reaches around 9,3GB + 2,6GB = 11,9GB.

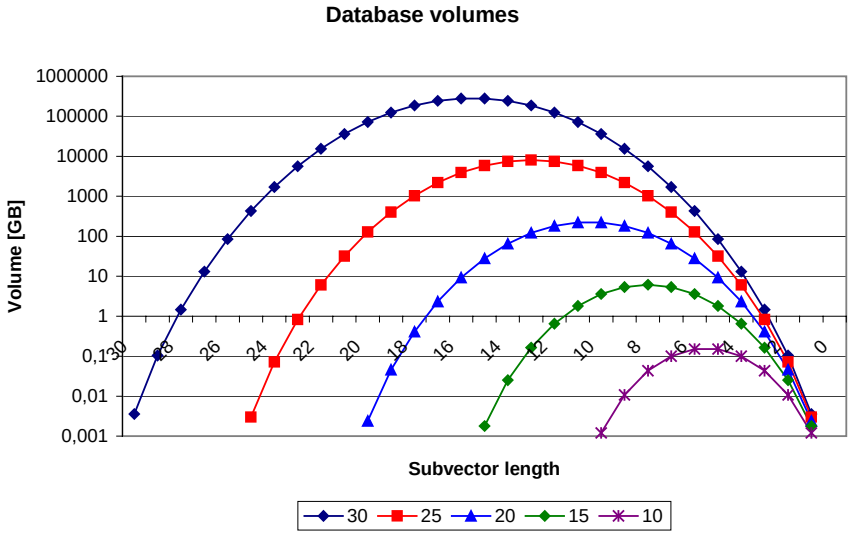


Fig. 5.24: Increasing data volume (in gigabytes) for different sub-vector lengths

Sub-graphs comparison

The whole database with the volume 11,9GB needs to be compared. The problem does not consist in the disc operations but in the CPU utilization. The process of comparison has been divided into two steps (estimation of genuine and impostor distribution) and furthermore it has been realized on 5 computers (PC_1: AMD Sempron™ 2200+, RAM 490MB, Windows 2000 (SP4); PC_2: Intel Pentium® III 700MHz, RAM 196MB, Windows 2000 (SP4); PC_3: AMD Athlon™ XP2000+, RAM 262MB, Windows 2000 (SP4); PC_4: Intel Pentium® III 700MHz, RAM 64MB, Windows 98 (SP2); PC_5: Intel Pentium® 4 – 3,20GHz, RAM 992MB, Windows XP Professional (SP2)). The computing times for matching (genuine and impostor distributions) at different computers can be found in Tab. 5.7.

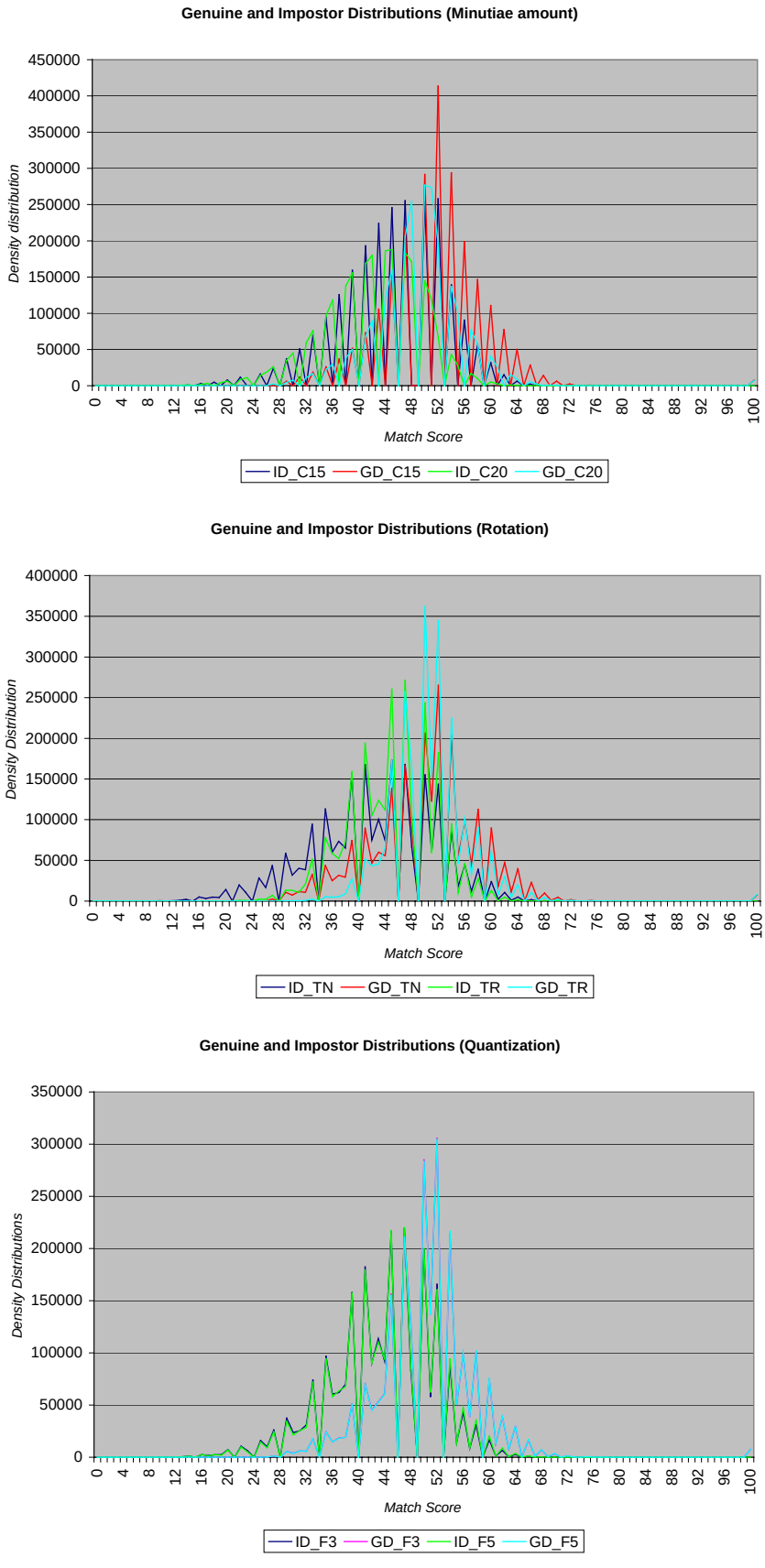


Fig. 5.25: Genuine and Impostor Distributions for different settings

After several weeks of comparising computations, we have come to the following final results. These results are presented in the following figures, tables and text discussions. The Fig. 5.25 presents Genuine and Impostor distributions for different settings. The graph on the top shows the distributions for the amounts of minutiae 15 and 20 pro file. The graph in the middle shows the influence of rotation to the distributions. The last graph in Fig. 5.25 shows the influence of quantization, using two different factors (3 and 5).

Tab. 5.7: Time consumptions for matching (genuine and impostor distributions)

[MinuNr]	PC_1 [min]	PC_2 [min]	PC_3 [min]	PC_4 [min]	PC_5 [min]
GD [15]	3,53	6,97	4,55	42,05	-
ID [15]	4,94	-	6,03	-	1,97
GD [20]	34,13	135,42	33,68	398,14	-
ID [20]	48,38	-	52,40	-	15,88

The following Fig. 5.26 presents Receiver Operating Curves (**ROC**s) for different sensors and different settings. The legend items in each graph have the following meaning: e.g. B15TN3 $\Rightarrow$ B = Bergdata sensor (D = dactyloscopic, V = Veridicom), amount of minutiae equal to 15 (or 20), TN = non-rotated (or TR rotated) minutiae files and quantization factor is 3 (or 5). The top image in the **ROC** graphs corresponds to the Bergdata sensor, the graph in the middle to dactyloscopic fingerprints and the graph on the bottom to the Veridicom sensor. In Fig. 5.27, we can see the comparison of best candidates from all three **ROC** graphs (from Fig. 5.26). The best curve (red) was obtained for the following settings: Bergdata sensor, 20 minutiae, rotated and quantization factor equal to 5. The other two curves are very similar; the same settings are: amount of 20 minutiae, two times rotation and two times quantization factor 5.

The Fig. 5.28 presents the Genuine and Impostor distributions for all three sensors. There are very small peaks at the end of the *x*-axis (by the match score equal to 100), and this phenomenon is further discussed in Tab. 5.8 and in the following text. In Fig. 5.28, we can see that the best Impostor Distribution corresponds to the Bergdata sensor and the best Genuine Distribution to the Veridicom sensor. The results of dactyloscopic finger images lie somewhere in between. The Fig. 5.29 presents the average Match scores of Genuine Distributions for different settings (with the same meaning of values as in Fig. 5.26) and three sensors. When we analyze this figure, the highest average match score corresponds to the Bergdata sensor and the second best is the Veridicom sensor. Dactyloscopic fingerprints provided the worst results.

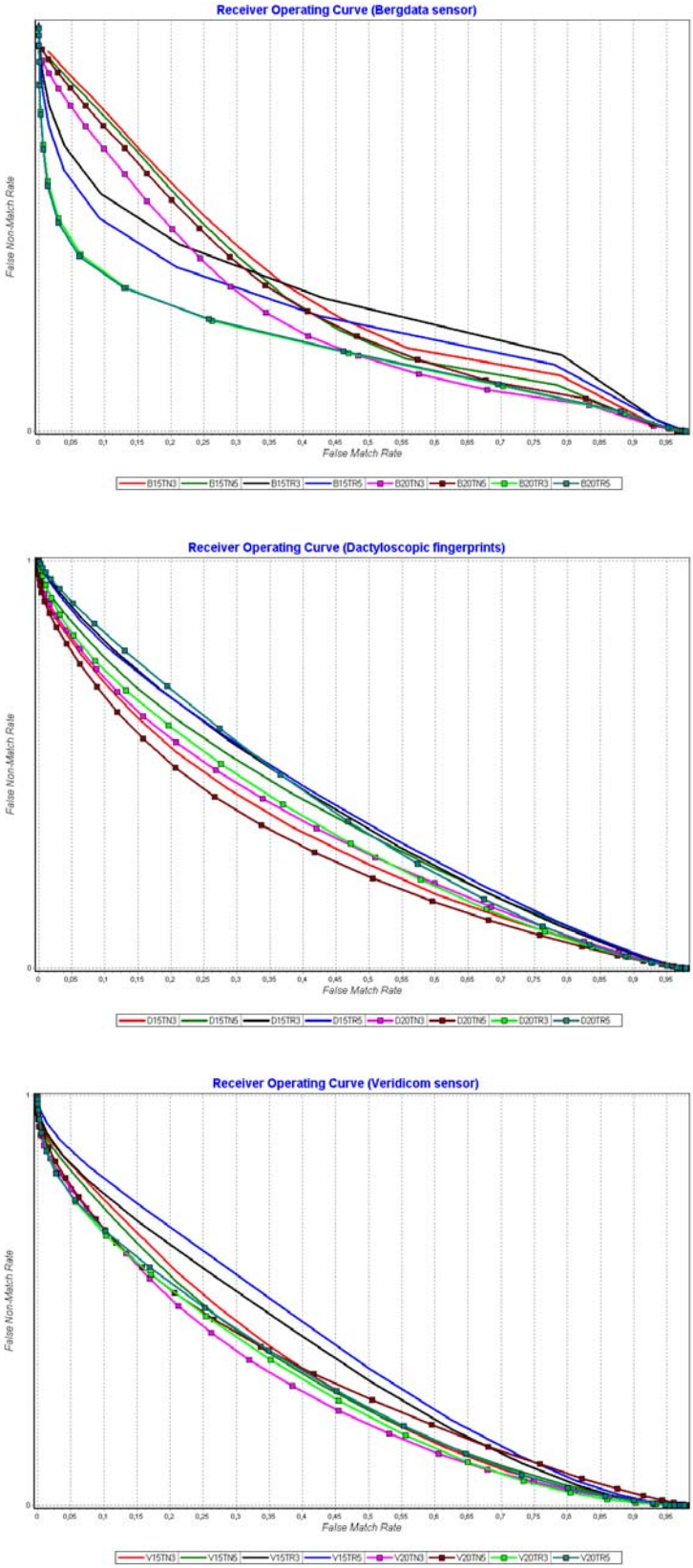


Fig. 5.26: Receiver Operating Curves for different settings and sensors

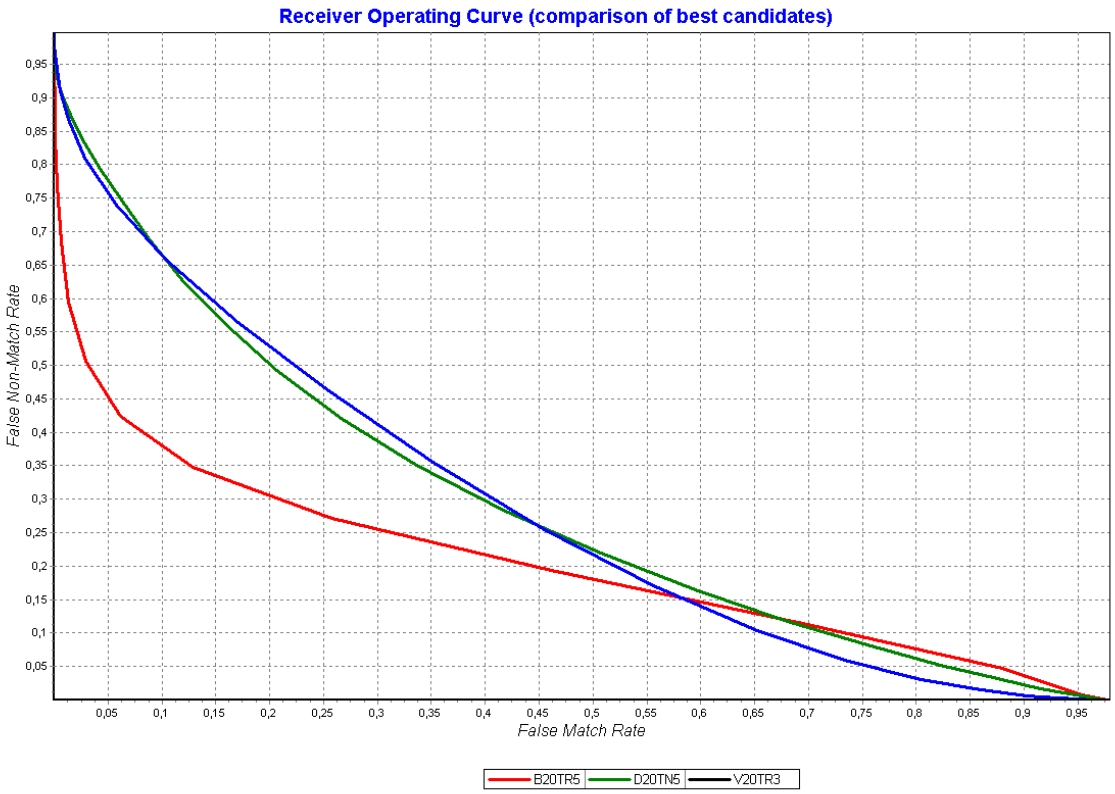


Fig. 5.27: Receiver Operating Curves for best candidates

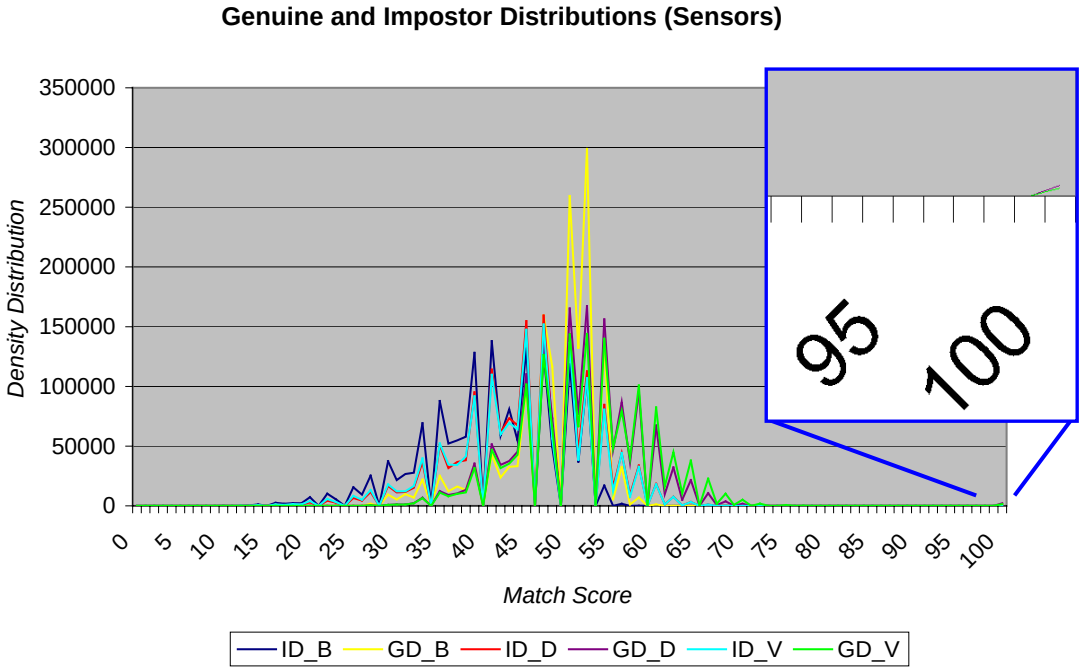


Fig. 5.28: Genuine and Impostor Distributions for all sensor types

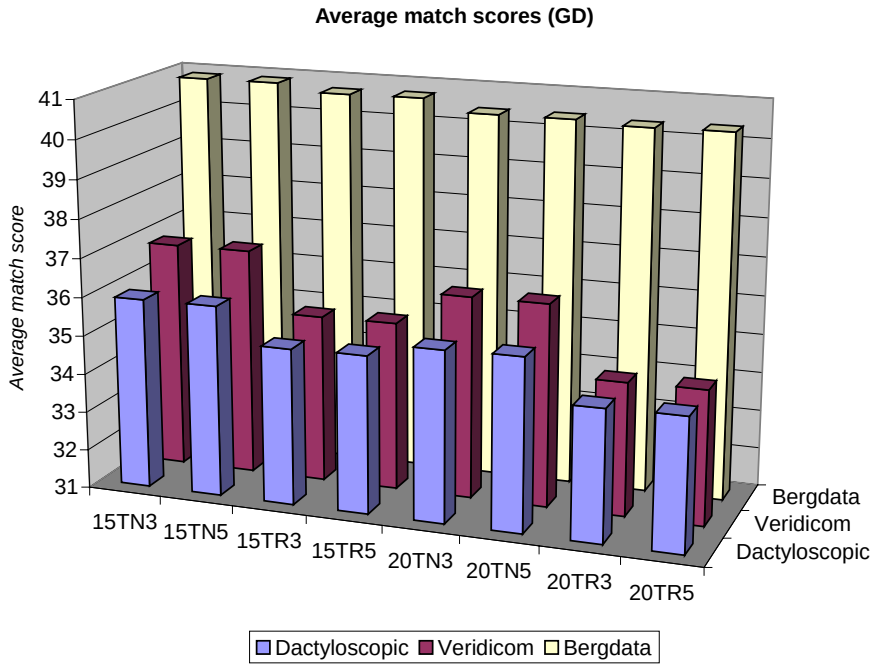


Fig. 5.29: Average match scores (GD) for different factor settings

The Tab. 5.8 presents the values of Match scores for maximal peaks of Genuine and Impostor distributions for different settings and further the maximal amount of total matches (with 100% equality) for 50 fingerprints. Some table cells are highlighted – these are the winners for corresponding settings. In general, it is wanted that the Match Score with a maximal peak in the Genuine Distribution reaches a higher value than for the Impostor Distribution. When the maximal amount of total (100%) matches is reached, then the value 0% is wanted for the Impostor Distribution (this corresponds to very good results) and the value 100% is wanted for the Genuine Distribution. The higher are match values, the better is the corresponding setting. As the particular algorithms have not been completely optimized, the maximal amounts of total matches have not reached the 100% value, but nevertheless the results have shown us that the proportion of matches is quite high – the concept of the whole system has been proven.

The Fig. 5.30 shows amounts of total matches (for matching of 50×50 fingerprints), i.e. the percentage amount of sub-graph comparisons with 100% equality (this corresponds to the end of the *x*-axis scale for the Genuine and Impostor distributions). Total match means that the corresponding hash values are the same and a match will be obvious in this case. If the match is found, we can use this sub-graph as a biometric key. But on the other hand, the values in the Fig. 5.30 also mean that the user needs to provide his/her fingerprints for acquirement more times to get a match result. This numbers are obtained as the amounts of all comparisons among 50×50 fingerprints (see Fig. 5.28) per finger. The causes for this are discussed in the Chapter 5.4, i.e. not optimized algorithms. But the functionality of key generation from the fingerprint is demonstrated.

Tab. 5.8: Match scores in maximal peaks and maximal amount of total matches

-	Match score in max. peak (GD)	Match score in max. peak (ID)	Max. amount of total matches (GD) [%]	Max. amount of total matches (ID) [%]
C = 15	52	52	16,38	0,00
C = 20	50	45	15,46	0,00
Non-Rotated	52	45	15,07	0,00
Rotated (TR)	50	47	16,77	0,00
F = 3	52	47	15,92	0,00
F = 5	52	47	15,92	0,00
Bergdata	52	41	5,23	0,00
Dactyloscopic	52	47	6,54	0,00
Veridicom	52	47	4,74	0,00

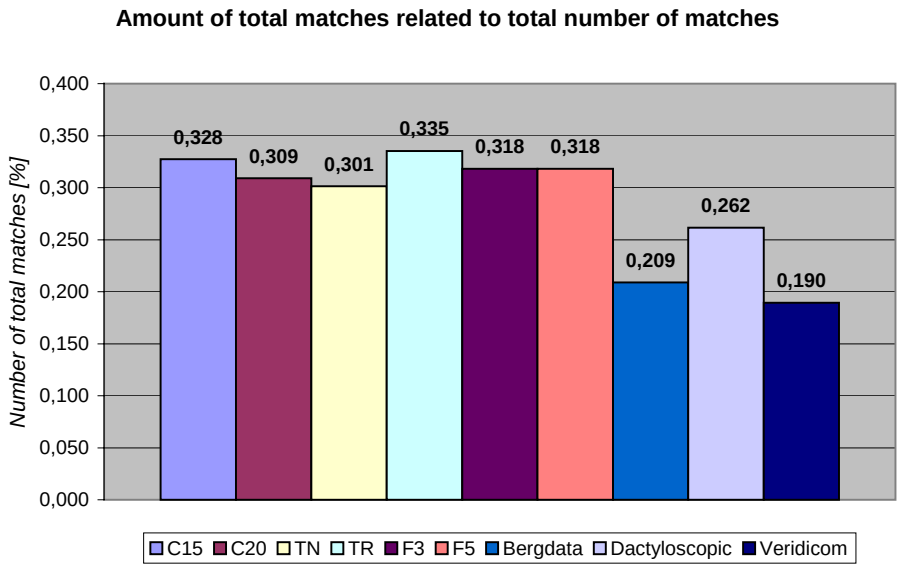


Fig. 5.30: Percentage amounts for total matches for different settings

Now, we want to decide, which settings have been the best ones. The decision can be made on the basis of the data from Tab. 5.9 where the particular winners of corresponding evaluation metrics are shown. The amount of 15 minutiae can be found 6 times and 20 minutiae totally 7 times $\Rightarrow$ the winner are 20 minutiae. Further, total 10 candidates can be found among the non-rotated fingerprints and 3 candidates among the rotated ones $\Rightarrow$ the winner are non-rotated fingerprints. Both quantization factors (3 and 5) have the same occurrence and it is therefore impossible to decide which quantization factor is better. When we evaluate the results for sensors and set to the first place 3 points, to the second one 2 points and

to the last one only 1 point, then the Bergdata sensor obtains 5 points, dactyloscopic fingerprints 4 points and Veridicom sensor only 2 points. The best settings are therefore characterized by the following set of data:

- Bergdata sensor, 20 minutiae pro file, non-rotated, factor 3 or 5.

Tab. 5.9: Winners of corresponding evaluation metrics

Evaluation metric	Winner settings
Genuine Distribution (Fig. 5.25)	▪ Minutiae amount C = 15 ▪ Non-rotated (**TN**) ▪ Factors are the same
Impostor Distribution (Fig. 5.25)	▪ Minutiae amount **C = 20** ▪ Non-rotated (**TN**) ▪ Factors are the same
ROC for Bergdata (Fig. 5.26)	1. **C = 20**, TR, F = 3 2. **C = 20**, TR, F = 5
ROC for dactyloscopic fingerprints (Fig. 5.26)	1. **C = 20**, **TN**, F = 5 2. C = 15, **TN**, F = 3
ROC for Veridicom (Fig. 5.26)	1. **C = 20**, **TN**, F = 3 2. **C = 20**, TR, F = 3
Average match scores (Fig. 5.29)	1. **B** → C = 15, **TN**, F = 5 2. V → C = 15, **TN**, F = 5 3. D → C = 15, **TN**, F = 5
Match scores in maximal peaks (Fig. 5.30 and Tab. 5.8)	▪ GD → C = 15, **TN**, F = $^3/_5$ ▪ ID → **C = 20**, **TN**, F = $^3/_5$ ▪ Sensors: D, **B**, V

5.4 Summary and Future Work

As we have learned, the biometric technologies are at the beginning of their broad practical application. There are many biometric attributes, such as fingerprint, face, retina, iris, voice, etc., which can be used for the verification systems (access systems). At the moment, it seems that the most required system based on such biometric verification method relates to personal document applications allowing a person to be verified or identified (in this case, its identity is unknown at the beginning), e.g. for border control purposes (see proposals in the Chapter 4.4). However, we are not limited only to verification or identification systems; we can also integrate a biometric system into a cryptographic system (or otherwise) and thus obtain a Biometric Security System (which is the main theme of this dissertation).

In the first chapter, fundamental biometric terms, such as biometric data, have been discussed and a biometric system has been described, with its processes, pros and cons, difference between verification and identification, etc. Very important part comes at the end of the first chapter, namely the error rates. The error rates are used for the evaluation of biometric systems [3] and we can determine on the basis of **FTA**, **FTE**, **FTM** + **FAR** (**FMR**), **FRR** (**FNMR**) + **ROC**, which of the tested systems (algorithms) is better or the best one. The same evaluation can be made for the acquisition devices (sensors). Some testing scenarios [3] can be used, e.g. constant sensor but changing algorithms, or constant algorithm and changing sensors. Such error rates can also be used for the evaluation of the reliability of the Biometric Security System.

The second chapter describes the actual state of biometric systems based on the fingerprint technology. This chapter starts with some introduction into the acquisition devices (sensors), fingerprint classification and fingerprint matching. The middle part of this chapter discusses the fingerprint recognition system with its main processes which represent a basis for further development. In the first steps of fingerprint recognition, the fingerprint images are processed and the results (minutiae) of such processing are needed for further usage in this work. The detailed description of these processes is out of the scope of this work and can be found, e.g., in [3, 8, 10, 22, 36, 42, 47, 48, 50, 58, 63, 65, 71, 76, 80, 90, 117]. The attention has been focused on the fingerprint recognition systems based on minutiae extraction and matching, because the proposed Biometric Security System exploits the minutiae details. The fingerprint classification is described more in detail, because one specific sub-step of such classification has been selected for the estimation or determination of the center, namely the procedure based on the Orientation Field computation. At the end of this chapter, some actual solutions for the application of biometric attributes to other purposes than for verification or identification tasks are presented. Some of the discussed systems can protect (encipher) the data using the biometric attribute. But none of the actual systems describes the solution proposed in this dissertation.

Finally, after introduction chapters, we come to the third chapter, which is related to another important question: Is there enough entropy or information strength in the fingerprint, so that the information from the fingerprint (in our case: set of minutiae) could be used as a random input for key generation? This chapter starts with a general introduction to the information theory (independent events, *Bayes'* Theorem, Entropy). The chapter continues by considerations related to the uniqueness (individuality) of fingerprints and then focuses on the problem of matching two fingerprints. The last part of this chapter is reserved to the development of my own fingerprint uniqueness model (which is the first goal of this work). First of all, it was necessary to define the role of image resolution (sensor vs. papillary line), fingerprint size (area) and terms minutia / antiminutia. Then we were able to describe the entropy factors on the basis of minutiae (and their coordinates, types and gradients, together with additional parameter defined as maximal number of placeable points in some defined matrix). Further, we have been able to define the minimal (12 minutiae $\Rightarrow 2^{243}$) and maximal (368 minutiae $\Rightarrow 2^{8075}$) entropy factors. As the input information (fingerprint image) varies, the positions of minutiae are too precise and therefore some increase of granularity or module size

is needed, and this step is described as quantization. If we define different quantization factors, the entropy factors for small values are still acceptable for the biometric key length, taking in consideration the cryptographic requirements for the key length. As the result, we have been able to state that the information in the fingerprint (with an average amount of minutiae and after quantization) is suitable for the symmetric cryptography and cryptography based on elliptic curves but it is not sufficient for the asymmetric cryptography.

The following fourth chapter describes the proposal of a Biometric Security System (which is the second goal of this work) based on the fingerprint technology. This proposal can be extended or used for another biometric attribute – see Fig. 4.12 and Fig. 5.31.

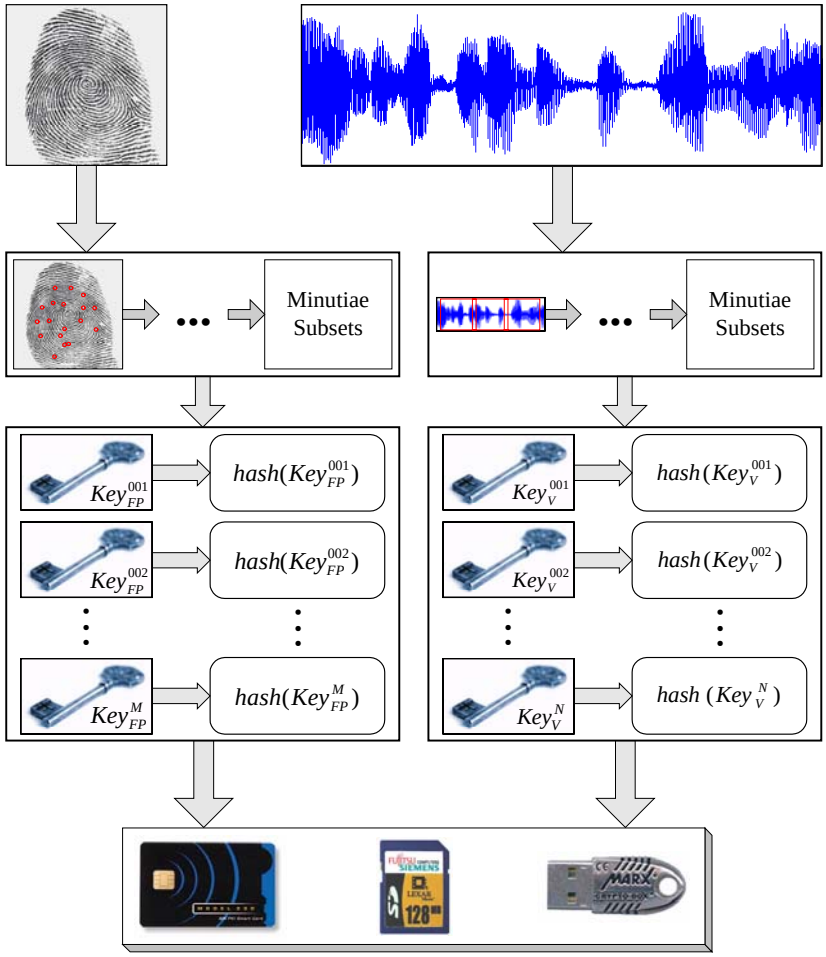


Fig. 5.31: Biometric Security System (fingerprint and voice technology) [31]

There are two concepts in the proposed Biometric Security System – Certificate Creation and Certificate Usage – and it is necessary to distinguish the differences between them. The first concept is generated only once (when neglecting e.g. expiration of certificate validity), preferably by a system administrator (CA). The second concept relates to the daily usage of such certificate for many applications – see two practical usage proposals in Chapter 4.4. All sub-steps of both concepts

(i.e. center computation, minutiae amount reduction, reference minutia estimation, creation of oriented closed graph, quantization, sub-graph generation and hash computation) have been described in detail in this fourth chapter (which is the third goal of this work).

Finally, the proposed Biometric Security System based on fingerprint technology has been implemented and tested. For the purpose of testing, a fingerprint database containing 12.500 fingerprints has been acquired or created. Three sensor types and a rolled fingerprint method (similar to dactyloscopic procedure) have been used for the acquisition. The implementation was realised in steps described in the Chapter 4. Two industrial algorithms (Siemens and Veridicom) have been used for the comparison. The results of key generation cannot be exactly compared with results of industrial algorithms, because only 100% correspondence of both sub-graphs is needed and accepted as match. Using the results of both industrial algorithms, we can make some conclusions relating to the quality of sensors / users and probability of good or bad results of key generation. The complete description of all results from testing phase, including their graphical representation, can be found in this chapter.

As the last question, we can ask where the strong or weak points of this system are and what can be improved. The answer to the first part of this question can be actually found in the Chapter 4.4. These two usage proposals are not the only possibilities. As soon as the reliability of this system (after optimization) achieves certain quality level, this solution can be implemented on credit cards (with biometric authentication and protection of secret data) or for daily data protection (e.g. for encryption of private data on USB memory sticks). The strong point of this solution consists furthermore in the amount of entropy which is considerably greater than in passwords or PINs used for the protection of the access to cryptographic keys nowadays. The weaknesses include e.g. the fact that the user needs to repeat the acquirement of the fingerprint when no match of hash value is found, and of course another fact that the liveness testing is necessary, i.e. it needs to be proven that the user is alive.

What could be still improved in the system? Each sub-step of the whole key generation process has enough space for optimization, e.g. the center computation method should be optimized so that the differences are very small, or the Method_3 should be more tolerant to images with worse quality. When we consider the percentage amounts of total matches (suitable for hash match), it is clear that the improvement of all sub-steps is needed because we surely would not want to repeat the acquirement of our fingerprints more times. On the other hand, the functionality of such solution has been proven – there are some total matches. After the optimization of all sub-steps, the data volume needs to be reduced. The actual storage devices are able to store such data volumes, but if we want to encapsulate such certificates into small chips, we need an efficient data reduction. I would like to state again that, according to my opinion, the main goal – to show that the proposed system is viable and operational – has been achieved. However, the space for further improvements, optimization and other usage suggestions still exists.

6. Acronyms

AFIS	Automated Fingerprint Identification System
ATM	Automatic Teller Machine
AVG	Average
BKA	Bundeskriminalamt
BSS	Biometric Security System
CA	Certification Authority
CBEFF	Common Biometric Exchange File Format
CCD	Charge Coupled Device
CPU	Central Processing Unit
CS	Coordinate System
DB	Database
DES	Data Encryption Standard
DIN	Deutsche Industrie Norm
DNA	Deoxy Nucleic Acid
DPI	Dots Per Inch
EER	Equal Error Rate
Eq.	Equation
FAR	False Acceptance Rate
FBI	Federal Bureau of Investigation
FDB	Fingerprints Database
FEAL	Fast Data Encipherment Algorithm
FMR	False Match Rate
FNMR	False Non-Match Rate
FP	Fingerprint
FRR	False Rejection Rate
FTA	Failure To Acquire (Rate)
FTE	Failure To Enroll (Rate)
FTIR	Frustrated Total Internal Reflection
FTM	Failure To Match (Rate)
FVC	Fingerprint Verification Competition
HMM	Hidden Markov Model
ID	Identity
LED	Light Emitting Diode
LPC	Linear Predictive Coding

MAC	Message Authentication Code
MD	Message Digest
MDS	Modification Detection Code
MFCC	Mel-Frequency Cepstrum Coefficients
MS	Match Score
OF	Orientation Field
PC	Personal Computer
PIN	Personal Identification Number
PKI	Public Key Infrastructure
PNN	Probabilistic Neural Network
RAM	Random Access Memory
RC	Ridge Count
RFC	Reference For Comments
ROC	Receiver Operating Curve
RSA	Rivest, Shamir, Adleman (Encryption Algorithm)
SDFC	Speaker Dependent Frequency Cepstrum
SDK	Software Development Kit
SP	Service Pack
USB	Universal Serial Bus
VAD	Voice Activity Detection

7. References

[1] Adams, C., Farrell, S.: *Internet X.509 Public Key Infrastructure – Certificate Management Protocols*, Entrust Technologies & SSE, 1999

[2] Adler, A.: *Reconstruction of Source Images from Quantized Biometric Match Score Data*, School of Information Technology and Engineering, University of Ottawa, 2004

[3] Arnold, M., Busch, C., Drahansky, M., Ihmor, H., Reinefeld, T., Zwiesele, A: *BioFinger – Evaluierung biometrischer Systeme (Fingerabdrucktechnologien)*, Darmstadt, FHG-IGD, 2004

[4] Ashbourn, J.: *Practical Biometrics – From Aspiration to Implementation*, Springer Verlag, 2004, ISBN 1-85233-774-5

[5] Aufreiter, R.: *Biometrie und Kryptographie – Match On Card als Weg zur komfortablen Sicherheit*, White Paper, Utimaco Safeware AG, 2001

[6] Aufreiter, R.: *Der Finger als Schlüssel – Aktuelle Biometrieverfahren im praktischen Einsatz*, Utimaco Safeware AG, 2003

[7] Bazen, A.M., Verwaaijen, G.T.B., Gerez, S.H., Veelenturf, L.P.J., Zwaag, B.J.: *A Correlation-Based Fingerprint Verification System*, University of Twente, The Netherlands, STW-2000, ISBN 90-73461-24-3

[8] Bhanu, B., Tan, X.: *Computational Algorithms for Fingerprint Recognition*, Kluwer Academic Publishers, USA, 2004, ISBN 1-4020-7651-7

[9] Black, P.E.: *Dictionary of Algorithm and Data Structures*, NIST, 2004

[10] Bolle, R.M., Connell, J.H., Pankanti, S., Ratha, N.K., Senior, A.W.: *Guide to Biometrics*, Springer Verlag, 2004, ISBN 0-387-40089-3

[11] Boneh, D., Franklin, M.: *Identity-Based Encryption from the Weil Pairing*, Stanford University, Proceedings of 21st International Conference Advances in Cryptography, 2001

[12] Bonfig, K.W., Drahansky, M.: *Biometrie*, Book, Kreuztal, 2004, p. 153, ISBN 3-933609-02-X

[13] Boyen. X.: *Multipurpose Identity-Based Signcryption*, Proceedings of 23rd International Conference Advances in Cryptography, 2003, Springer

[14] Bromba, M., Hribernig, G., Scheiter, T.: *Daumen als Schlüssel*, Funkschau 13/1998

[15] Bucholcer, J., Škvarek, J., Dudycha, V.: *Konvoluční metoda biometrického šifrování soukromého klíče pro zaručený elektronický podpis*, Vojenská akademie v Brně, Z.L.D. s.r.o., 2001

[16] Burrows, J.H.: *Security Requirements for Cryptographic Modules*, U.S. Department of Commerce / National Institute of Standards and Technology, 1994

[17] Canisius, L., Lappe, U.: *Fingerabdruckerkennung in der Haustechnik*, in2systems, 2002

[18] CC: *Biometric Technology Security Evaluation under the CC*, Government of Canada, Communications Security Establishment, 2001

[19] Chirillo, J., Blaul, S.: *Implementing Biometric Security*, Wiley Publishing, 2003, ISBN 0-7645-2502-6

[20] N.N.: *Biometrics and PKI Based Digital Signatures*, A Short White Paper, Daon, 2003

[21] Dornberger, R., Probst, F.: *Biometrics – Ein Überblick,* Fachhochschule Solothurn Nordwestschweiz, 2003

[22] Drahansky, M.: *Fingerabdruckerkennung mittels neuronaler Netze*, Diploma Thesis, Brno University of Technology, CZ, 2001

[23] Drahansky, M., Orsag, F.: *Fingerprints and Speech Recognition as parts of the Biometry*, Proceedings of 36th International Conference MOSIS'02, Ostrava, MARQ, 2002, CZ, p. 177-183, ISBN 80-85988-71-2

[24] Drahansky, M.: *Fingerprint Recognition*, 16th Biennial International Eurasip Conference BioSignal 2002, Brno, MUNI, CZ, p. 301-303, ISBN 80-214-2120-7

[25] Drahansky, M.: *Fingerprint Recognition and Smart Cards*, Proceedings of 8th EEICT Conference, Brno, FIT BUT, CZ, 2002, p. 436-440, ISBN 80-214-2116-9

[26] Drahansky, M., Orsag, F.: *Biometric Security Systems: Fingerprint and Speech Technology*, Proceedings of the 1st Indian International Conference on Artificial Intelligence, Tallahassee, IICAI 2003, USA, p. 703-711, ISBN 0-9727412-0-8

[27] Drahansky, M., Orsag, F., Smolik, L.: *Biometric Security Systems*, Proceedings of St. Nicholas Security Workshop, Prague, ECOM, CZ, 2003, p. 1-10, ISBN 80-903083-3-3

[28] Drahansky, M., Smolik, L., Orsag, F.: *Entwurf eines biometrischen Sicherheitssystems*, Bonn, Bundesamt für Sicherheit in der Informationstechnik, GE, 2003, p. 4

[29] Drahansky, M., Nötzel, R., Bonfig, K.W.: *Sensoren zur Fingerabdruckerkennung*, Sensoren, Signale, Systeme, Kreuztal, 2004, GE, p. 49-60, ISBN 3-933609-19-4

[30] Drahansky, M., Orsag, F., Zboril, F.: *Biometrics in Security Applications*, Proceedings of 38th International Conference MOSIS'04, Ostrava, MARQ, CZ, 2004, p. 6, ISBN 80-85988-98-4

[31] Drahansky, M., Orsag, F.: *Biometric Security Systems: Robustness of the Fingerprint and Speech Technologies*, BT2004 – International Workshop on Biometric Technologies, Calgary, CA, 2004, p. 99-103

[32] Drahansky, M., Smolik, L.: *Entropic Numbers from the Fingerprint*, BMWA – Workshop on Biometrics, Royal Statistical Society, London, GB, 2004

[33] Drahansky, M.: *Nutzung biometrischer Daten zur Gewinnung personenbezogener kryptographischer Schlüssel*, Biometrie – BIOSIG2004, Fraunhofer Gesellschaft – IGD, Darmstadt, GE, 2004

[34] Drahansky, M.: *Multibiometric Systems*, Proceedings of the 10th Conference and Competition STUDENT EEICT 2004, Brno, FIT BUT, CZ, 2004, p. 5, ISBN 80-214-2635-7

[35] Drahansky, M., Smolik, L.: *Biometric Certificates*, DSM – Data Security Management, Vol. 8, No. 4, p. 4, ISSN 1211-8737

[36] Emiroglu, I.: *Automatic Fingerprint Enhancement and Feature Extraction System*, University of Hertfordshire, 1997

[37] Engl, P., Peter, C.: *Fingerabdruckerkennung als Anwendung neuronaler Netze*, Diploma Thesis, FH Regensburg, 2000

[38] Fisher, B., Perkins, S., Walker, A., Wolfart, E.: *Hypermedia Image Processing Reference*, Department of Artificial Intelligence, University of Edinburgh, UK

[39] Frenzen, C.L.: *Convolution for Mathematical Problems in Biometrics*, National Biometric Test Center, Collected Works 1999 – 2000

[40] Froehling, W.: *Konzept und exemplarische Implementation eines gesicherten Kanals zur Übertragung biometrischer Daten*, University Hamburg, 2003

[41] Fuscaldo, D.: *Fingerprints May Be Key to PC Security*, Dow Jones News Service, 2002

[42] Garris, M.D., Watson, C.I., McCabe, R.M., Wilson, C.L.: *NIST Fingerprint Image Software (NFIS)*, NISTIR 6813, National Institute of Standards and Technology, 2001

[43] Gellert, W., Küstner, H., Hellwig, M., Kästner, H., Reichardt, H.: *Kleine Enzyklopädie – Mathematik*, VEB Bibliographisches Institut Leipzig, 1986

[44] Göhler, W.: *Formelsammlung (Höhere Mathematik)*, Verlag Harri Deutsch, 1999, ISBN 3-8171-1592-X

[45] Haseltine, E.: *The Future of Science (and Biometrics)*, 2003

[46] Henke, S.: *Verfahren der biometrischen Authentisierung und deren Unterstützung durch Chipkarten*, University in Hamburg, 1999

[47] Hong, L.: *Automatic Personal Identification Using Fingerprints*, Michigan State University, Department of Computer Science, 1998

[48] Hong, L., Jain, A., Pankanti, S., Bolle, R.: *Fingerprint Enhancement*, Michigan State University, Department of Computer Science, 1998

[49] Hong, L., Jain, A.: *Multimodal Biometrics*, Michigan State University, Department of Computer Science, 1998

[50] Hong, L., Wan, Y., Jain, A.: *Fingerprint Image Enhancement: Algorithm and Performance Evaluation*, Michigan State University, Department of Computer Science, 1998

[51] Hongning, F., Zhu, X.: *Oriented Walk Double Covering and Bidirectional Double Tracing*, Shandong University & National Sun Yat-sen University, 2003

[52] Huffman, W.C., Pless, V.: *Fundamentals of Error-Correcting Codes*, Loyola University of Chicago & University of Illinois at Chicago, 2003

[53] Hurlburt, R.T.: *Comprehending Behavioral Statistics*, Thomson Wadsworth™, USA, 2003, ISBN 0-534-60102-2

[54] Ingram, D., Bloch, R.F.: *Mathematical Methods in Medicine – Part I: Statistical and Analytical Techniques*, John Wiley & Sons Ltd., 1984, ISBN 0-471-90045-1

[55] Jain, A., Hong, L., Bolle, R.: *On-line Fingerprint Verification*, Michigan State University & IBM T.J. Watson Research Center

[56] Jain, A., Hong, L., Pankanti, S., Bolle, R.: *An Identity Authentication System Using Fingerprints*, Michigan State University & IBM T.J. Watson Research Center, 1997

[57] Jain, A., Hong, L., Kulkarni, Y.: *A Multimodal Biometric System Using Fingerprint, Face, and Speech*, Michigan State University, Department of Computer Science and Engineering, 1999

[58] Jain, A., Pankanti, S.: *Fingerprint Classification and Matching*, Michigan State University + IBM T.J. Watson Research Center, 2001

[59] Jain, A.: *Hierarchical Kernel Fitting for Fingerprint Classification and Alignment*, ICPR, 2002

[60] Jozefek, A.: *Principy některých daktyloskopických klasifikačních systémů*, Ústav kriminalistiky Právnické fakulty UK, 1972

[61] Juels, A., Sudan, M.: *A Fuzzy Vault Scheme*, RSA Laboratories + MIT Laboratory for Computer Science, 2002

[62] Jung, D.W., Park, R.H.: *Robust Fingerprint Identification Based on Hybrid Pattern Recognition Methods*, Department of Electronic Engineering, Sogang University, Korea, World Scientific, 2001

[63] Karu, K., Jain, A.K.: *Fingerprint Classification*, Michigan State University, Department of Computer Science, 1995

[64] Kasaei, S., Deriche, M., Boashash, B.: *Fingerprint Feature Enhancement using Block-Direction on Reconstructed Images*, Signal Processing Research Centre, Australia, 1997

[65] Kay, K.: *Introduction to Fingerprint Recognition*, 2003

[66] Köhntopp, M.: *Technische Randbedingungen für einen datenschutzgerechten Einsatz biometrischer Verfahren*, Proceedings zur Arbeitskonferenz Sicherheitsstrukturen, Hamburg, 1999

[67] Kong, A., Griffith, A., Rhude, D., Bacon, G., Shah, S.: *Department of Defense & Federal Biometric System Protection Profile for Medium Robustness Environments*, Common Criteria DoD, 2002

[68] Kung, S.Y., Mak, M.W., Lin, S.H.: *Biometric Authentication – A Machine Learning Approach*, Prentice Hall Information and System Science Series, 2004, ISBN 0-131-47824-9

[69] Lewis, S., Steigerwalt, T.: *Biometric Encryption*, Roberto C. Goizueta Business School, Emory, 2004

[70] Maio, D., Cappelli, R., Lumini, A., Maltoni, D.: *Fingerprint Classification by Directional Image Partitioning*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 21, No. 5, 1999

[71] Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: *Handbook of Fingerprint Recognition*, Springer, New York, 2003, ISBN 0-387-95431-7

[72] Maltoni, D., Jain, A.K.: *Biometric Authentication – ECCV 2004 International Workshop BioAW 2004 in Prague*, Springer Verlag, ISBN 3-540-22499-8

[73] Mansfield, A.J., Wayman, J.L.: *Best Practices in Testing and Reporting Performance of Biometric Devices*, National Physical Laboratory & San Jose State University, 2002

[74] Marcialis, G.L., Roli, F., Loddo, P.: *Fusion of Multiple Matchers for Fingerprint Verification*, University of Cagliari, Italy, 2002

[75] Marzban, C.: *A Comment on the ROC Curve and the Area Under it as Performance Measures*, University of Washington & University of Oklahoma, 2004

[76] Matyas, V., Riha, Z.: *Biometric Authentication Systems*, Masaryk University & Ecom-Monitor.com, 2000

[77] Menezes, A.J., Oorschot, P.C., Vanstone, S.A.: *Handbook of Applied Cryptography*, CRC Press, 1996, ISBN 0-8493-8523-7

[78] Müller, R.: *Fingerprint Verification with Microprocessor Security Tokens*, Munich University of Technology, 2001, ISBN 3-8316-0015-5

[79] Nanavati, S., Thieme, M., Nanavati, R.: *Biometrics – Identity Verification in a Networked World*, Wiley, 2002, ISBN 0-471-09945-7

[80] Nolde, V., Leger, L.: *Biometrische Verfahren*, Fachverlag Deutscher Wirtschaftsdienst GmbH & Co. KG, Cologne, 2002, ISBN 3-87156-464-8

[81] Orsag, F.: *Biometric Security Systems: Speaker Recognition Technology*, Dissertation Thesis, FIT BUT, 2004

[82] Pankanti, S., Prabhakar, S., Jain, A.K.: *On the Individuality of Fingerprints*, IBM T.J. Watson Research Center + DigitalPersona Inc. + Michigan State University, 2001

[83] Paulson, P.J.: *What is Fingerprint Biometric Encryption?*, Biometrics, 2002

[84] Petermann, T., Scherz, C., Sauter, A.: *Biometrie und Ausweisdokumente*, Arbeitsbericht Nr. 93, TAB - Büro für Technikfolgen – Abschätzung beim Deutschen Bundestag, 2003

[85] Plaz, M., Lang, A.: *Fingerprint Authentication System*, HSR – Hochschule für Technik, Rapperswil, 2004

[86] Prabhakar, S., Jain, A.K., Pankanti, S.: *Learning Fingerprint Minutiae Location and Type*, DigitalPersona Inc. & Michigan State University, Department of Computer Science and Engineering & IBM T.J. Watson Research Center, 2002

[87] Ratha, N.K., Chen, S., Jain, A.K.: *Adaptive Flow Based Feature Extraction in Fingerprint Images*, Michigan State University, Department of Computer Science, 1995

[88] Ratha, N.K., Connell, J.H., Bolle, R.M.: *Enhancing Security and Privacy in Biometrics-Based Authentication Systems*, IBM Systems Journal, Vol. 40, No. 3, 2001

[89] Ratha, N.K., Senior, A., Bolle, R.M.: *Automated Biometrics*, IBM Thomas J. Watson Research Center, 2003

[90] Ratha, N., Bolle, R.: *Automatic Fingerprint Recognition Systems*, Springer Verlag, 2004, ISBN 0-387-95593-3

[91] Rila, L., Mitchell, C.J.: *Security Protocols for Biometrics-Based Cardholder Authentication in Smartcards*, University of London, 2003

[92] Ross, A., Jain, A.: *Fingerprint Mosaicking*, ICASSP, 2002

[93] Ross, A.: *Information Fusion in Fingerprint Authentication*, Dissertation Thesis, Michigan State University, Department of Computer Science and Engineering, 2003

[94] Santesson, S., Nystrom, M., Polk, T.: *Internet X.509 Public Key Infrastructure – Qualified Certificates Profile*, Microsoft & RSA Security & NIST, 2004

[95] Shapiro, D.L., Swan, D., Heinrichs, M.: *Advanced Wireless Security using Biometric Encryption for Sender Validation*, Integrated Technology Solutions Inc., Columbia, 2002

[96] Schuckers, S., Hornak, L., Norman, T., Derakhshani, R., Parthasaradhi, S.: *Issues for Liveness Detection in Biometrics,* West Virginia University and Center for Identification Technology Research, 2003

[97] *Security Requirements for Cryptographic Modules*, FIPS PUB 140-1, Federal Information Processing Standards Publication, National Institute of Standards and Technology, 1994

[98] Smith, R.E.: *How Authentication Technologies Work*, 2002

[99] Smith, R.E.: *An Overview of Authentication Techniques*, Secure Computing, 2002

[100] Soutar, C., Roberge, D., Stoianov, A., Gilroy, R., Kumar, V.: *Biometric Encryption*, Bioscrypt Inc., 2001

[101] Stinson, D.: *Cryptography: Theory and Practice*, CRC Press LLC, 1995, ISBN 0849385210

[102] Stroustrup, B.: *The C++ Programming Language*, Addison-Wesley, 1991, ISBN 0-201-53992-6

[103] Stroustrup, B.: *Die C++ Programmiersprache*, Addison-Wesley, 2000, ISBN 3-8273-1660-X

[104] Struif, B., Müller, R.: DIN V 66400, 2002

[105] Straub, T.: *Spezifikation von X.509-Zertifikatsprofilen unter dem Gesichtspunkt Benutzbarkeit*, Technical University Darmstadt, 2004

[106] Sudan, M.: *Coding Theory: Tutorial / Survey*, FOCS 2001

[107] Swart, B., Cashman, M., Gustavson, P., Hollingworth, J.: *Borland C++ Builder 6 Developer's Guide*, SAMS Publishing, 2003, ISBN 0-672-32480-6

[108] N.N.: *The Tale of Two Cities – Biometrics and Cryptography*, 2003

[109] N.N.: *Theories and Applications of Biometrics*, 2001

[110] Uludag, U., Jain, A.K.: *Multimedia Content Protection via Biometrics-Based Encryption*, Proceedings of the International Conference on Multimedia and Expo, Baltimore, 2003

[111] Uludag, U., Pankanti, S., Prabhakar, S., Jain, A.K.: *Biometric Cryptosystems: Issues and Challenges*, Proceedings of the IEEE, Vol. 92, No. 6, 2004

[112] Waldmann, U., Scheuermann, D., Eckert, C.: *Schutz biometrischer Daten bei Authentisierung auf Smartcards*, Fraunhofer Gesellschaft – Institut für Sichere Telekooperation, Darmstadt, 2003

[113] Wikipedia: *Cryptographic Key Length*, Wikipedia – The Free Encyclopedia, 2004

[114] Wilson, C.L., Candela, G.T., Watson, C.I.: *Neural Network Fingerprint Classification*, National Institute of Standards and Technology, 1993

[115] Williams, L.C.: *A Discussion of the Importance of Key Length in Symmetric and Asymmetric Cryptography*, SANS Institute, 2002

[116] Woodward, J.D., Orlans, N.M., Higgins, P.T.: *Biometrics – Identity Assurance in the Information Age*, McGraw-Hill / Osborne, 2003, ISBN 0-07-222227-1

[117] Zhang, D., Jain, A.K.: *Biometric Authentication – First International Conference ICBA 2004 in Hong Kong*, Springer Verlag, 2004, ISBN 3-540-22146-8