2009

Establishing the digital chain of evidence in biometric systems

Nick Bartlow
West Virginia University

Establishing the Digital Chain of Evidence in Biometric Systems

Nick Bartlow

Dissertation submitted to the
College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy
in
Computer and Information Science

Bojan Cukic, Ph.D., Chair
Lan Guo, Ph.D.
Lawrence Hornak, Ph.D.
Keith Morris, Ph.D.
Arun Ross, Ph.D.
Natalia Schmid, Ph.D.

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2009

Keywords: Chain of Evidence, Chain of Custody, Biometric Watermarking, Hardware Fingerprinting, Image Forensics, Keystroke Dynamics, Biometric System Frameworks.

Abstract

Establishing the Digital Chain of Evidence in Biometric Systems

Nick Bartlow

Traditionally, a chain of evidence or chain of custody refers to the chronological documentation, or paper trail, showing the seizure, custody, control, transfer, analysis, and disposition of evidence, physical or electronic. Whether in the criminal justice system, military applications, or natural disasters, ensuring the accuracy and integrity of such chains is of paramount importance. Intentional or unintentional alteration, tampering, or fabrication of digital evidence can lead to undesirable effects. We find despite the consequences at stake, historically, no unique protocol or standardized procedure exists for establishing such chains. Current practices rely on traditional paper trails and handwritten signatures as the foundation of chains of evidence.

Copying, fabricating or deleting electronic data is easier than ever and establishing equivalent digital chains of evidence has become both necessary and desirable. We propose to consider a chain of digital evidence as a multi-component validation problem. It ensures the security of access control, confidentiality, integrity, and non-repudiation of origin. Our framework, includes techniques from cryptography, keystroke analysis, digital watermarking, and hardware source identification. The work offers contributions to many of the fields used in the formation of the framework. Related to biometric watermarking, we provide a means for watermarking iris images without significantly impacting biometric performance. Specific to hardware fingerprinting, we establish the ability to verify the source of an image captured by biometric sensing devices such as fingerprint sensors and iris cameras. Related to keystroke dynamics, we establish that user stimulus familiarity is a driver of classification performance. Finally, example applications of the framework are demonstrated with data collected in crime scene investigations, people screening activities at port of entries, naval maritime interdiction operations, and mass fatality incident disaster responses.

This work is dedicated to my family for their unwavering support, encouragement, and understanding

And to my fiancée Kellyn, for putting up with me during the "double shifts" over the long haul

Acknowledgements

- *To Dr. Bojan Cukic*- I offer my deepest thanks to my advisor for all that he has provided me over the years. The wisdom, guidance, and experience he has imparted on me is truly priceless. As a mentor, he has exposed me to a special ability to motivate and inspire others in good times and bad. I thank him for always sensing when I was on the brink and giving me the chance to temporarily reacquire sanity. Finally, I am extremely grateful for unique opportunities he has selflessly provided me when he could have easily done otherwise.

- *To Dr. Arun Ross*- I thank him for never failing to field my incessant questioning whether in class or in his office. His innate ability to convert relatively "inaccessible" knowledge into something laymen such as myself can digest is unprecedented. I also thank him for his willingness to work on papers with me providing insightful commentary regardless of the time of night.

- *To Dr. Lawrence Hornak*- I offer many thanks for his attempts to prevent me from embarrassing myself by eliminating comments likely to cause "inflamatory" effects on the sensors folks and the greater electrical engineering community. Additionally, I'm very thankful for having had the opportunity to closely observe a man with acutely refined leadership and management skills.

- *To Dr. Natalia Schmid*- I thank her for peaking my interests in the field of biometrics many years back. Her class was fundamental in paving the start of my path through grad school. Additionally, thank you for patiently explaining signals theory and associated mathematical concepts to me in class and office.

- *To Dr. Nancy Lan Guo and Dr. Keith Morris*- I offer my thanks for providing insightful perspectives outside of the biometric applications I am typically accustomed to dealing with. I believe your commentary has been extremely helpful in giving this work an opportunity to have as broad an impact as possible.

- *To My Lab Mates*- I'd like to thank Nathan Kalka, Yue Jiang, Mayra Sacanamboy, Dejan Desovski, Petar Popic, Nevena Stojanovic, Paola Bracchi, Yan Liu, David Owen, Qi Cheng, and Yan Ma who have all helped me at one time or another throughout my work on my degree. Whether providing help on a specific problem, sympathy for our mutually chosen (masochistic) path, or a model for which to base my behavior as a grad student, thank you all for your support throughout the years.

Contents

List of Figures

This page intentionally contains only this sentence.

List of Tables

Chapter 1

Introduction

1.1 Motivation

Traditionally, a chain of evidence or chain of custody refers to the chronological documentation, or paper trail, showing the seizure, custody, control, transfer, analysis, and disposition of evidence, physical or electronic [142]. Whether in law enforcement, homeland security, or military operations, ensuring the accuracy and integrity of such chains is of paramount importance. Intentional or unintentional alteration, tampering, or fabrication of evidence in such arenas can lead to undesirable effects. In law enforcement and certain homeland security operations, the chain of evidence must establish, according to the requirements of the criminal justice system, that alleged evidence is indeed related to the crime in question as opposed to having been planted fraudulently. In military operations, identification of enemy combatants is critical to the success of operational objectives and such identifications must be verifiable and logged accordingly to justify various courses of action. In homeland security operations such as responding to natural disasters, the process of documenting the identity of victims to be passed on to family members must be checked and rechecked to prevent dispersal of erroneous information. Regardless of the application, there is typically an underlying theme of identifying individuals, which naturally entails biometric data. Examples of such biometric data might include face, fingerprint, or iris images. While traditional standards require the creation of a trail of paper documentation, creation and enforcement of electronic equivalents of such trails is now possible. Besides facilitating more efficient transmission and storage, adaptation of electronic chains of evidence can potentially offer a greater capacity to verify evidence, especially when applied to biometric data. Development of such a framework is an essential starting point for organizations that wish to enhance, or replace currently existing paper based evidence chains and develop a means to establish and maintain digital chains of evidence.

1.2 Goal

Creating a verifiable digital chain of evidence simplifies to a multi-component validation problem. At any given point in time, an individual should be able to validate the content, transmission, and source of evidence. Content validation refers to whether evidence was inappropriately modified at one entity in the chain before being transmitted to another entity. Transmission validation deals with whether evidence was inappropriately modified during the transmission from one entity to the next. Finally, source validation ensures that data originated from the source in which it was claimed to originate from. The goal of this work is to develop a conceptual framework for creating a digital chain of evidence which allows for validation at the three aforementioned levels. To do so, we first investigate and develop topics which can be employed as security mechanisms within the framework including: biometric watermarking, digital hardware fingerprinting, and the behavioral biometric of keystroke dynamics. Next we develop the framework which relies upon these three security mechanisms and cryptography. Specifically, we use cryptography to handle security threats of data interception, data modification, data fabrication, repudiation of origin, and denial of receipt. Biometric watermarking is applied to add another layer of protection against data interception, data modification, data fabrication, and repudiation of origin. Keystroke dynamics (or any other biometric) is used to prevent unauthorized user access. Finally, digital hardware fingerprinting is also used to prevent data modification and data fabrication.

1.3 Contributions

This work provides the following set of original contributions:

1. Development of an iris digital watermarking system which is not only resistant to common application scenarios such at database compression and partial progressive decoding, but also capable of withstanding the rewatermarking process which might be seen in the proposed chain of evidence. Our work extends current biometric watermarking techniques by modifying existing approaches to allow for selective encoding in the region of interest in iris biometric images. Additionally, our approach provides a novel asymmetric implementation of the watermarking scheme.

2. Demonstration of the ability to perform source validation on biometric modalities which collect data with capture devices outside of typical photographic cameras. We apply an

approach designed for digital cameras to a series of biometric fingerprint readers as well as iris cameras using sensors which respond to the infrared band of the electromagnetic spectrum.

3. Demonstration that input stimulus familiarity is a driver of classification performance in keystroke dynamic systems.
4. Development of a conceptual framework for establishing and maintaining digital chain of evidence dealing with biometric data which relies on elements of cryptography, biometric watermarking, digital hardware fingerprinting, and biometrics. This also includes the process of vetting the framework against security threats related to confidentiality and integrity. Finally, we include an example instantiation of the conceptual framework using specific examples from each of the security mechanisms upon which framework relies.

1.4 Organization

The remainder of this dissertation is organized as follows. Chapter 2 provides a summary of related work regarding traditional evidence chains, biometric watermarking, hardware fingerprinting, and keystroke dynamics. Chapters 3-5 describe specific contributions in the areas of iris digital watermarking, digital hardware fingerprinting, and keystroke dynamics. Chapter 6 describes the proposed conceptual framework for establishing and maintaining digital chains of evidence. Finally, Chapter 7 concludes the work by providing a summary of the accomplishments as well as future directions for research.

Chapter 2

Related Work

This chapter provides a summary of the related work in the fundamental areas of research relevant to this effort. The fields of interest include traditional evidence chain systems, biometric watermarking, digital hardware fingerprinting, and keystroke dynamics. While no summary can be complete, each subsection attempts to offer sufficient breadth and depth to provide the reader with a basic working knowledge of the field.

2.1 Chain of Evidence

2.1.1 Introduction

According to Breitman, a "chain of custody", synonymous with a "chain of evidence" in our work, refers to the to "the history of a piece of physical evidence after it has been identified and preserved" [21]. Another definition is offered by Wikipedia, "Chain of custody refers to the chronological documentation, and/or paper trail, showing the seizure, custody, control, transfer, analysis, and disposition of evidence, physical or electronic." Most commonly found in criminal justice systems, a chain of evidence serves two primary purposes as defined by McCormick's "On Evidence" [31], "First, it identifies the object being offered, to establish that it is in fact the object that caused or contributed to the plaintiffs injury. This is described as "authentication." The second purpose is to establish that the condition of the object is substantially the same as when the injury occurred". While these services are specifically aimed at criminal justice applications, they need only be altered slightly to generalize to any application which deals with physical or electronic evidence. Perhaps such a generalization can be found in the term "evidence management" defined as "the administration and control of evidence related to an event so that it can be used to prove the circumstances of the event, and so that this proof can be tested by independent parties with confidence that the evidence provided is the evidence collected related to the event" [143]. Whether in the criminal justice system, military operations, or other arenas, this documentation and management of evidence is a critical component to ensure the proper course of actions are taken in the systems which it supports.

2.1.2 Traditional Paper Based Chains

As mentioned in the introduction, traditionally, chains of evidence have been tied to the criminal justice systems. Given the consequences, one might think there would be a widely accepted protocol for establishing a chain of evidence or chain of custody. We find however, historically, no unique protocol or standardized procedure exists. Breitman states that "Under the Federal Rules of Evidence, that standard of proof requires only that the party offering an item of physical evidence introduces such proof as is "sufficient to support" a finding that the [item in question] is what its proponent claims: Federal Rule of Evidence, 901(a)" [21] [58]. Additionally, he points out that "the ultimate issues of authentication and condition are left for the jury. If the proponent's evidence is facially sufficient to support a finding of authenticity, then all other issues such as credibility and probability are left for the jury [21]." Taking these accounts into consideration, the onus of establishing a burden of proof in a criminal justice system falls on many shoulders. At the highest level, it is the responsibility of prosecutors and defense attorneys. At the lowest level, it is the responsibility of officers, lab technicians, evidence clerks and all other individuals supporting local, state, and federal law enforcement activities. As one might suspect, this lack of a prescribed method for establishing an evidentiary chain of custody results in great variability in terms of what procedures and protocols are practiced and observed at the various levels of law enforcement and prosecution. Naturally, documenting information about evidence such as item descriptions, date of receipts, case numbers, the name of the officer providing the evidence, and inventory numbers provide the foundation for establishing such a chain [38]. However, information of this nature does not encompass even minimal requirements to remove "reasonable doubt" that evidence is original, and not been intentionally or unintentionally tampered with or planted. The reader can probably recall several high profile cases when "smoking gun" type evidence was presented but the failure to properly administer or present the chain of custody rendered the evidence either inadmissable or not valid in the eyes of the jury. While one may assume most agencies have internal protocols for establishing chains of evidence, few agencies have public documentation of such protocols. Perhaps the most inclusive publicly available record is provided by the Minnesota Department of Corrections [14]. Their "Evidence Handling" policy describes the procedures for the collection, preservation, storage, and disposal of various forms of physical evidence.

2.1.3 Digital Chains of Evidence

As can be seen in the previous section, there are enormous challenges that stand in the way of developing standardized protocols for establishing a chain of physical evidence. Given the current state of technology, copying, fabricating, and deleting electronic data is easier than ever, and establishing a digital equivalent of a chain of evidence involves even more challenges. Additionally, due to the lack of standardization in traditional chains, we are provided with no "template" in which to base a foundation for a digital chain of evidence. Consistent with traditional chains of evidence, we see that the burden of proof once again falls on the presenter when dealing with electronic evidence. Berg states, "Chain of custody can be one of the most difficult issues faced by the forensic professional trying to introduce a digital image as evidence in a criminal case. If a defendant alleges an image has been altered, or could have been altered, the burden of proof falls upon the state to prove otherwise. If the image is a fingerprint linking the defendant to a crime scene, it is inevitable that the defense attorney will raise a question about the integrity of the image. In many cases, the success of the argument will hinge upon the procedures used to safeguard the security of the images [15]." Fortunately, if applied appropriately, existing technologies can potentially offer more efficient and accurate means to transmit, authenticate, and present electronic evidence.

Despite the need and potential advantages of developing such digital equivalents, it is apparent that little effort has been documented toward this end. Perhaps the most inclusive attempts at formalizing a digital chain of evidences are presented by Duerr et. al in [37] and Francoeur in [42]. Duerr's work proposes an information assurance approach to authenticating digital evidence. To that regard, it attempts to meet the five primary security services of access control, confidentiality, integrity, availability, and non-repudiation. While the work provides a high level overview of some of the problems associated with the task, it provides little detailed descriptions of how to actually go about providing these services. The main component it does touch on is integrity, explaining various aspects of traditional PKI and digital signatures. Francoeur's work focuses on the difficulties associated with establishing an electronic equivalent to a handwritten signature. Figure 2.1 shows his seven component diagram of a digital chain of evidence which touches on a number of aspects relevant to the proposed work. The work also provides useful information regarding the U.S. e-Sign Act which in part ensures that no signature or record will be deemed inadmissable in courts merely because of its electronic form. Additionally, it also provides information regarding the challenges associated with denial of signatures and the sequence in which a digital signature should be created. Similar to Duerr's work, we unfortunately see that while outlining many services a digital chain of evidence should provide, it does little to describe the mechanisms for actually providing them. Finally, while mentioning some of the challenges associated with including techniques such as watermarking in digital chains of evidence, Berg's work does not address frameworks, implementation details, or methods of performing the various aspects of validation to authenticate digital evidence. He does however, conclude his work with the passage "In the end, when an image is introduced in court, the first question that will need to be answered remains, "Is this image a fair and accurate representation of the scene or object as it was found?" Questions about chain of custody or the validity of specific computer enhancement techniques will ultimately be answered in accordance with the recognized scientific principles of the day." The proposed approaches and techniques presented

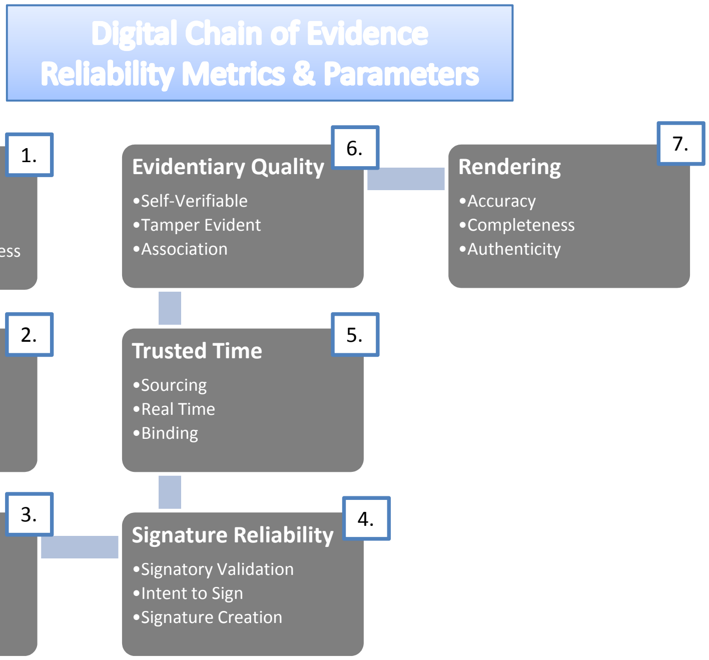


Figure 2.1: Francouer's Digital Chain of Evidence [42].

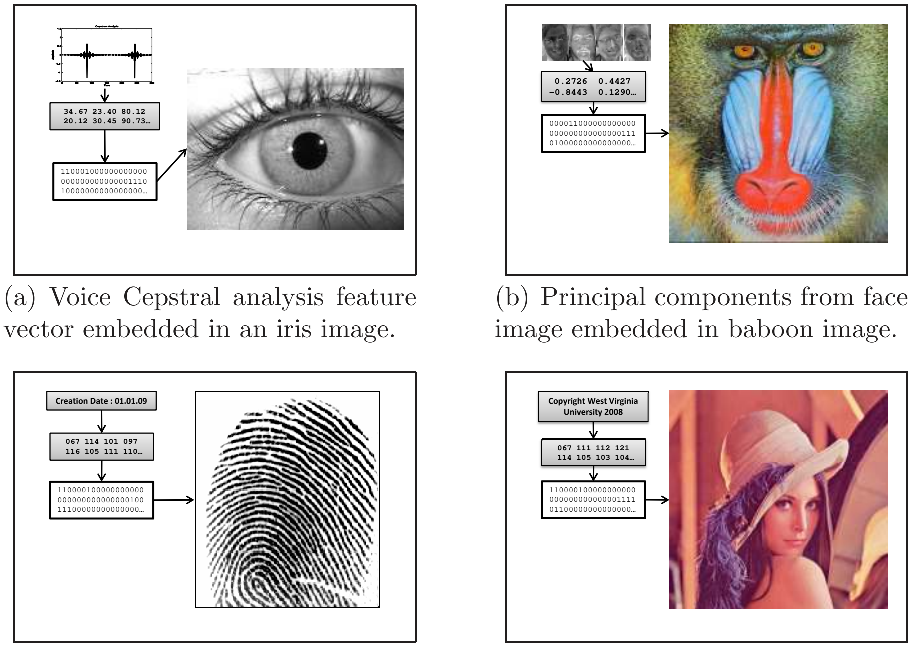
in the remaining chapters should help to define the "recognized scientific principles of the day."

2.2 Biometric Watermarking

Biometric watermarking is a specialized version of digital watermarking which typically involves imperceptibly embedding data in raw biometric host media for the purpose of providing additional security to a biometric system. Watermarks can range from non-biometric strings of random digits to system specific identifiers such as organization names and file creation dates to feature vectors from biometrics different than the host data. The additional security gained may result from using the watermark as a mechanism for proving file authenticity, tracking chain of custody and data reproduction, or to afford a multimodal biometric option. Biometric watermarking is typically used in tandem with cryptography and importantly provides a layer of security which remains intact after the decryption process. A biometric watermarking system should operate without significantly degrading the performance of the host biometric system it protects.

2.2.1 Introduction

First coined by Tirkel et. al [130], the term “digital watermark” originated in 1993. Unlike their physical predecessors (i.e. currency, copyright marks, etc.), digital watermarks are usually imperceptible to the human eye and require the use of machines to be detected in and extracted from the host media in which they are embedded. Although separate, digital watermarking is closely related to the field of steganography where secret messages are clandestinely embedded in larger, unrelated messages. Traditionally, digital watermarking has been employed as a means of copyright protection which allows an individual to prove (or disprove) ownership by embedding and extracting data suitable for verification of ownership. Additionally, digital watermarking can be applied to verify the authenticity of media, provide copy protection or reproduction management, and offer another mechanism for content description [149, 8]. Biometric watermarking is a specialized version of digital watermarking with the most noticeable difference being either the content of the watermark, the host data, or both. Figure 2.2 shows examples of the four main classes of digital watermarking as they relate to biometric watermarking. Although cases (a) and (c) are perhaps most popular where biometric data serves as the host, case (b) can also be considered biometric watermarking when biometric data is embedded in non-biometric host data.



(a) Voice Cepstral analysis feature vector embedded in an iris image.

(b) Principal components from face image embedded in baboon image.

(c) Image creation date embedded in fingerprint image.

(d) Copyright information embedded in Lena image.

Figure 2.2: The four classes of digital watermarking (a) Biometric watermark embedded in biometric host data (b) Biometric watermark embedded in non-biometric host data (c) Non-biometric watermark embedded in biometric host data (d) Non-biometric watermark embedded in non-biometric host data. Cases (a)-(c) represent biometric watermarking
. The images of Lena and the baboon are reproduced from the USC-SIPI Image Database

Besides the four main classes of watermarking, there are many types of biometric watermarking with differentiations in three main categories (visibility, blindess, and symmetry) as outlined in Table 2.1. The issue of visibility or perceptibility relates to whether or not the watermark is noticeable by humans either visually or audibly depending on the nature of the

Type	***Description***
Visible	The embedded information is noticeable by humans, either visually in pictures, or audibly in sound files.
Invisible	The embedded information is either completely imperceptible or not immediately noticeable by humans lacking the assistance of machines.
Public (blind)	The original host file is not required to detect / extract the embedded watermark.
Semi-blind	The embedded watermark is detected with additional information relative to the watermark encoding scheme but does not require the entire original host file.
Private (non-blind)	The embedded watermark can only be detected / extracted with both the watermarked image and the original host file.
Symmetric (Private Key)	A secret / private key is utilized to encode the watermark in the host image. This requires communication of the secret key between the sender and receiver.
Asymmetric (Public Key)	A public-private key pair are used to encode the watermark in the host file. The use of a public key pair prevents the need to communicate a secret key between sender and receiver as only the public portions of the key pairs need to be available. Optionally, this method can ensure image integrity and non repudiation of origin..

Table 2.1: Description of digital watermarking types applicable to biometric watermarking.

host data. The detection and extraction processes may or may not rely on the original host data or auxiliary data to complete, this notion is referred to as blindness. Finally, watermarking systems must make use of keys, either private, similar to symmetric cryptographic systems or public key pairs akin to asymmetric cryptosystems. A biometric watermarking system encompassing any combination these categories is imaginable but an invisible, blind, and asymmetric system is arguably the most difficult to conceive.

Beyond the scope of classes and categories, the quality of a biometric watermarking system can be evaluated by five types of characteristics: imperceptibility, robustness, fragility, capacity, and performability. Table 2.2 describes each of these five characteristics in depth. Although the characteristics of imperceptibility, robustness, fragility, and capacity will generalize to to digital watermarking, the characteristic of performability is specific to biometric watermarking.

2.2.2 Algorithms

Digital and biometric watermarking algorithms both operate on a similar framework which can be generalized into three operating modules, namely, an encoding, decoding, and authentication. Figure 3.1 outlines these modules, providing a generalization of the watermarking framework for both digital and biometric watermarking. The first operating module, the watermark encoder, embeds the watermark into the host data. The watermark can range from a random binary bit sequence to biometric eigen-face coefficients utilized for face recognition. Additionally, the encoding module may also utilize an embedding key / secret key which discerns specific embedding locations in the host data to be watermarked. The second operating module, the watermark decoder, takes as input the watermarked host data and processes it in order to extract the watermark. If an embedding key / secret key was used during the encoding module then the same key is required for the decoding process. Additionally, depending on the algorithm, original host data may also be required to extract the watermark. The last module, authentication, compares the recovered watermark to the original indicating the similarity between the two. In the event that the watermark is biometric in nature, the biometric data can optionally be passed on to its authentication system.

Characteristic	***Description***
Imperceptibility	The degree in which the host image is visibly altered or distorted due to the presence of the watermark. Watermarked images that bear no visible difference from their original host image are said to be imperceptible. Rarely, this characteristic may be evaluated beyond the scope of human perception. In these cases, the question is raised as to whether or not it is possible to reveal that an image contains a watermark through the aid of a machine or program (without access to the original host image).
Robustness	The ability of the watermark to be detected and extracted after the watermarked image has been subjected to any variety of transformations (i.e. compression, filters, affine transformations).
Fragility	The ability to detect any file transformation by way of the watermark. The detection might result from an inability to extract the watermark or from an extracted watermark that in not intact.
Capacity	The amount of information which can be embedded in the host data of a watermarking system. This is a function of the type and size of the host data that is being watermarked and the robustness of the watermarking system in terms of detectability and extractability.
Performability	The degree in which the watermark affects the performance of the biometric system(s) in question. At a minimum, biometric watermarking systems should not have a significant adverse affect on the performance of the biometric system(s) which they protect. Here performance can entail matching error rates, image quality, efficiency of computation time, etc. Some biometric watermarking schemes may positively affect biometric system performance.

Table 2.2: Characteristics of biometric watermarking systems

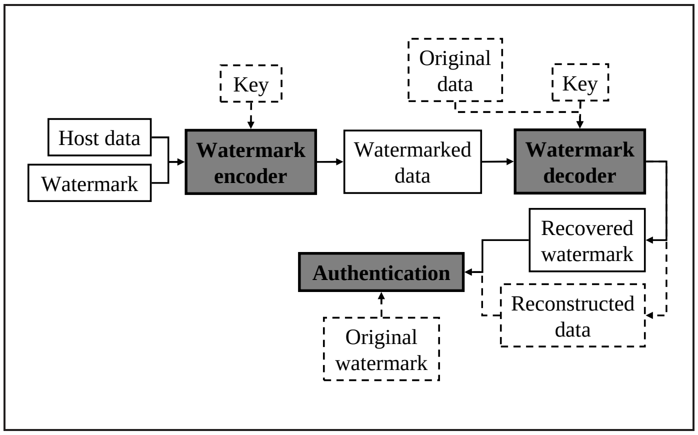


Figure 2.3: Generalized block diagram of a watermarking process. Shaded blocks indicate the main watermarking modules while dashed blocks / lines indicate optional areas of processing which are algorithm specific.

Watermark embedding and decoding techniques fall into two categories: spatial and transform domain techniques. Each category has specific advantages and disadvantages but in general spatial domain techniques are of lower complexity and more robust to biometric replacement attacks while transform domain techniques are of higher complexity but in addition are more robust to geometrical attacks such as rotation, scaling, and translation. In [63], an extension of the spatial domain watermarking technique known as amplitude modulation is utilized to hide a face image inside a fingerprint image. Eigen-face coefficients are watermarked into a fingerprint image at locations specified by a secret key. After decoding, both the reconstructed fingerprint and the face image are used for validation. They verify their experiments by demonstrating that the matching performance of the reconstructed fingerprint images is virtually unaffected by their watermarking system. Low et. al. in [85] , watermark a non-biometric host image with off-line handwritten signature in the form of a discretized bit string. They experiment and evaluate with three watermarking techniques:

least signification bit (LSB), CDMA spread spectrum in spatial, and CMDA spread spectrum in a transform domain such as discrete wavelet (DWT). Their experiments show that CMDA in the wavelet domain provides the most conclusive results with respect to jpeg compression and image quality. Similarly, Noore et. al. [102], utilize the DWT to watermark fingerprint images with face and demographic text data. They validate their experiments by illustrating that the matching performance of the watermarked fingerprint is unaffected by the watermark and demonstrate that their watermarking system is tolerant to various degradations such as cropping rotation, jpeg compression, noise, and median filtering.

2.2.3 Application Scenarios and Attacks

Depending on the intended use, biometric watermarking systems are subject to a series of application scenarios and attacks. Application scenarios can be thought of as normal usage patterns that a watermarking system should realistically be expected to withstand without serious side effects on the performance of any of the characteristics outlined in Table 2.2. Examples of application scenarios can include, but are not limited to: database (re)compression, partial progressive decoding, and noisy channel transmission. Each of these scenarios can have an effect on one or more characteristics of a biometric watermarking system. For instance, a highly compressed watermarked image may lead to difficulties in the watermark extraction process as a compression algorithm often significantly alters an image, which in turn alters the watermark itself. Occasionally operational environments result in slow data transmission speeds which may force a system to progressively decode portions of an image as it becomes available. This type of application scenario can have an effect on the robustness of the extraction process, the performance of the biometric system(s) in question, and potentially the imperceptibility of the watermark. Many techniques are capable of tolerating scenarios similar to these. That said, the expected application environment should be considered when attempting to appropriately choose a watermarking technique as the current state of the technology finds a given technique may be well suited for one application scenario but not perform well in another

Perhaps the most notable difference between the general field of digital watermarking and biometric watermarking is its relationship to the characteristic of performability. For obvious reasons, a biometric watermarking system must minimize the effect it has on the biometric system it protects. Issues such as matching performance, image quality, computational efficiency, and even legal repercussions must not be ignored. Regarding matching performance, a biometric watermarking system should not impede the main block of the biometric system(s) in question, specifically the feature extraction and matching blocks. It should be noted that this effect could potentially propagate itself in two ways. Perhaps the most obvious effect is when the host data is used in a biometric system; here the presence of the watermark may impede the feature extraction process as the watermark may add noise to the image. Naturally this may lead to inaccuracies in the matching block. A less obvious effect is when a biometric feature vector serves as a watermark and is also used as part of a biometric system. In this scenario, accurate extraction of the watermark is of utmost importance as small changes in the values of feature vectors can lead to significant changes in authentication results. Little or no work exists studying the latter of the two effects.

Application scenarios aside, biometric watermarking systems must also deal with attacks or malicious attempts to subvert a system. Such attacks may involve removal, alteration, or replacement of the embedded watermark found in an image. Although some of these may also fall in the application scenario domain, examples of attacks include: rotation, scaling, translation, cropping, masking, and (re)watermarking. Similar to application scenarios, different biometric watermarking techniques can handle different attacks with varying degrees of success. Although not specific to biometric watermarking, Zheng et. al provide an excellent breakdown of so-called RST (rotation, scaling, translation) invariant watermarking algorithms in [149]. Often the ability to handle a given attack lies in the domain in which a biometric watermarking technique operates. For instance, rotation attacks are handled with greater ease by watermarking techniques which operate in tranform domains (i.e. Fourier, DCT, wavelet, etc.) This type of attack is arguably more difficult to handle in techniques that fall in the spatial domain. Conversely, the spatial domain is more akin to dealing with a biometric replacement attacks (i.e. replacing the watermarked iris or face region of an image) as the biometric ROI is known in the spatial domain but more difficult to localize in a transform domain. Little work has been done addressing attacks specific to biometric watermarking systems; most work focuses only on attacks to the general area of digital watermarking but it should be noted that many such specific attacks exist.

2.2.4 Patents, Tools, and Commercial Products

Searching the United States Patent Office for “Biometric Watermarking” yields well over 100 entries of varying relevance to the field. The most notable is entitled “Biometric Watermarks” and was issued in 2001 to GTE Service Corporation [98]. This patent outlines the general schematic for a biometric watermarking system. A more recent patent can be found in USPO 7,305,089 issued to Canon in 2007 which includes an watermarking system embedded in a camera with the intended purpose of associating a photographers biometric information with images taken by the camera [96]. At least two freely available tools related to digital watermarking and biometric watermarking exist. Stirmark Benchmark 4.0 is a software tool designed to perform robustness testing of image watermarking algorithms [113]. Checkmark also provides a bed of attacks to evaluate the robustness of a watermarking system [111]. Many commercial entities offer a broad range of digital watermarking solutions that can potentially fall under the category of biometric watermarking. Perhaps the most widely known of such companies is DigiMarc Corporation based Oregon, US.

2.3 Digital Hardware Fingerprinting

Falling within the field of image forensics, digital hardware fingerprinting is the process of identifying the source hardware used to capture an image. More often than not, this identification process is independent of the scenery or primary image content but rather based on some other feature of the image. Such a process is has a number of applications including verifying authenticity and integrity of images, enforcing copyright protection, establishing ownership of data, tracing the origin of data, and establishing a chain of evidence in criminal cases or other arenas.

2.3.1 Introduction

Digital hardware fingerprinting deals with the investigation of techniques used to identify the unique characteristics of devices that capture images (e.g., digital cameras, camcorders, and scanners, biometric devices, etc.) [121]. These unique characteristics can then be used to identify the source used to capture an image which can then be subsequently used to verify image authenticity, integrity, etc. In this sense, “source” can encompass any number of ideas. Take for instance the most commonly used example of digital photography, here the “source” is a camera and the hardware fingerprinting system is attempting to identify which camera captured the image. In this simple example “source” identification can take place at four different levels. Figure 2.4 shows four levels that may be considered in a source hardware identification problem: technology, brand, model, and unit. Based on the four levels, various questions can be answered:

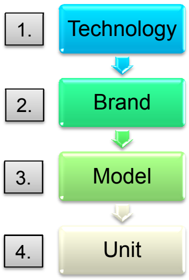


Figure 2.4: Source Hardware Identification Levels.

- Was the image in question captured from a sensor relying on technology X or technology Y? (Technology)
- Was the image in question captured from a device manufactured by vendor X or vendor Y? (Brand)
- Was the image in question captured from a device corresponding to model X or model Y manufactured by vendor Z? (Model)
- Was the image in question captured from a unit A or unit B of model X manufactured by vendor Z? (Unit)

Naturally, unique challenges exist to performing source model identification at the different levels. Furthermore, the choice of which feature to use may be in part dependent on the level at which a system must operate. The overwhelming majority of research on digital image forensics deals specifically with images captured from photographic cameras and document scanners.

2.3.2 Features and Classification Techniques

There are many proposed approaches to performing source hardware identification at different levels that are well summarized in a surveys of the field by Sencar et. al in [121] and Khanna et al. in [72]. To facilitate a better understanding of the different approaches, this section is broken down at a high level based on the features used for classification as is presented in [121].

CFA and Demosaicing Artifacts

The most sensors found in digital cameras are based on concept of a Color Filter Array (CFA) composed of a surface of pixels that each have their own spectrally sensitive filter [6]. Depending on the choice of CFA, an appropriate demosaicing algorithm must be chosen to correctly render high spatial frequency image details [121]. These demosaicing algorithms introduce an interpolation effect which produces unique correlations between color values of image pixels. These correlations, or demosaicing artifacts, can in turn be used to identify source camera models [121]. Approaches utilizing these artifacts have been proposed with varying success by Popescu [115], Bayram et. al [13], Long et. al. [84], and Swaminathan et. al. [127] with the highest classification accuracy of 95% on four camera models achieved by Long et. al's PCA based approach.

Lens Distortions

Lens distortions result from changing image magnification which in turn changes the distance from the optical axis [121]. Choi et al. proposed an approach based on lens distortions to distinguish between three camera models in [28]. This work uses various parameters from algorithms that attempt to compensate for such distortions as features for identification. The proposed approach involving a radial symmetric distortion model achieves a 91% accuracy.

Sensor Dust Characteristics

Although not inherent to the capture devices, the notion of sensor dust characteristics has been explored by Dirik et al. in [35]. In the work, the authors examine digital single-lens reflex (DSLR) cameras which typically acquire dust occluding the sensor when lens are changed. The approach uses a technique based on match filtering and contour analysis [121]. While the experimental results claim an accuracy of 92% in correctly identifying a single camera's images from a bed of images taken from other cameras, the approach may run into challenges as cameras increasingly embed corrective algorithms to compensate for dust [121].

Imaging Sensor Imperfections

Arguably the most attractive approaches to camera source identification are based on imaging sensor imperfections. Due to small imperfections in the manufacturing process, whether from the presence of hot or cold pixels or pattern noise, each sensor typically has a unique signature which manifests itself in all images it captures. Geradts et al. proposed a technique based on the presence of dead pixels in [46]. Perhaps the most promising approach to date was proposed by Lukas et al. in [86]. Based on measuring pixel nonuniformity (PNU) noise, the approach analyzes the differences in images resulting from imaging sensor imperfections [86]. This approach uses a wavelet denoising algorithm to model the photo-response nonuniformity (PRNU) component of the PNU which in turn serves as a reference template for each camera. Then, a correlation algorithm is used to match the noise patterns from images to be identified to the reference templates. The approach achieves 100% accuracy on a 9 camera set which does include some duplicate instances of camera models. Sutcu et. al [126] tested the technique on a larger database of images arriving at more realistic ($< 100\%$) performance rates. Fridrich et. al extends the approach in [26] by applying preprocessing techniques to the noise extraction algorithm. Finally, Khanna et. al [70], and Guo et. al [49] also proposed approaches using sensor pattern noise to fingerprint flatbed scanners instead of digital cameras.

Not falling specifically in any of the four previously mentioned groups is a somewhat general approach proposed by Kharrazi et al. which outlines 34 image features including average pixel values, RGB pair correlations, and neighbor distribution center of mass [73]. The features are then input into multi-class classifiers and achieve an accuracy of 97% on a four camera database.

2.4 Keystroke Dynamics

Keystroke recognition is a behavioral biometric which utilizes the unique manner in which a person types to verify the identity of an individual. Typing patterns are predominantly extracted from computer keyboards, but the information can potentially be gathered from any input device having traditional keys with tactile response (i.e. cellular phones, PDA's, etc). Although other measurements are conceivable, patterns used in keystroke dynamics are derived mainly from the two events that make up a keystroke: the Key-Down and Key-Up. The Key-Down event takes place at the initial depression of a key and the Key-Up occurs at the subsequent release of that key. Various unique features are then calculated based on the intra-key and inter-key timing variations between these events. After feature extraction, a wide range of algorithms can be employed to establish whether the unique pattern confirms or denies the claimed identity.

2.4.1 Introduction

The earliest form of keystroke recognition emerged in the early 1900s during the days of WWI. During the war, the French used listening posts in which operators were able to recognize the "fist" of enemy radio operators communicating in Morse code. These trained individuals would learn to recognize operators by differing lengths of pauses, dots and slashes, and varying transmission speeds. This intelligence subsequently allowed the French to establish the identity of entities such as enemy battalions. Far more sophisticated than electromechanical telegraphs used to transmit Morse code, keyboards of today offer many more opportunities to establish the unique manner in which one types. Intuitively, coarse level differentiation can be achieved by investigating typing speeds. For instance, a professional typist who averages 90 or more words per minute would be easily distinguished from a "hunt and peck" amateur who averages only 20-25 words per minute. That said, this feature only goes so far as many people type at similar speeds and the average speed that an individual types can vary significantly depending on many factors. The time it takes an individual to locate a key (sometimes referred to as “seek-time”) also varies from key to key. For instance, left-handed individuals may have quicker seek-times for keys on the left side of the keyboard and vice versa [95]. Along those same lines, use of the shift keys to modify characters can also vary from individual based on handedness and typing skill. Trained professionals will always modify characters on the right side of the keyboard with the left shift key while amateurs may continually use the right shift key to do so [12]. Language undoubtedly plays a large role in the individuality of a typing signature. Given that a person speaks English, commonly used words like {the, and, you, are} are often “programmed” in one’s mind and typed quickly as opposed to an individual of a different native language. Additionally, individuals typically exhibit a consistent pattern of errors including replacements, reversals, and extraneous hits. In an extreme case, the consistent lack of errors is a pattern in itself.

2.4.2 Keyboard Technology and Semantics

There are four different kinds of switch technology used in keyboards today; pure mechanical, foam element, rubber dome, and membrane [97]. Each switch type has various characteristics such as feel, durability, price, etc. No matter the key switch technology chosen, when a key is depressed, a degree of “bounce” is present. Bounce can be defined as the effect when the contact device rapidly engages and disengages over an extremely short period of time [97]. Keyboards, either external to desktop PCs or internal to laptops and other devices are computers in their own right as they contain a microprocessor, RAM, and sometimes ROM. Using their processors and controllers, they filter out the difference between bounce and two successive keystrokes. Each stroke therefore consists of two events, when the plates are engaged and when the engagement is released or disengaged. Scan codes resulting from these events are sent from the controller in the keyboard to the event handler in the BIOS of the device in question (usually a PC) [97]. Scan codes are recorded by the processor based on a matrix composed of all the keys on the keyboard. The keyboard matrix operates on a buffer that allows for the processing of simultaneous keystroke events. As mentioned before, when a key is pressed down, the plates become engaged. It is at this point that the keyboard processor sends a "make code" encoded as a hex value to the device. The make code can be thought of as including both the key engaged and various other state flags indicating if / how the key was modified by any of the various control keys such as shift, alt, etc. Once the key disengages, a corresponding "break code" is sent to the PC [97]. These ideas form the basis of keyboard technology at its lowest-level.

Using this background as a foundation, the upper level semantics of keyboard operation can be defined. The basis of all features included in keystroke recognition is founded on the keystroke event and the associated make code / break code correlation described previously. Instead of dealing with terms like "make code," "disengagement," etc., researchers usually yield to the more intuitive, higher level definitions below.

1. **Key-Down**- The event that fires when a key is pressed down. This corresponds to the event of the keyboard processor sending the device (usually a PC) a "make code." It should be noted that this event will continually fire until the key being depressed is released. The speed at which the Key-Down event fires while a key is depressed is referred as the "repeat rate." This is a user customizable property in virtually all operating systems.

2. **Key-Up**- The event that fires when a currently depressed key is subsequently released.

3. **Keystroke**- The combination of an initial Key-Down event and the corresponding Key-Up event.

4. **Hold Time**- The length of time between an initial Key-Down event and the corresponding Key-Up event. Hold time is sometimes referred to as "dwell time."

5. **Delay**- The length of time between two successive keystrokes. It should be noted that this time can be positive or negative (overlapping strokes). Some works refer to delay as "latency" or "flight."

Some highly specialized keyboards can record other information such as the pressure of key strikes, but the foundation of the technology is based on the events defined above.

2.4.3 Feature Representation and Classification

A wide variety of algorithmic approaches have been explored as suitable candidates for the task of keystroke recognition. The problem of keystroke recognition fits well within the general fields of pattern recognition and machine learning; the two main tasks involved in solving problems within these fields are to define the representation of the feature space and the algorithm used to predict the class of samples. As mentioned in previous sections, the features in keystroke recognition are primarily derived from the elements that make up a keystroke. Most algorithms utilize raw times or first order statistics such as minimum, maximum, mean, median, and standard deviation of hold times and latencies [12, 44, 66, 104, 104, 94] for feature representation. Here, hold times are for individual keys whereas latencies are measured between two keystrokes often defined as “digraphs.” Using these statistics, one can either calculate fixed length feature vectors as outlined in [12] or variable length feature vectors as outline in [16]. Fixed length or static size feature vectors will always have a predetermined length despite the length of the input sequence. The size of variable length or dynamic feature vectors will depend on the size of the input sequence. Although the vast majority of keystroke recognition systems rely on single key hold times and digraph latencies, some approaches define other feature sets including trigraph durations, ordering of keystrokes (when shift-key modification is required), etc. [16].

Beyond feature representation, a keystroke recognition system must employ an algorithm to predict the class of incoming samples. In general, the approaches can be broken down into two sections: distance metric based approaches and machine learning approaches. After calculating the feature vector for an incoming sample, the chosen algorithm must predict the class of the sample (genuine or imposter). Many approaches will do so by comparing the incoming sample to one or more reference samples in a template database through a distance metric. Popular distance metrics include: Euclidean, Mahalanobis, Manhattan, Chebyshev, and Hamming. When distance metrics are employed to compare two samples, the smaller the score the closer the two samples are to each other. Gaines and Lisowski [44], Garcia [45], Young and Hammon [145], and Joyce and Gupta [66] are all examples of algorithms that utilize one or more of these distance metrics as classification schemes. Table 2.3 provides an overview of selected work in keystroke recognition including the works listed above. The table includes the features / algorithm used, input requirements, the scope, and performance. Under the performance column the raw totals in terms of FAR and FRR are presented within parentheses when listed in the work.

Table 2.3: Overview of Selected Works in Keystroke Recognition

Work	Feature(s) / Algorithm	Input	Scope	Performance
Gaines & Lisowski (1980) [44]	Latency between 87 lowercase digraphs using sample t-tests	300-400 word passage 2 times	7 secretaries	FAR 0% (0/55) FRR 4% (2/55)
Garcia (1986) [45]	Latency between 87 lowercase digraphs and space key & Complex Discrimination using Mahalanobis distance function	Individual's name & 1000 common words 10 times each	(N/A)	FAR 0.01% (N/A) FRR 50% (N/A)
Young & Hammon (1989) [145]	Plurality of features including: digraph latencies, time to enter selected number of keystrokes and common words using Euclidean distance	(N/A)	(N/A)	(N/A)
Joyce & Gupta (1990) [66]	Digraph latencies between reference strings using mean and standard deviation of latency distance vectors	Username, password, first name, last name 8 times each	33 users of varying ability	FAR 0.25% (2/810) FRR 16.36% (27/165)
Brown & Rogers (1993) [22]	Latencies and Hold Times using Euclidean distance and neural networks	Usernames, 15-16 character avg. ≈ 1,000 sequences tested	21 & 25 users	FAR 4.2%-11.5% (N/A) FRR (N/A)
Obaidat & Macchiarolo (1993) [104]	Digraph latencies between reference strings using Neural Networks	15 character phrase 20 times each	6 users	97% overall accuracy
Obaidat & Sadoun (1997) [105]	Digraph latencies and key hold times using multiple machine learning algorithms	Username 225 times / day for 8 weeks	15 users	FAR 0% (N/A) FRR 0% (N/A)
Monrose & Rubin (1997) [95]	Latencies and Durations with Normalized Euclidean distance & weighted/non-weighted maximum probability	Passages of text over 7 weeks	(N/A)	Identification Framework
Maisuria & Ong & Lai (1999) [91]	Digraph latencies with neural networks (multi-layer perceptron)	passwords 60 times over 3 periods	20 users	FAR ≈ 30% (N/A) FRR ≈ 15% (N/A)
Monrose, Weiter, & Wetzel (2001) [94]	Digraph latencies and key hold times, algorithm employed is unclear	8 character password	20 users	FAR % (N/A) FRR 45% (N/A)
Bergadano, Gunetti, & Picardi (2002) [16]	Trigraph duration using degree of disorder	683 character text 5 times	44 users	FAR 0.04% (1/10,000) FRR 4% (N/A)
Yu & Cho (2004) [146]	GA-SVM's and wrapper FSS on hold times and digraph intervals	6-10 character passwords 150-400 gen / user & 75 imp	21 users	FAR 0% (N/A) FRR 3.69% (N/A)
Bartlow & Cukic (2006) [12]	Random Forests on digraph latencies and hold times digraph latencies	usernames + 8 & 12 char passwords ≈ 9,000 sequences	41 users	FAR 2% (N/A) FRR 2% (N/A)
Sung & Cho (2006) [125]	GA-SVM's and wrapper FSS on hold times and digraph intervals	6-10 character passwords 150-400 / user & 75 imposter	21 users	FAR 3.85% (N/A) FRR 13.10% (N/A)

As the field has matured, many other machine learning approaches have emerged as viable solutions for prediction mechanisms in keystroke recognition. Neural networks have widely been employed with works by Obaidat et. al [104, 105], Brown et. al [22], and Maisuria et. al [91]. Cho and Yu have applied Support Vector Machines (SVM's) to the problem extensively [146, 125]. Additionally, Bartlow and Cukic explored the decision tree approach of Random Forests [12] (see Table 2.3 for more information on listed works).

2.4.4 Applications and Challenges

In application, the uses of keystroke recognition can range anywhere from stand-alone biometric systems to augmenting general computer security systems. Depending on various system specific security characteristics such as database size and operational risks, keystroke recognition is suitable as a stand-alone biometric. Although not on the level of physiological biometrics such as iris, fingerprint, and face, many works in the literature indicate that the attainable performance rates are within the scope of what some operational profiles would require. Much like the physiological biometrics, performance is typically measured by conventional error measures such as False Accept Rate (FAR), False Reject Rate (FRR), and Equal Error Rate (EER). In terms of EER, many of the previously cited works achieve performance $\leq 5\%$ (see Table 2.3). Naturally, FAR and FRR's can be tailored based on where one wishes to fall on a traditional Receiver Operating Characteristic (ROC) curve. It is important to note that the literature has not firmly established whether the technology is sufficient for biometric systems operating in identification mode as the focus of past research is almost exclusively tailored to verification based systems. It is also important to note the trend of decreasing data requirements as earlier works required extremely long passages of text whereas most recent works require only usernames, passwords, or both. Related to this trend, keystroke dynamics need not be applied only at the time of login, which may lead to time-of-check-time-of-use vulnerabilities. Instead, they can be applied transparently throughout the span of a period of use. This feature can allow systems to continually check for the presence of insider threat where an authorized user may login to a system and subsequently allow an unauthorized user access. If a system does not require a continual verification environment, keystroke recognition is also very suitable for a challenge-response type framework where the user is periodically authenticated.

Besides stand-alone biometric systems, keystroke recognition can be used as an augment to traditional username / password systems. This process is often called credential hardening or password hardening. Monrose et. al first proposed the idea in [94] and Bartlow et. al also explored the concept in [12]. Both works show how the addition of keystroke recognition to traditional authentication mechanisms can drastically reduce the penetration rate of these systems. Works of this nature may also bode well in online authentication environments such as banking and e-commerce websites which are now commonly requiring secondary verification layers.

Either as a stand-alone biometric or an augment to a traditional username / password scheme, keystroke dynamics are arguably more cancellable or replaceable than physiological biometrics. The idea of cancellable biometrics touches on the fact that the threat of biometric compromise exists and is often realized. With fingerprint, face, iris, etc., it is often difficult to reissue a biometric authentication mechanism as fingers, faces, and irises are not easily removed and replaced in humans. In keystroke recognition however, the behavior which induces the biometric can be changed. In other words, if a user's keystroke recognition template is compromised, the data in which the template is based (i.e. password / passphrase) can simply be changed which will result in a new biometric template. For obvious reasons, this is seen as a very attractive feature of keystroke recognition.

Beyond the scope of academic research, many patents have been issued in the field including: Garcia (4,621,334 - 1986) [45], Young & Hammon (4,805,222 - 1989) [145], Brown & Rogers (5,557,686 - 1996), and Bender & Postley (7,206,938 - 2007). In addition to patents, there are many commercial offerings of keystroke recognition systems. Two popular systems are BioPassword ©(http://www.biopassword.com/) and iMagic Software © (http://www.imagicsoftware.com). Systems such as these are attractive as the overhead of keystroke recognition in terms of hardware deployment and seamless integration into currently existing authentication systems is typically much less than that associated with physiological biometrics such as fingerprint, iris, and face.

Despite the maturity of the field over the last 30 years, there are still many challenges that have yet to be solved. Three main challenges are associated with the data required to train keystroke recognition systems. First, few works have formally set out to determine the amount of sequences required to sufficiently establish a typing signature ready for operational deployment. For a system to be deployable, it must have a realistic training requirement that users are willing to incur. It seems that repeatedly typing a username and password combination 50 or more times would be unacceptable in the eyes of most users, yet 5 may be insufficient in terms of meeting established security goals. Second, as passwords need to be replaced / reissued, the problem of retraining needs to be addressed. Once again, these retraining requirements have yet to be firmly established. Third, the behavioral nature of this keystroke recognition requires a slightly more involved data collection process than what is typical in conventional physiological biometric systems. Most notably, one cannot simply compare genuine input of one user to genuine input from another user in order to establish an instance of imposter input as the data is often different for every user (i.e. usernames / passwords). As a result, most academic research will have users type the credentials or data associated with other users to arrive at imposter sequences for training. Clearly this is not feasible in operational systems as passwords are frequently reset. Therefore, the issue of automatic generation of imposter data is an area that needs to be explored.

Chapter 3

Biometric Watermarking

3.1 Introduction

The threat associated with identity theft in a networked society becomes even more formidable as biometric systems become commonplace as mechanisms for identity management. In order to maintain long schematic life cycles and desired levels of interoperability, systems must often store raw biometric images in addition to templates. Furthermore, despite the push to develop international standards for templates in various biometric modalities, one can assume raw images will continue to be stored in many systems for reasons of insurance and support of legacy systems. Beyond these reasons, original digital images must be captured and stored when collecting evidence for criminal justice, military operations, and some civilian applications.

Whether in an effort to protect the identity of individuals or to ensure the integrity of a data in a chain of evidence, state of the art cryptographic protocols must be employed to protect the data of individuals enrolled in such systems. Furthermore, even with the available technologies for protecting such IT systems, no system is impenetrable. Therefore, protection mechanisms must be in place even after decryption and replaceability of authentication credentials must be available regardless of the biometric nature of the data [61]. The work outlined in this subsection includes a framework that combines biometric watermarking and public key cryptography to address the previously mentioned challenges. Using voice feature descriptors to watermark raw iris images, the proposed system offers multiple levels of authentication through a potentially multimodal biometric system while also offering authentication, integrity, and non-repudiation of origin through asymmetric cryptography. Through watermarking, the scheme offers another degree of protection in terms of tracking the origin of data, adds another layer of authentication, and improves the degree of biometric replaceability by encoding a behavioral biometric into the raw image. By varying input parameters of our encoder, our watermarks can offer many degrees of robustness while leaving performance of both the primary (iris) and secondary (voice) biometrics virtually unaffected. In the work, we test the effect of our watermarking scheme on iris image quality and scoring performance. We test the ability to extract our watermarks related to three real-life application scenarios: database compression, partial data loss (progressive decoding), and data transmission over a noisy channel. Additionally, we investigate the notion of rewatermarking as it relates to the chain of evidence framework presented in Chapter 6.

3.2 Watermark Encoding and Decoding

Amplitude Modulation is a spatial domain watermarking technique originating from [76]. This is one of the more widely studied modulation techniques in the field of communication and signal processing. Presented below is a brief overview of such encoding and decoding techniques as was discussed in [76], followed by how it is modified to suit biometric needs.

3.2.1 Encoding

In [76], bits are embedded multiple times by modifying pixel values, B_{ij}, in the blue channel. These modifications are either additive or subtractive, depending on the value of the bit, s, and its proportionality to the luminance, L_{ij}, as seen in Equation 3.1

$$B_{ij} \leftarrow B_{ij} + (2s-1)L_{ij}q \begin{cases} B_{ij} \in \text{embedding locations} \\ s \ \in \ \text{bit} \\ L_{ij} \in \ \text{Luminance} \\ q \in \ \text{Encoding strength} \end{cases} \tag{3.1}$$

3.2.2 Decoding

The decoding process estimates a linear combination of the pixels in a cross-shaped neighborhood around the encoded bit as seen in Equation 3.2

$$B_{ij} = \frac{1}{4c}(\sum_{k=-c}^{c} B_{i+k,j} + \sum_{k=-c}^{c} B_{i,j+k} - 2B_{ij}) \{ \ c \in \text{neighbor size} \tag{3.2}$$

After decoding and arriving at an estimated pixel B$_{ij}$ value, the difference between the estimated and watermarked pixels is averaged over all embedding locations for that bit. Finally the sign of this value indicates the bit (if positive =1, if negative = 0). However, to attenuate robustness to compression, cropping, and affine transformations, an adaptive thresholding method is introduced: append 2 bits to every bit stream that are always set to {0, 1} respectively as seen in Equation 3.3

$$bit = \begin{cases} 1\ \delta^b > \frac{\delta^0+\delta^1}{2} \\ 0 \text{ otherwise} \end{cases} \begin{cases} \delta^0 \text{ average diff of all 0 reference bits} \\ \delta^1 \text{ average diff of all 1 reference bits} \\ \delta^b \text{average diff of current bit} \end{cases} \tag{3.3}$$

3.2.3 Adaptation to Biometric Data

Biometric image medium is usually captured in grayscale. Because of this, the encoding process has to be slightly modified to take this into consideration. For example, [61] modifies the encoding equation to take in local image information such as gradient, P_{GM}, and standard deviation, P_{SD}, of the cross-shaped neighborhood to adjust watermarking strength. Parameters A and B aid in adjusting the strength of the standard deviation and gradient when modulating the bits to be encoded. The following equation represents this adaptation.

$$\begin{gathered} P_{WM}(i,j) = P(i,j) + \\ (2s-1)P_{AV}(i,j)q\left(1+\frac{P_{SD}(i,j)}{A}\right)\left(1+\frac{P_{GM}(i,j)}{B}\right) \end{gathered} \tag{3.4}$$

P$_{AV}$ represents the average pixels in a 5x5 neighborhood centered around i,j. Certain considerations need to be addressed in terms of encoding watermarks in the biometric region of interest (ROI) in the host image; namely, the degree in which encoding the watermark in the ROI affects matching performance of the biometric.

3.3 Framework Design

In this subsection we briefly describe the overall framework of the proposed system. It is generalized such that it could be applied to various biometric authentication environments such as internal or external web-based identity management and point-of-entry (POE) applications. Figure 3.1 shows all phases of a potential verification process. Beginning with a request by the user (Alice) to access to some resource protected by the system (Bob), Alice supplies her authentication data, specifically an iris image watermarked with a voice feature descriptor. This data could be stored electronically in a web-based profile or physically on a secure token. The most important aspect of this data is that is it will have been previously encrypted using public key infrastructure (PKI). More specifically, the data will first be encrypted with Bob's public key and then signed with Alice's private key. For an explanation of PKI see [59]. At this point, the system would process the user's ID and encrypted data. Bob's first step involves decrypting the data using Alice's public key and his private key. Next, the system passes the authenticated and decrypted iris image through the watermark decoder. The extraction of the voice descriptor is then used to further verify the authenticity of the image. Finally, the system can proceed with biometric verification using the iris image or optionally conduct multimodal verification using both the iris image and the voice feature descriptor. Upon conclusion of the process, the system has provided multiple levels of authentication; through cryptography, after decryption using watermark verification, and potentially with multimodal biometric verification. The system provides data integrity through cryptography, and non-repudiation of origin through PKI. Additionally, the watermarking provides a level of tracking cryptography alone cannot provide as the

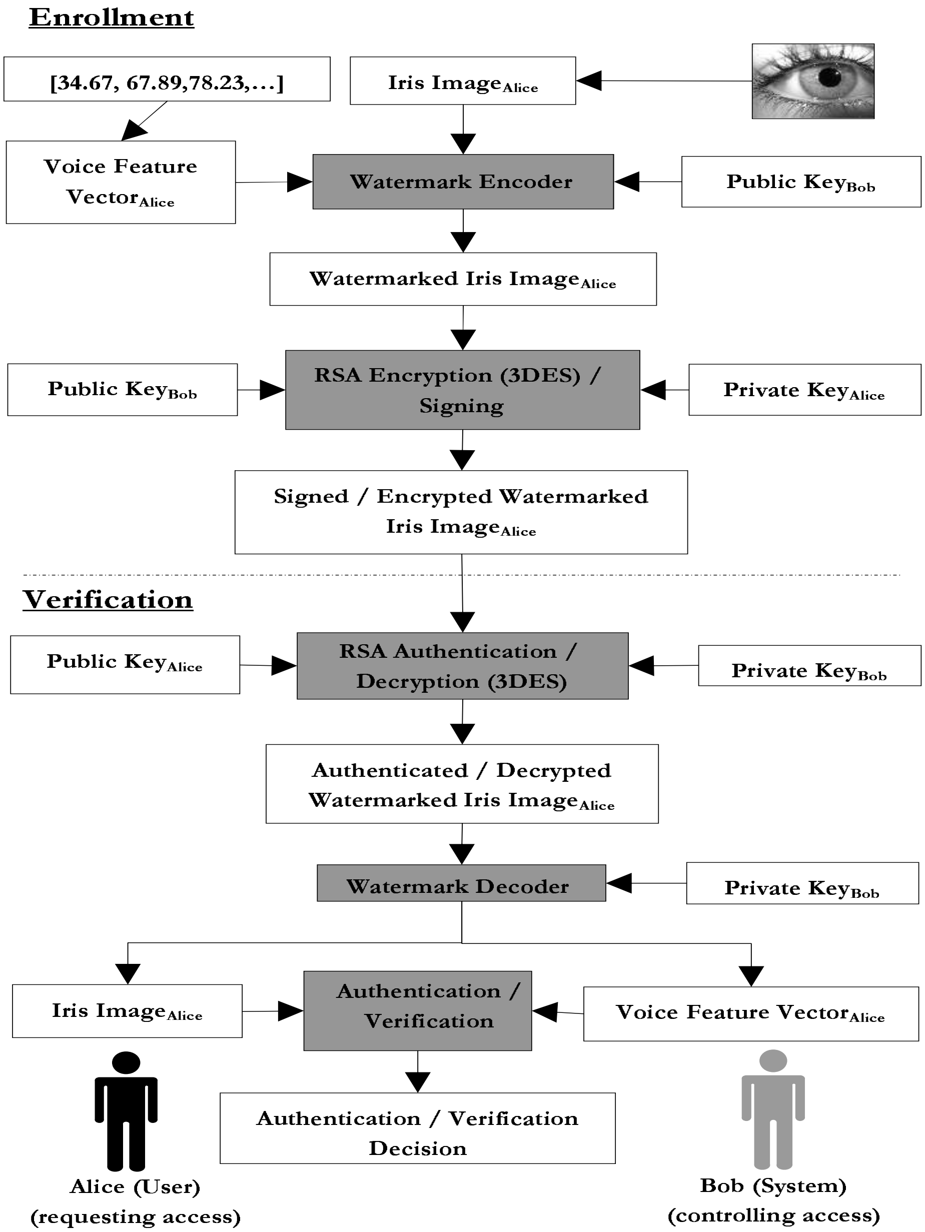

Enrollment
[34.67, 67.89,78.23,...]
Iris Image$_{Alice}$
Voice Feature Vector$_{Alice}$
Watermark Encoder
Public Key$_{Bob}$
Watermarked Iris Image$_{Alice}$
Public Key$_{Bob}$
RSA Encryption (3DES) / Signing
Private Key$_{Alice}$
Signed / Encrypted Watermarked Iris Image$_{Alice}$
Verification
Public Key$_{Alice}$
RSA Authentication / Decryption (3DES)
Private Key$_{Bob}$
Authenticated / Decrypted Watermarked Iris Image$_{Alice}$
Watermark Decoder
Private Key$_{Bob}$
Iris Image$_{Alice}$
Authentication / Verification
Voice Feature Vector$_{Alice}$
Authentication / Verification Decision
Alice (User) (requesting access)
Bob (System) (controlling access)

Figure 3.1: System Framework - Enrollment and Verification

decrypted images will still have watermarks that indicate the origin of an image. If necessary, the system could also include file hashing techniques to add another measure of data integrity and watermark fragility.

3.4 Experimental Design

We chose to test our watermarking scheme on a subset of images from the WVU biometric database. One hundred iris classes were selected, with two images per class. Images were watermarked with randomly generated 64 bit binary sequences. The length of the sequence was based on the work done by Monrose, Reiter, and Wetzel in which voice was used as a seed to generate cryptographic private keys. Using 60 bit sequence, they were able to reliably reconstruct keys. The 60 bit sequence represented a passphrase of approximately 10 words, this was assumed to be more than adequate length for the proposed system [93].

Our amplitude modulation technique involves three parameters for watermarking iris images; encoding strength, the number of times the watermark sequence is encoded or repeated, and the percentage of watermark encoding that take place in the iris itself. Encoding strength represents the degree in which the intensity of a bit is modulated from it's orginal value. Number of times encoded represents the redundancy of the watermark encoding (the larger the number the more times the same watermark is encoded in a host image). Finally, encoding location represents what portion of the watermark is encoded in the iris region of the host image. The following list represents the parameter variation scheme we chose to analyze:

- **Encoding Strength-** (0.1, 0.06, 0.04)
- **Number of Times Encoded-** (60, 40, 20)
- **Encoding Location-** (67%, 33%, 0%) (in iris)

Therefore, each of the 200 iris images were watermarked a total of 27 ways (3 X 3 X 3) for a grand total of 5,400 watermarked images. The first two parameter sets were arrived through preliminary extraction tests. The third set of values were chosen as a trade-off between minimally affecting the iris recognition system and keeping the images robust to tampering.

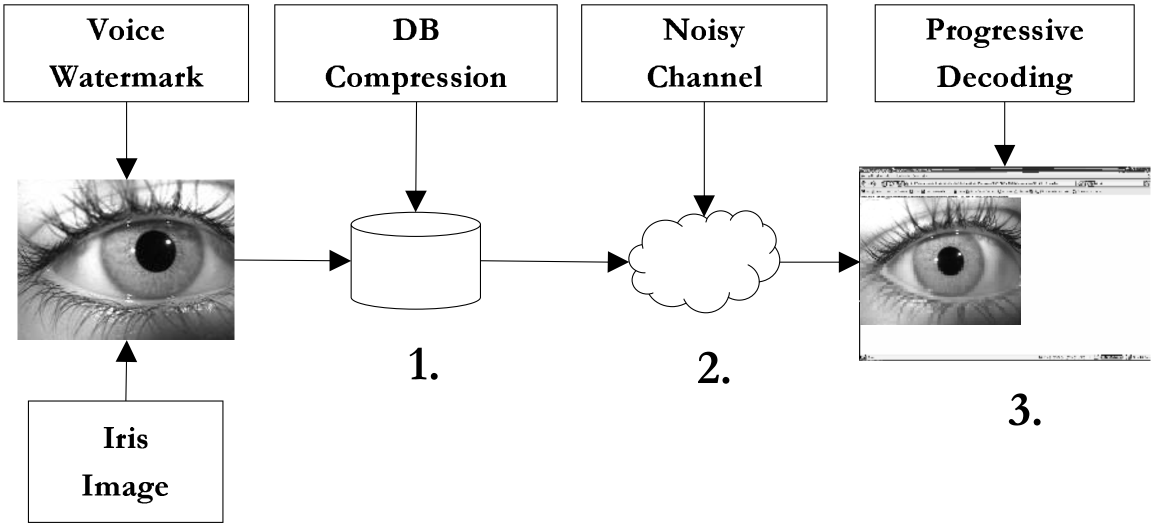


Figure 3.2: Experimental Points of Analysis

Beyond analyzing the change in iris image quality, matching performance, and perceptability of the different parameter variations, we set up three watermark extractability experiments outlined in Figure 3.2. The first of which involves database compression. Most biometric databases containing raw images store data in formats derived from lossy compression algorithms. As there is no current standard for compressing iris images, we took

Original Size	JPEG Quality	J2K Ratio	Compressed Size
301 KB	80	0.95 bpp	≈ 36 KB
301 KB	60	0.75 bpp	≈ 23 KB
301 KB	40	0.55 bpp	≈ 17 KB
301 KB	20	0.35 bpp	≈ 10 KB

Table 3.1: Compression Statistics

two popular compression algorithms (JPEG and JPEG 2000) and compressed the original images in the following ways:

- **JPEG: (compression quality)**- (80,60,40,20)
- **JPEG 2000 (J2K): (bpp)**- (0.95,0.75,0.55,0.35)

Table 3.1 shows statistics for the two compression algorithms. It should be noted that the compressed sizes for the JPEG images represent averages across compression qualities whereas the sizes for J2K are exact.

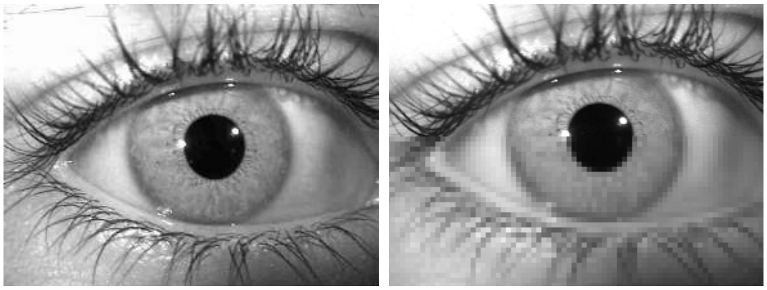
Next, we attempted to simulate environments that may be subject to partial results associated with progressive image transmission (i.e. web browsers). Using the wavelet transform, we simulated partial image transmissions by incrementally decreasing the amount of detail coefficients thresholded similar to techniques in [23]. The percentages of the image that were partially decoded were as follows:

- **Partial wavelet decoding:**- (25%, 50%, 75%).

Figure 3.3 shows a simulated progressively transmitted image as well as an image after application of zero mean Gaussian noise.

The final extraction experiment involved transmitting data over a noisy channel as in mobile communication and satellite systems. We applied additive zero mean white Gaussian noise (AWGN) to the compressed images using Equation 3.5.

$$\text{Noised Image} = \text{I(x,y)} + \text{N} \quad \text{where N} \sim N(0,\sigma^2)$$
$$\sigma^2 \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\} \tag{3.5}$$



(a) JPEG (20) (b) Partial Decoding (50%)

Figure 3.3: Compressed and Progressively Decoded Irides

Beyond the notions of extractability in the application scenarios tested, it is important to consider the success of the watermarking scheme in terms of specific application to a digital chain of evidence. As we will see in Chapter 6, in the proposed framework for a digital chain of evidence, images will be subject to ongoing watermarking and reconstruction as evidence is passed along the chain of entities. The process involves watermarking an image at one entity, then extracting the watermark and reconstructing the image, followed immediately by rewatermarking the image. With this in mind, it is important to establish that the content of the evidence does not change substantially as it is passed along the chain and subject to this rewatermarking process. Simulating a chain of evidence with up to 10 different entities, we examine the degree that iris images change by tracking the number of pixels that change and the degree in which the intensity values change for those pixels that are different. This experiment is performed by looking at 100 images from the set with the watermarking parameter combination #17 in Table 3.2.

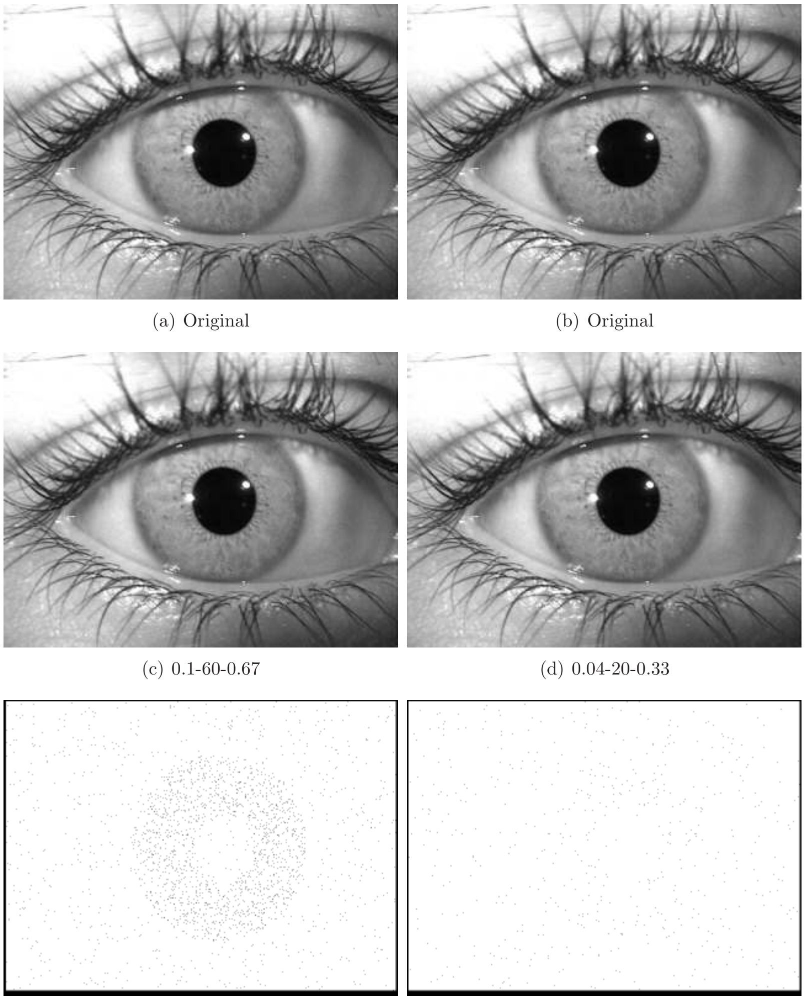
3.5 Experimental Results

3.5.1 Watermark Perceptibility

As a requirement of the proposed system, watermarks should be imperceptible to the user. Figure 3.4 shows the visibility of the watermark in two watermarking schemes. Images (a) and (b) represent the original, (c) and (d) represent the watermarked images, and (e) and (f) represent the difference images between the original and the watermarked. The schemes chosen represent what we consider the most perceptible and least perceptible watermarking parameter combinations. Clearly, the goal of imperceptibility to the naked eye is met as even the most perceptible parameter combination is not noticeable without the help of electronic processing techniques.

3.5.2 Effect on Image Quality

As defined in [67], overall iris image quality ranges from 0-1 based on 7 quality factors. The original quality distribution of the experimental dataset was evenly distributed with an approximately equal amount of images from 10 quality bins (0-0.1,0.1-0.2,0.2-0.3, etc.). In our experiment, each of the watermarking parameter combinations have been assigned a number 1-27. Table 3.2 outlines these assignments in order of encoding strength, number of times encoded, and percent encoded in the iris. Differences in quality between original and watermarked as well as between original and reconstructed images were calculated. All differences were found to be significantly small. The average quality difference across parameter combinations for watermarked images was -0.003, and the average quality difference across parameter combinations for reconstructed images was -0.005.



(a) Original

(b) Original

(c) 0.1-60-0.67

(d) 0.04-20-0.33

(e) 0.1-60-0.67 diff. img.

(f) 0.04-20-0.33 diff. img.

Figure 3.4: Perceptability of Watermarked Images

3.5.3 Effect on Matching Performance

For each user, the hamming distance is calculated between two templates obtained from non-watermarked images. Next, a watermark is applied to all images 27 different times; one for each combination of the parameters. Then the average hamming distance is calculated between all reconstructed image pairs. The Equal Error Rate (EER) of the original image

1	0.1-60-0.67	**10**	0.06-60-0.67	**19**	0.04-60-0.67
2	0.1-60-0.33	**11**	0.06-60-0.33	**20**	0.04-60-0.33
3	0.1-60-0.00	**12**	0.06-60-0.00	**21**	0.04-60-0.00
4	0.1-40-0.67	**13**	0.06-40-0.67	**22**	0.04-40-0.67
5	0.1-40-0.33	**14**	0.06-40-0.33	**23**	0.04-40-0.33
6	0.1-40-0.00	**15**	0.06-40-0.00	**24**	0.04-40-0.00
7	0.1-20-0.67	**16**	0.06-20-0.67	**25**	0.04-20-0.67
8	0.1-20-0.33	**17**	0.06-20-0.33	**26**	0.04-20-0.33
9	0.1-20-0.00	**18**	0.06-20-0.00	**27**	0.04-20-0.00

Table 3.2: Watermarking Parameter Combinations

set was calculated to be 7.164%, and the average EER across the 27 reconstructed schemes was 6.975%. Based on this, we conclude that our watermarking scheme has little impact on average hamming distance across all users. As a result, we believe that recognition performance would remain relatively unaffected by our watermarking system. Cross comparing original to watermarked images provides further evidence to this effect. The average EER across all 27 watermarking schemes was 7.354%, not significantly higher than the original EER.

3.5.4 Extraction after DB Compression

Figure 3.5 shows the effect of JPEG compression on watermark extractability. The x-axis represents each of the 27 parameter combinations as seen in Table 3.2. The y-axis represents average percentage bit error across all 200 images for each parameter combination. As the graph indicates, our watermarking scheme offers an average bit error at or below 5% for JPEG compression down to a quality of 40 for many parameter combinations. Figure 3.6 shows the effect of J2K compression on extractability. We see here that the extractability seems to be more robust offering even more combinations below 5% average bit error across

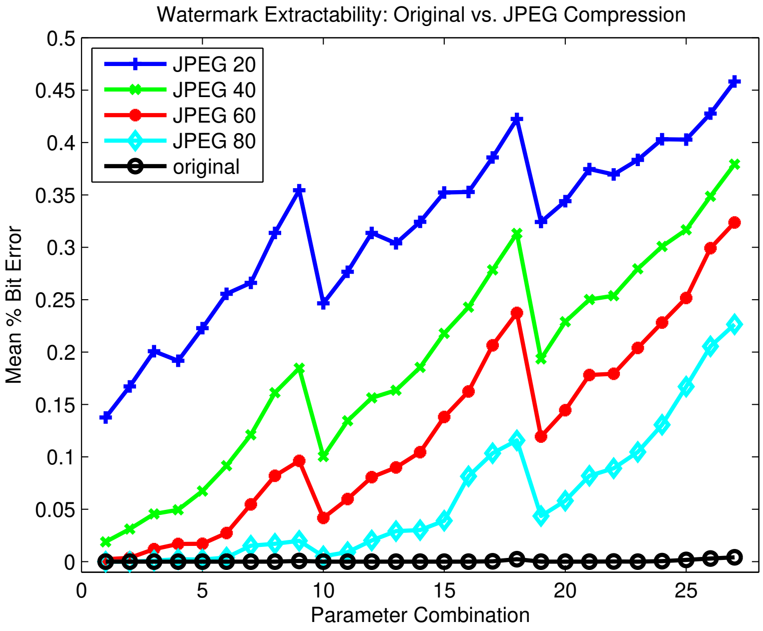


Figure 3.5: Extractability: Original vs. JPEG compressed

compression levels. Furthermore, we see less sensitivity to a change in the encoding strength (see parameter combinations 9-10 & 18-19).

3.5.5 Extraction after Partial Progressive Transmission

Figure 3.7 shows the ability to extract watermarks after various intervals of partial progressive image transmission. The intervals were applied to a previously JPEG compressed image at a quality level of 80. The graph demonstrates that the first 6 parameter combinations allow for relatively successful watermark extraction with bit errors falling below 5%. This holds true with up to 50% of the image partially decoded.

Figure 3.8 shows the same partial progressive extraction after J2K compression at 0.95

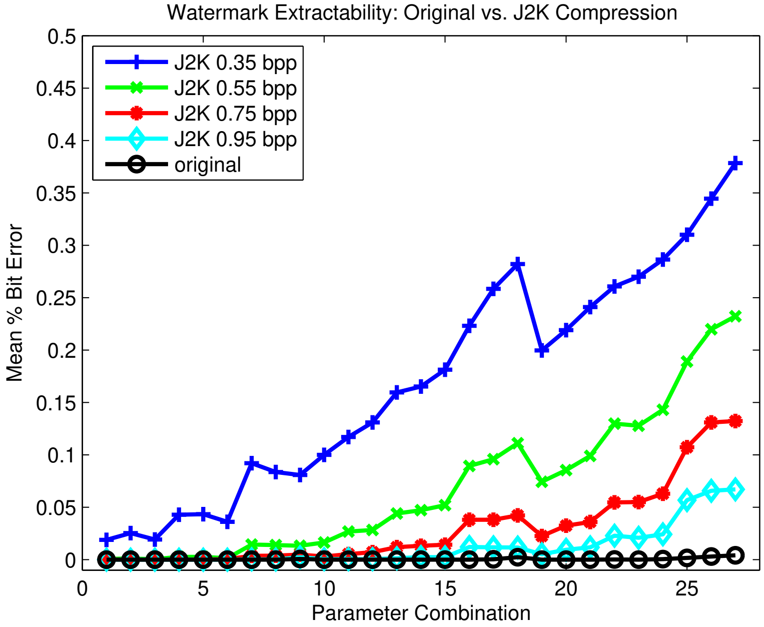


Figure 3.6: Extractability: Original vs. J2K compressed

bpp. We again notice that the J2K compression appears to be less detrimental to the extraction process, offering lower levels of bit error compared to similar JPEG compression ratios.

3.5.6 Extraction after Transmission over a Noisy Channel

We found extraction at a level below 5% bit error was attainable at three amounts of additive white gaussian noise (AWGN) (10^{-5},10^{-4},10^{-3}) for the first 7 watermarking parameter combinations on JPEG (80) compressed images. These results can be seen in Figure 3.10 Once again consistent with the J2K compression results, extraction in our watermarking scheme was less sensitive to AWGN applied to 0.95 bpp J2K compressed images as compared to the

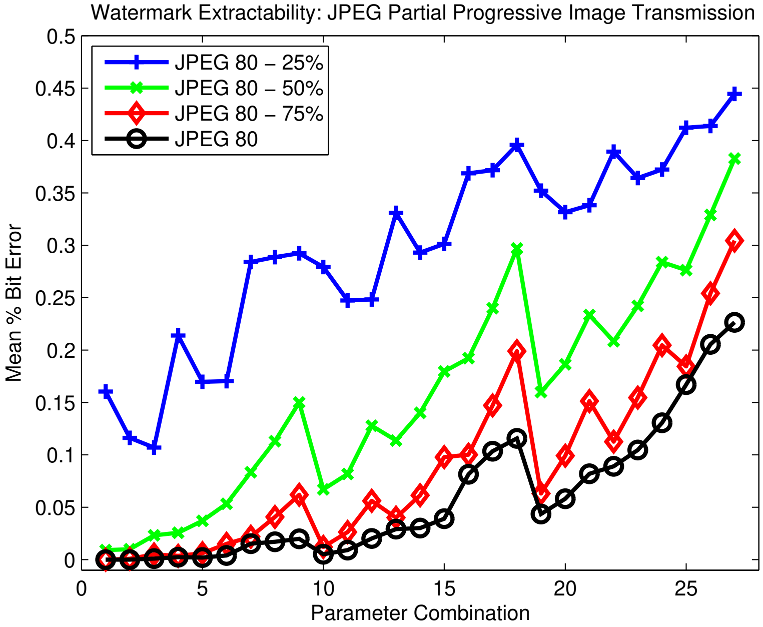


Figure 3.7: Extractability:JPEG Partial Progressive Transmission

JPEG (80) compressed images as seen in Figure 3.10 In this case, it offered approximately 20 combinations with bit error less than or equal to 5% for noise levels (10^{-5},10^{-4},10^{-3}). Furthermore, it offered 6 combinations below 5% bit error at noise level (10^{-2}).

3.5.7 Rewatermarking Along a Chain of Evidence

As mentioned in the experimental design, we also considered the success of the watermarking scheme in terms of specific application to a digital chain of evidence. With this in mind, it is important to establish that the content of the evidence does not change substantially as it is passed along the chain and subject to this rewatermarking process. Simulating the generation of evidence, we first watermarked 100 images with 64 bit binary sequences which

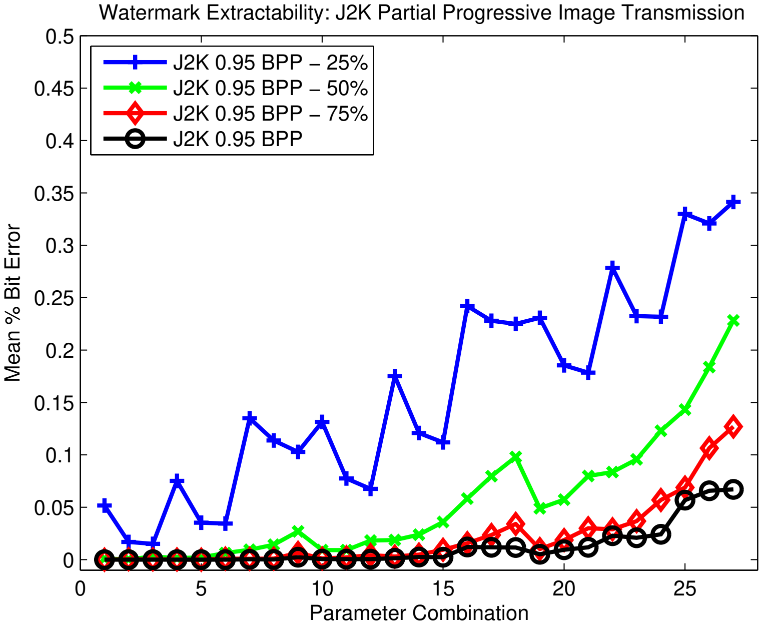


Figure 3.8: Extractability:J2K Partial Progressive Transmission

could simulate a biometric feature vector as in the previous experiments. Then, to simulate transmission of the evidence across a chain of entities we apply the following three-step process 10 times on all of the 100 images:

1. Extract watermark and verify its validity.

2. Reconstruct image so evidence viewing / analysis can take place.

3. Rewatermark the reconstructed image with current entity's biometric feature vector before transmission to next entity in the chain.

To analyze the effect of rewatermarking on the appearance of the iris digital evidence, Figure 3.11 shows the percentage of pixels in the reconstructed image which have different

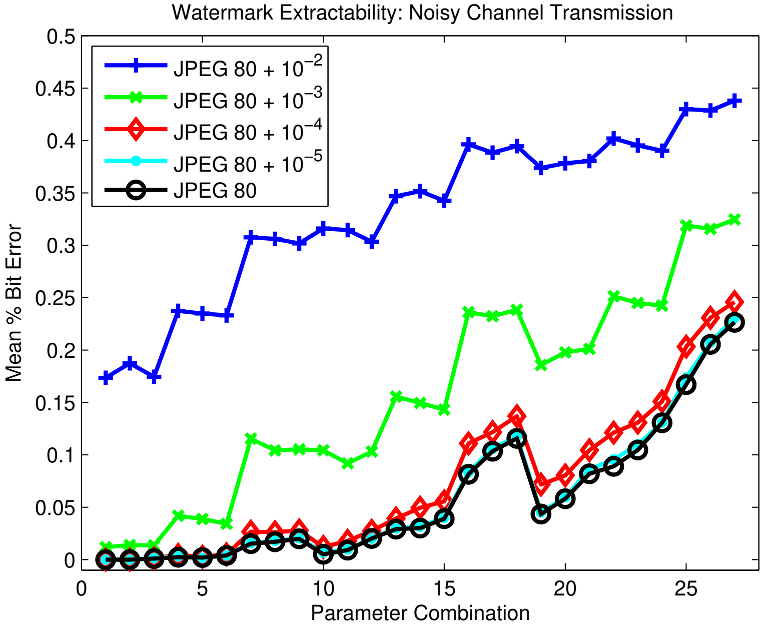


Figure 3.9: Extractability:JPEG Noise

values compared to the original, unwatermarked iris digital evidence. We see that after the second entity in the chain reconstructs the evidence after extracting the watermark from the first entity, 0.14% or 424 out of 307, 200 (640 x 480) pixels are different as compared to the original evidence. Note, this figure represents the average difference across all 100 iris images tested. Looking further at the figure, this mean percentage difference between the reconstructed images and the original images linearly increases as the evidence is moved along the chain with the final difference falling at 1.35% or 4, 149 pixels. Having established the percentage of pixels which change as the evidence is moved along a chain of entities, we look into the degree in which changed pixels vary in terms of intensity. Figure 3.12 takes the same form as the previous figure except the y-axis reflects the average intensity difference for pixels of the reconstructed images having different values than the original image. In

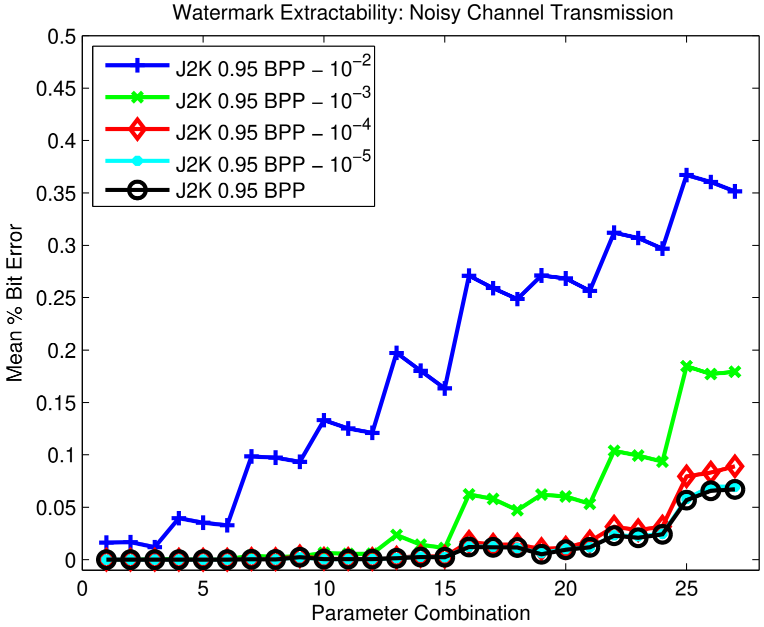


Figure 3.10: Extractability:J2K Noise

this case, the difference remains steady around 3.29 intensity points as the evidence moves along the chain. Naturally, this is a function of the encoding strength parameter chosen. As mentioned, the encoding parameter combination chosen was #17 from Table 3.2. This resulted in perfect extraction of the watermark in every case. The differences observed in quanity and intensity between the original and reconstructed images along the chain of evidence are not likely perceptible to the human eye. This is supported by Figure 3.13 which displays the reconstructed images after the first and tenth reconstructions. Looking only at (a) and (b), one would not be able to distinguish between the two images without the assistance of software techniques. Furthermore, in (c) and (d), the visible portions of the difference maps between the original and reconstructed images (first and tenth) are faint at best. Besides human perceptibilty, it is unlikely that such differences would have any level

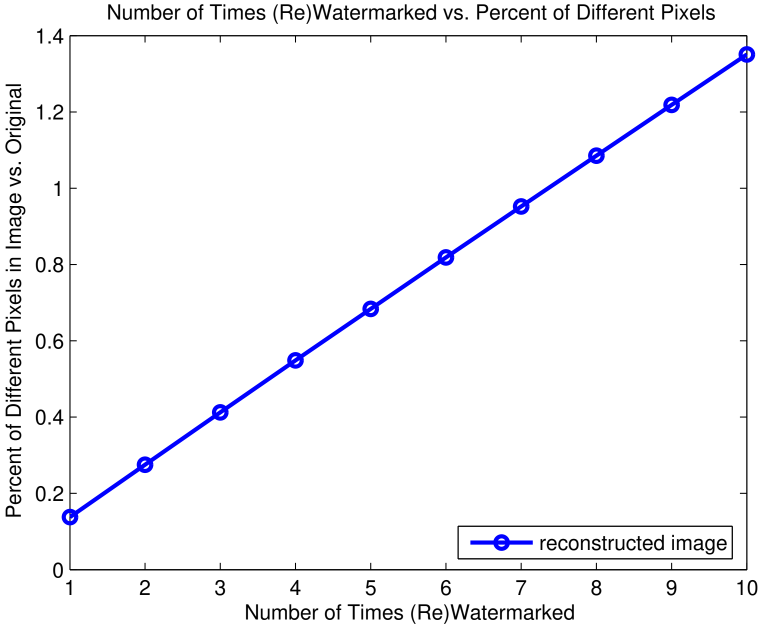


Figure 3.11: Percentage Pixel Differences Between Reconstructed and Original Evidence

of serious impact on software-based biometric analysis that might take place at an arbitrary entity in the chain. This is corroborated by the results of the experiments on biometric image quality and matching performance as well.

3.6 Chapter Summary

The combination of the techniques employed by the system offers various attractive features and advantages. It affords multiple levels of authentication; through cryptography, after decryption through watermarking, and through multimodal biometric verification. By watermarking a feature vector from a behavioral biometric, we achieve both added protection after decryption and a degree of biometric replaceability both at the raw image level and at the feature vector level. Replaceability is achieved as a new watermark can be embedded to

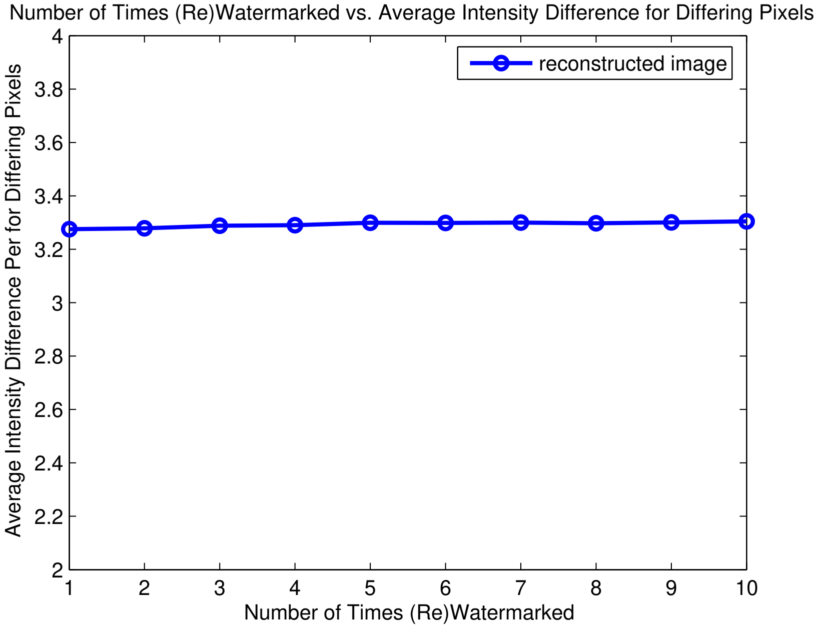
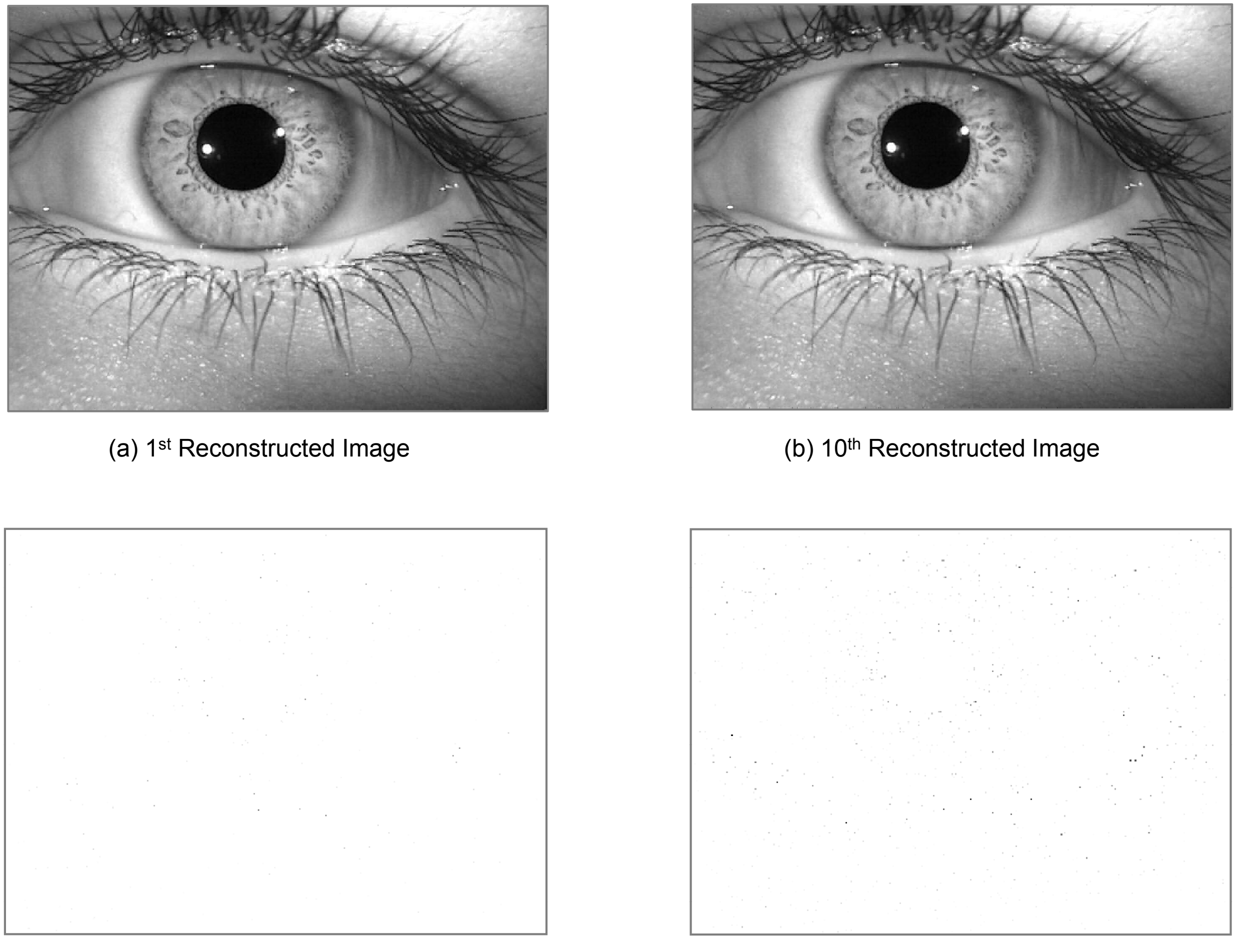


Figure 3.12: Intensity Differences Between Reconstructed and Original Evidence

replace the iris biometric or a new passphrase can be assigned to replace the voice biometric. The system also decreases the chance of a true compromise of credentials. Through public key encryption, the profile is resistant to exposure whether stored electronically or on a smart card. The framework offers multiple levels of forgery detection with a combination of watermarks and file hashing. Essentially the profile can be considered completely fragile as it can detect modification of a single bit through cryptographic hash comparisons. Additionally, the system offers non-repudiation of origin through PKI as well as origin tracking through watermarking. Finally, the cost of compromise is lower than systems watermarking biometric data with physiological biometric templates as they effectively double the risk associated with compromised profiles / tokens. In these systems, an unauthorized user has indefinite access to both the original raw biometric data (one modality) and the biometric template data (another modality). Given template reconstruction technologies, both biometrics can be considered compromised, whereas only one biometric is compromised in the event of a behavioral template watermark (replaceable) [7].

Beyond the attractive features of the framework, we have shown that our watermarking scheme does not significantly impede iris image quality or biometric matching performance. Additionally, we have demonstrated that our watermarking scheme provides a degree of robustness to three realistic watermarking application scenarios: database compression, data transmission over a noisy channel, and partial data loss (progressive decoding). Finally, we have demonstrated how rewatermarking images as evidence moves along the entities in a digital chain of evidence is not likely to have detrimental impacts on the evidence in terms of human pereceptibility or biometric matching. The results should suit systems requiring the establishment of a chain of evidence nicely as reconstructed images have minimal quantities of altered pixels.



(a) 1st Reconstructed Image

(b) 10th Reconstructed Image

(c) Difference Map Between Original and 1st Reconstructed Image

(d) Difference Map Between Original and 10th Reconstructed Image

Figure 3.13: Perceptibility of Reconstructed Evidence Along the Chain of Evidence.

This page intentionally contains only this sentence.

Chapter 4

Digital Hardware Fingerprinting

4.1 Introduction

As the field of biometrics continues to grow, so does its areas of application. Such areas can include access control in protected sites and border control, remote authentication in commercial applications, and identification of criminal suspects or enemies on the battlefield. Regardless of the intended application, various measures must be taken to ensure the accuracy and integrity of these deployments. Two ways that biometric systems can be compromised include fabrication and alteration of data. Fabrication of biometric data could occur at many points within a biometric system and usually is the result of an act with malicious intent. Whether at the time of data acquisition, matching, or database access, various vulnerabilities may allow raw biometric images to be created and maliciously injected into a system. Similarly, biometric data may also be maliciously altered throughout the course of operation in a biometric system. Besides actions with malicious intent, unintentional alteration of images during the collection, transmission, or storage blocks of a system can take place. To make matters worse, whether intentional or unintentional, there often is no obvious cue that an image has been fabricated or altered in the first place. This is of particular importance to applications where a "chain of evidence" must be established. Such a chain is useful in assembling cases to prosecute criminal activity, establishing identity dominance in the battlefront, and discovering fraudulent activity in commercial systems. In an effort to minimize the presence of fabricated / altered images in such systems, the notion of source identification is applied. Falling under the field of digital forensics, digital hardware fingerprinting provides the ability to identify and validate the source hardware which captured an image. Whether establishing a chain of evidence or addressing a specific biometric vulnerability, application of digital hardware fingerprinting for biometric image source validation should prove to be very useful.

Digital hardware fingerprinting is the process of identifying the source hardware used to capture an image regardless of the scenery or primary image content. The primary method of identifying the source hardware from which an image originated is analyzing differences in images resulting from imaging sensor imperfections [86]. Due to slight inconsistencies in the production process, all sensors are subject to small manufacturing imperfections. These imperfections lead to the necessary observance of noise (sometimes visually undetectable by humans) in images collected. Although previous work has focused on devices using optical technology to capture images, such noise would also be present in sensors relying on different technologies for image capture such as capacitance, thermal, or piezoelectric signals. Identification of source hardware can potentially occur at different levels of granularity. Figure 4.1 shows four levels that may be considered in a source hardware identification

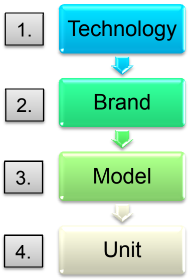


Figure 4.1: Source Hardware Identification Levels.

problem: technology, brand, model, and unit. Based on the four levels, various questions can be answered:

- Was the image in question captured from a sensor relying on technology X or Y? (Technology)
- Was the image in question captured from a device manufactured by vendor X or Y? (Brand)
- Was the image in question captured from a device corresponding to model X or Y manufactured by vendor Z? (Model)
- Was the image in question captured from a unit A or unit B of model X manufactured by vendor Z? (Unit)

Naturally, unique challenges exist to performing source model identification at different levels. In this chapter, we provide a look into the feasibility of determining source identification at the unit level (4) using photo-response nonuniformity noise (PRNU) present in biometric fingerprint readers as well as biometric iris cameras. To our knowledge, it is the first work to demonstrate the ability to identify the hardware source used to collect biometric fingerprint and iris images. To do so, we adopt the technique presented by Lukas et al. in [86]. Secondly, we formally establish the effect of varying the amount of images used to arrive at reference templates for capture devices at the unit level.

4.2 Approach

As a means to identify biometric capture devices, whether fingerprint readers or iris cameras, we adopt the approach proposed by Lukas et al. in [86]. This approach is based on estimating pixel nonuniformity (PNU), a portion of the photo-repsonse nonuniformity (PRNU) inherent to every image captured by the readers. The remainder of this section is broken down into two parts: a description of the general framework for identifying hardware sources through PNU noise and a description of the wavelet-based denoising algorithm utilized in [86].

Identification Process The process of device identification can then be broken down into two main steps:

1. **Calculate Reference Patterns**. For each fingerprint reader, calculate a reference pattern by taking an average of the noise residual estimates across multiple training images as seen in Equation 4.1.

$$\mathbf{R}_i = \frac{\sum_{k=1}^{N} \mathbf{p}^{(k)} - F(\mathbf{p}^{(k)})}{N} \tag{4.1}$$

Here, N represents the number of images used to generate the reference pattern, $\mathbf{R}_i$, $\mathbf{p}^{(k)}$ represents each image in the training set, and F represents a denoising filter. It should be noted that while F can represent any denoising filter, Lukas et al. found that a wavelet-based approach yielded the best results [86]. The specifics of the wavelet-based denoising is described later in this chapter. Figure 4.2 shows an example of reference patterns for two different fingerprint readers within one of the databases tested.

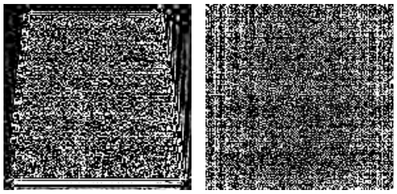

(a) Microsoft (b) BioTouch

Figure 4.2: Example reference patterns for Microsoft and BioTouch Readers from the WVU database.

2. **Correlate Noise Residuals to Reference Patterns**. For each image to be tested, extract the noise residual $\mathbf{p}^{(k)} - F(\mathbf{p}^{(k)})$, and measure the correlation, $\mathbf{C}$, to each reference pattern, $\mathbf{R}_i$, for all of the reference patterns in question. In [86], Lukas et al. propose the correlation measure seen in Equation 4.2, although in theory, any correlation measure could be applied.

$$\mathbf{C}_i = corr(\mathbf{p}, \mathbf{R}_i) = \frac{(\mathbf{p} - \bar{\mathbf{p}}) \cdot (\mathbf{R}_i - \bar{\mathbf{R}}_i)}{\|(\mathbf{p} - \bar{\mathbf{p}})\| \|(\mathbf{R}_i - \bar{\mathbf{R}}_i)\|} \tag{4.2}$$

Wavelet-based Denoising Algorithm The wavelet-based denoising approach in [86] is described in four steps.

1. Calculate the first through fourth wavelet decompositions of the original noisy image using the 8-tap Daubachies Quadratic Mirror Filters (QMF). The vertical, horizontal, and diagnoal subbands are denoted by $v(i,j)$, $h(i,j)$, and $d(i,j)$ respectively. Here (i,j) represents the coefficients for each pixel in each of the three subbands.

2. In each subband, estimate the local variance of the noise-free image for each wavelet coefficient using MAP estimation for four sizes of a $W \times W$ neighborhood N, for $W \in \{3,5,7,9\}$ as seen in Equation 4.3.

$$\hat{\sigma}_W^2(i,j) = max\Big(0, \frac{1}{W^2} \sum_{(i,j)\in N} h^2(i,j) - \sigma_0^2\Big) \tag{4.3}$$

Then apply Equation 4.4 to arrive at the minimum of the four variances as the final estimate.

$$\hat{\sigma}_W^2(i,j) = min\Big(\sigma_3^2(i,j), \sigma_5^2(i,j), \sigma_7^2(i,j), \sigma_9^2(i,j)\Big) \tag{4.4}$$

3. Determine the denoised wavelet coefficients by applying the Wiener filter as seen in Equation 4.5

$$h_{den}(i,j) = h(i,j)\frac{\hat{\sigma}^2(i,j)}{\hat{\sigma}^2(i,j) + \sigma_0^2} \tag{4.5}$$

Similarly, the filter is applied to $v(i,j)$ and $d(i,j)$.

4. Steps 1-3 are repeated for each level and color channel.

In [86], the authors used $\sigma_0 = 5$ in the experiments as do we in this work. It should be noted that due to the grayscale nature of the fingerprint and iris images, it is not necessary to perform Step 4 across multiple color channels.

4.3 Experimental Design

As mentioned previously, this work investigated the notion of source identification in both fingerprint readers and iris cameras. In total, four databases were tested including three fingerprint databases and one iris database. Of the three fingerprint datasets considered, the first was a WVU collection consisting of images from two sensor models (3 units each). The second collection came from both WVU and Clarkson which consists of images from three sensor models (2 units each). The last set is from the first three years of the Fingerprint Verification Competition (FVC) which consists of images from 8 different units [88, 89, 90]. The WVU data was collected specifically with hardware fingerprinting experiments in mind, while the other two datasets were collected primarily for biometric testing purposes. With that in mind, the WVU dataset consists of fingerprint images from 4 subjects who each provided 100 images per sensor (25 images from 4 digits) for a total of $2,400$ images. The WVU / Clarkson datasets each have substantially more images and the experiment only used a subset of the fingerprint images available from each. Specifically, we randomly selected $1,000$ images per sensor pertaining to the right index and thumb. The FVC dataset was comprised of 10 subjects with 8 images per subject, collected from the index and middle finger, totaling 640 images. A summary of fingerprint readers including model details can be found in Table 4.2 and example fingerprints and corresponding noise residuals can be found in Table 4.1.

The final dataset tested comes from images collected with four different iris cameras. The first set of images comes from the ICE Challenge database and the second set comes from the CASIA III database [99, 107]. The final two sets of images were collected with two different iris cameras at WVU. Considering only images with dimensions of 640 X 480 pixels, 200 images were used from each device. A summary of iris cameras including model details can be found in Table 4.4 and example images and corresponding noise residuals can be found in Table 4.3.

Although the amount of users in the WVU and FVC data may be prohibitively small for a traditional biometric experiment, we note this experiment is not studying biometric recognition or identification. Instead, we are studying sensor identification. To that effect, we believe the variety subjects and associated number of fingerprint provides sufficient variation for our tests. We applied a cross-validation framework for all datasets for testing the proposed methods. In our experiments, we tested the success of the digital hardware fingerprinting techniques while varying the number of images used to generate reference patterns using the methodology described in the previous section. Table 4.5 lists the training and testing breakdowns for each dataset. It is important to note that 10 fold cross validation was applied for each test. Therefore, total number of tests in the WVU and FVC datase results range depending on the split with WVU ranging from $3,990$ (399 * 10) to $1,440$ (144 * 10) and FVC results range from 790 (79 * 10) to 160 (16 * 10) tests. On the other hand, the test size was constant for the WVU / Clarkson set due to the availability of images and includes includes $5,000$ tests (500 * 10 folds). Finally, applying the same 10 fold cross validation, the number of tests in the iris experiments varied from 720 (72 * 10) to $1,990$ (199 * 10). It

Reader	Fing.	Noise	Sensor	Fing.	Noise
WVU Identix #1			WVU Microsoft #1		
WVU Identix #2			WVU Microsoft #2		
WVU Identix #3			WVU Microsoft #3		
WVU Precise			Clarkson Precise		
WVU Secugen			Clarkson Secugen		
WVU CrossMatch			Clarkson CrossMatch		
FVC KeyTronic			FVC Microelectronics		
FVC Identicator			FVC Identix		
FVC Biometrika			FVC Precise		
FVC CrossMatch			FVC DigitalPersona		

Table 4.1: Example fingerprints and noise residuals from three different data sets.

Brand	Model	Tech.	Width	Height
Microsoft (WVU 1-3)	Fingerprint Reader	O	355	390
Identix (WVU 1-3)	BioTouch200	O	256	255
Precise Biometrics (WVU / Clarkson)	AX 100	C	200	200
Secugen (WVU / Clarkson)	Hamster III	O	260	300
CrossMatch (WVU / Clarkson)	Verifier 300 LC	O	640	480
KeyTronic (FVC)	Secure Desktop Scanner	O	300	300
Microelectronics (FVC)	TouchChip	C	256	364
Identicator Technology (FVC)	DF-90	O	448	478
Identix (FVC)	TouchView II	O	388	374
Biometrika (FVC)	FX2000	O	296	560
Precise Biometrics (FVC)	100 SC	C	300	300
CrossMatch (FVC)	V300	O	640	480
DigitalPersona (FVC)	U.are.U 4000	O	328	364

Table 4.2: Fingerprint reader details from three different data sets. Tech. = Technology {O=optical, C=capacitive}.

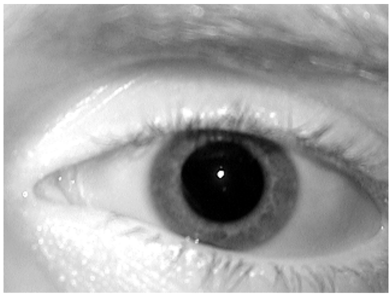
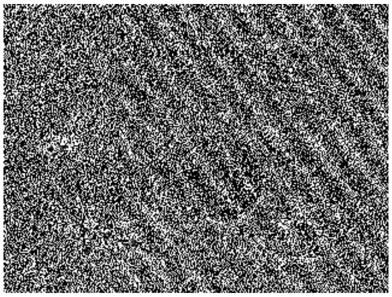
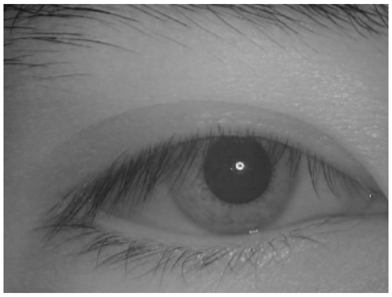
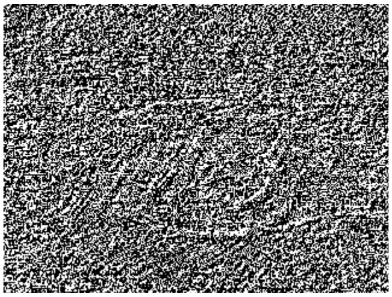
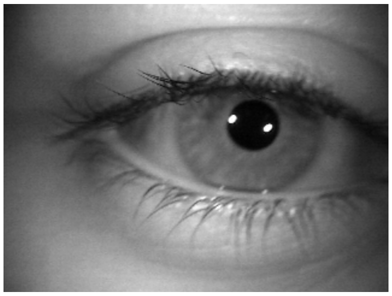
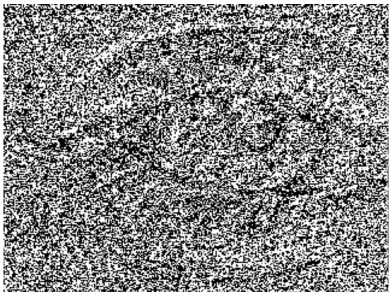
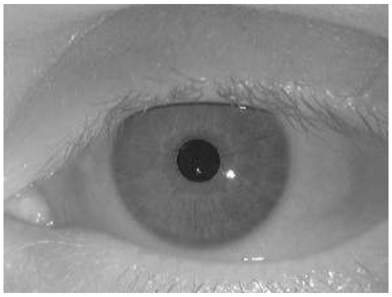
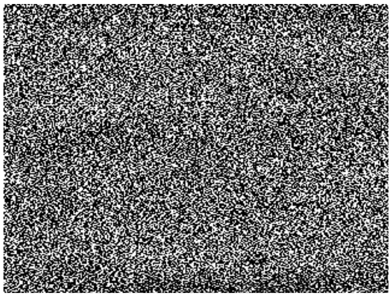
Camera	Iris Image	Noise Residual
ICE		
CASIA		
WVU 1		
WVU 2		

Table 4.3: Example irises and noise residuals from four different data sets.

Image Set	Brand	Model	Width	Height
ICE	LG	EOU 2200	640	480
CASIA	OKI	IRISPASS h	640	480
WVU 1	OKI	IRISPASS h	640	480
WVU 2	EverFocus	EQ100A/EN	640	480

Table 4.4: Iris camera details.

WVU									
Train	1	2	4	8	16	32	64	128	256
Test	399	398	396	392	384	368	336	272	144

WVU / Clarkson									
Train	1	2	4	8	16	32	64	128	256
Test	500								

FVC									
Train	1	2	4	8	16	32	64	n/a	n/a
Test	79	78	76	72	64	48	16	n/a	n/a

Iris									
Train	1	2	4	8	16	32	64	128	n/a
Test	199	198	196	192	184	168	136	72	n/a

Table 4.5: Experimental training variation measured in images / device / fold.

is important to note that there is no overlap of images within the training and testing sets across any of the experiments.

4.4 Experimental Results

4.4.1 Fingerprint Readers

The results in this section report sensor identification at the unit level. In this case, we wish to distinguish which unit the image was captured from given a pool of units. Therefore, a test noise residual is compared against reference patterns for each unit from the dataset in consideration. For each dataset, we provide example histograms of match / non-match distributions, confusion matrices for specific train / test sets, and Cumulative Match Characteristic (CMC) curves as the train / test splits vary.

FVC Dataset

The first set of experiments was performed on the FVC data. Figure 4.3 displays the difference in correlation between match and non-match comparisons of test noise residuals and sensor reference patterns. As we can see in the figure, perfect separation is achieved when considering Identicator test residuals against reference patterns produced from 32 training images per sensor. While perfect separation was achieved in this instance, this was not the case across all sensors in all train and test splits. An example of which can be seen in Table **??**. In the table, we see that the Identix test residuals are occasionally misclassified as having originated from other sensors. It is interesting to note that the distribution of errors is fairly uniform across the other seven sensors. With the exception of the Identix sensor, no errors are made on experiments training on at least 8 images. Figure 4.4 shows a Cumulative Match Characteristic (CMC) plot which indicates the overall accuracy across sensors as the train / test splits are varied. Here we see the rank one identification rate when training on 1 image per sensor falls around 85%. This is fairly high considering this is the smallest amount of data that could be used to generate a reference pattern for a sensor. When 64 images are used to generate reference patterns for each sensor, the rank 1 identification rate exceeds 98%. Furthermore, the only reason the identification rate is not 100% is due to errors made on classifying the Identix noise residuals.

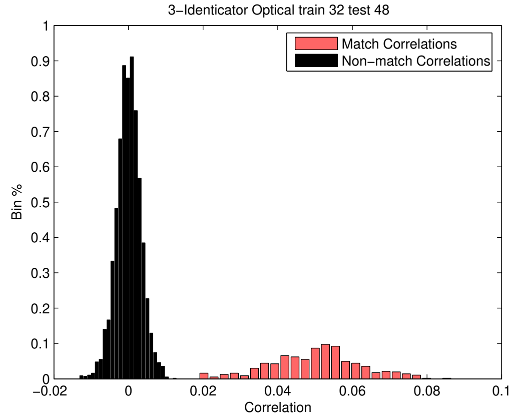


Figure 4.3: FVC example match and non-match distributions with 32 training images per sensor.

Classified / Actual	Key-Tronic	Micro-electronic	Identi-cator	Identix	Biometrika	Precise	Cross-Match	Digital-Persona
KeyTronic	480	0	0	0	0	0	0	0
Microelectonic	0	480	0	0	0	0	0	0
Identicator	0	0	480	0	0	0	0	0
Identix	14	18	23	365	14	13	10	28
Biometrika	0	0	0	0	480	0	0	0
Precise	0	0	0	0	0	480	0	0
CrossMatch	0	0	0	0	0	0	480	0
Digital Persona	0	0	0	0	0	0	0	480

Table 4.6: FVC confusion matrix when training on 32 images per sensor

WVU Dataset

The performance on the WVU dataset is the highest among the three datasets considered. Virtually all instances achieve perfect separation, therefore it is not beneficial to display a histogram of match and non-match distributions. However, in Table 4.7, the confusion matrix resulting from training on 1 image per sensor is displayed. Again considering the minimal training requirement, we see only sporadic errors across the 3, 990 test cases per

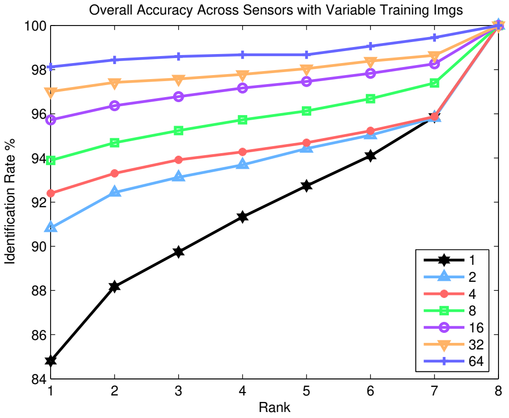


Figure 4.4: FVC sensor identification as a function of training set size.

Classified Actual	BioTouch #1	BioTouch #2	BioTouch #2	Microsoft #1	Microsoft #2	Microsoft #3
BioTouch #1	3942	14	32	1	1	0
BioTouch #2	6	3967	17	0	0	0
BioTouch #3	3	10	3977	0	0	0
Microsoft #1	0	0	0	3989	1	0
Microsoft #2	0	0	0	0	3990	0
Microsoft #3	0	0	0	0	0	3990

Table 4.7: WVU confusion matrix when training on 1 image per sensor

sensor. This high level of performance is reflected in the CMC curve shown in Figure 4.5. Here, only 2 train / test scenarios do not achieve perfect rank 1 identification (training on 1 and 2 images per sensor). Nevertheless, the rank 1 identification when using only 1 image per sensor exceeds 99%.

WVU / Clarkson Dataset

The most challenging dataset for sensor identification ending has been the WVU / Clarkson dataset. Observing a notable difference compared to the previous to databases, Figure 4.6

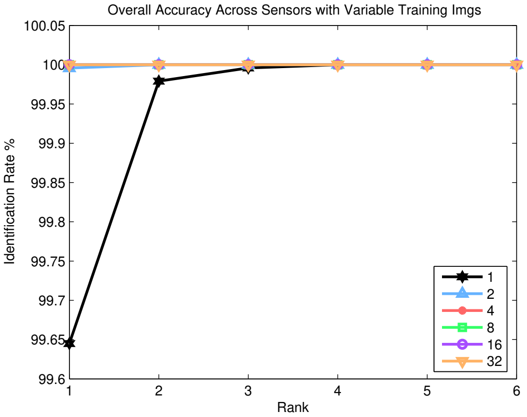


Figure 4.5: WVU sensor identification as a function of training set size.

shows a slight overlap between match and non-match distributions when generating reference templates from 128 training images. In all but one sensor in the WVU and FVC datasets, training on 128 images exceeded the minimum required to achieve perfect separation. This pattern can also be seen in Table A.22 which displays the confusion matrix when training on 128 images. While the results can still be considered promising as the overall rank 1 identification accuracy is near 90%, we observe far more errors across the test cases. It is also interesting to note where the errors are made. Somewhat intuitively, more errors are made misclassifying the WVU Secugen noise residuals as Clarkson Secugen noise residuals than any other sensor's residuals. This may be the case as they are the same model sensor. Surprisingly, this pattern does not hold true for the WVU CrossMatch residuals as they are misclassified as Clarkson residuals the fewest number of times when compared to the other sensors. We are still investigating why this may be the case. Figure 4.7 displays the CMC plots for the final dataset. Once again, this data proved to be the most challenging of the

Classified Actual	WVU Precise	WVU Secugen	WVU CrossMatch	Clarkson Precise	Clarkson Secugen	Clarkson CrossMatch
WVU Precise	4979	0	0	21	0	0
WVU Secugen	364	3146	179	492	596	223
WVU CrossMatch	313	188	3781	366	246	106
Clarkson Precise	32	3	0	4963	2	0
Clarkson Secugen	14	25	7	14	4940	0
Clarkson CrossMatch	27	1	2	1	10	4959

Table 4.8: WVU / Clarkson confusion matrix when training on 128 images per sensor

three sets as we see the rank 1 identification accuracy drops to 45%. However, any reasonable application of source identification will likely have access to more than one training image to generate reference patterns. Along those lines, the rank 1 identification rate when training on 256 images is approximately 95% which can once again be considered promising results.

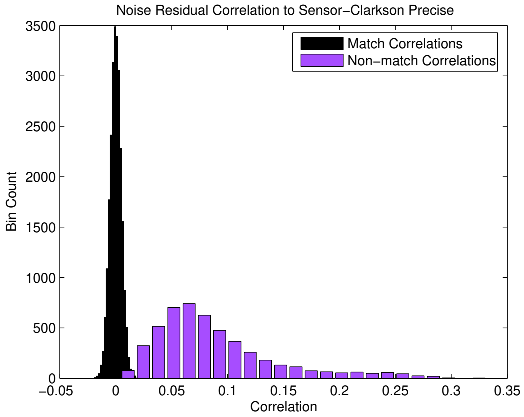


Figure 4.6: WVU / Clarkson example match and non-match distributions with 128 training images per sensor.

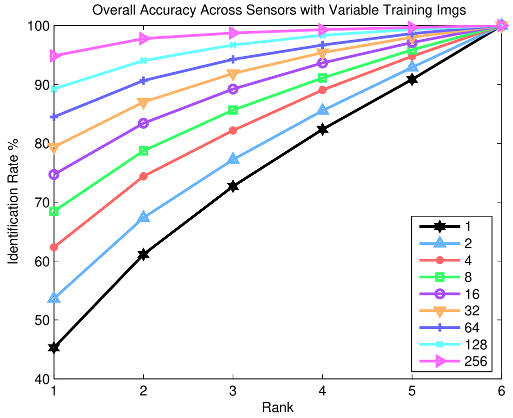


Figure 4.7: WVU / Clarkson sensor identification as a function of training set size.

4.4.2 Iris Cameras

Although we consider fewer devices (a function of availability alone), the hardware fingerprinting experiments for the iris devices are similarly encouraging. Figure 4.8 displays match and non-match distributions for noise residual correlations to the ICE LG camera's reference patterns after training on 4 images. We see that although the separation between match and non-match distributions is not perfect, there is a well defined split between the distributions. This trend holds true against all sensors when training on only 4 images per sensor. This can be seen in the confusion matrix for the identification experiment in Table 4.9. Here a small number of classification errors are made in two of the sensors across each of the 1, 960 tests. This amount of errors continues to drop as more images are used to generate reference patterns.

This relationship is visualized in Figure 4.9. In the figure we see the rank 1 identification rate approaches 98%. Furthermore, this rate approaches 100% after using 8 or more images to

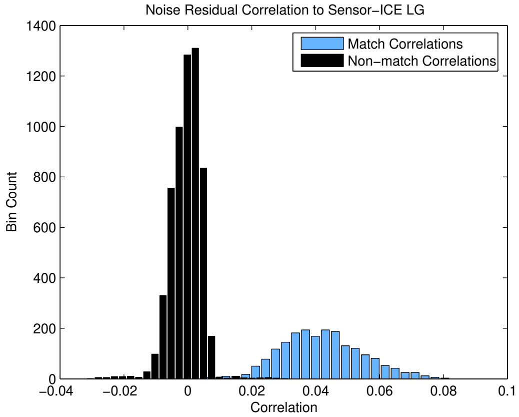


Figure 4.8: ICE LG sensor example match and non-match distributions with 4 training images per sensor.

Classified / Actual	ICE LG	CASIA OKI	WVU OKI	WVU EverFocus
ICE LG	1951	8	1	0
CASIA OKI	0	1960	0	0
WVU OKI	20	13	1926	1
WVU EverFocus	0	0	0	1960

Table 4.9: Iris confusion matrix when training on 4 images per camera

generate reference templates for each device. These results are encouraging as they indicate that the approach can be used to perform source validation on devices that capture face (from [86], fingerprint, and iris images despite the different technologies used within the different capture devices associated with each modality.

4.5 Discussion

While the results of the experiments clearly demonstrate that inherent PNU noise can be used as a means for performing sensor hardware identification in biometric fingerprint readers and

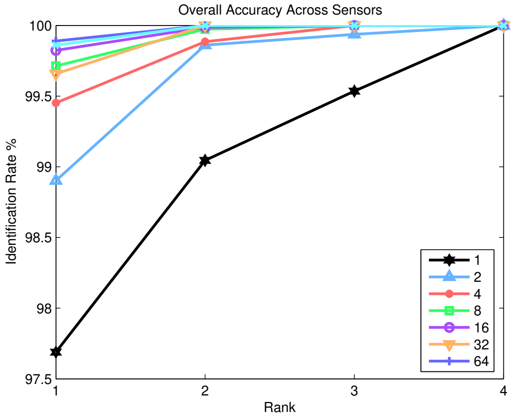


Figure 4.9: Iris camera identification as a function of training set size.

iris cameras, there are a number of considerations which must be mentioned. Most notably, the databases tested only contain limited sets of pairs of identical units (3 sensors of each model in WVU, 2 sensors of each model in WVU / Clarkson, and 2 matching iris camera models). Increasing the number of identical units may result in a decrease in identification performance. At this point, it is unclear if sensor identification with the applied technique would be possible in a pool of 100's or $1,000$'s of the same model sensor at the unit level. Additionally, as different models of fingerprint readers capture images at different resolutions (see 4.2) cropping of noise residuals was performed when necessary as the chosen method of correlation requires that the noise residuals have the same dimension. Although the results seem to indicate this method of handling dissimilar images sizes is sufficient, there are a number of options one may choose to exercise when dealing with this issue, not the least of which is resizing the original image before denoising. The images could be resized but this may introduce artifacts that could artificially enhance performance. To avoid this, different correlation procedures could be applied such as normalized cross correlation which would not require equally sized images.

4.6 Chapter Summary

This chapter investigated the notion of sensor identification in biometric fingerprint readers and iris cameras. We established the prospects of performing identification based on estimating PNU noise inherent to image through wavelet based denoising as proposed by Lukas et al. in [86]. Beyond presenting the ability to perform sensor fingerprinting on biometric devices at the unit level, we established the effect of varying the number of images used in arriving at reference patterns. Having looked at the minimum training requirements, we conclude that sensor identification can be performed with a great deal of accuracy (in three databases) even if one has access to only 1 image to establish a reference pattern. The application of a digital hardware fingerprinting technique such as the one tested in this work can be used both as a method of counteracting vulnerabilities at the time of check in biometric systems as well as allowing an individual to establish a "chain of evidence" which is often critical in systems such as assembling cases to prosecute criminal activity, establishing identity dominance in the battlefront, and discovering fraudulent activity in commercial systems.

This page intentionally contains only this sentence.

Chapter 5

Keystroke Dynamics

5.1 Introduction

Over the history of the field, a large number of keystroke dynamics studies have taken place in controlled environments. Whether subject to heavily supervised lab conditions, a variety of explicit rules, or standardized and uniform capture equipment, these experiments are not typically in line with our proposed application environments. Although stand-alone deployment in an environment with homogenous equipment is undoubtedly still a relevant area of application, current application demands are moving toward web-oriented environments in which choice of hardware, user supervision, and other external noise factors are not controllable. Although we admittedly do not cover all potential sources of noise in this type of application (i.e. mimicry), we attempt to simulate this "free" environment to the best of our ability by employing a completely unsupervised, web-based collection system with no hardware restrictions. This environment will be well suited to chain of evidence applications where individual entities within the chain may be accessing evidence from remote unsupervised situations.

As noted previously, the relationship between the individuals components of the credential sets are of specific interest to this work. In particular, the degree of familiarity that the user has to each component. Figure 5.1 presents the idea that the more times a user is exposed to a stimulus (component of a credential set), the more familiar they become with typing the component. We hypothesize that the username is easily the most familiar of the three components considered as most individuals are exposed (required to type it) multiple times on a daily basis. Following the username, an English password is likely the component with a lesser degree of typing familiarity as it may not be typed daily but it has probably been typed before if not multiple times. Finally, a random password containing characters of varying capitalization, digits, and special characters probably represents the lowest attainable familiarity as there is very little chance that the user has ever seen such a password, much less typed it. By assigning these three types of components to create credential sets, we are able to investigate the degree in which familiarity to input stimulus affects keystroke dynamics performance. Here again, we find this to be of particular interest in chain of evidence applications where the input stimulus is likely to be some derivation of the entity's name, which in turn can serve as a form of digital signature.

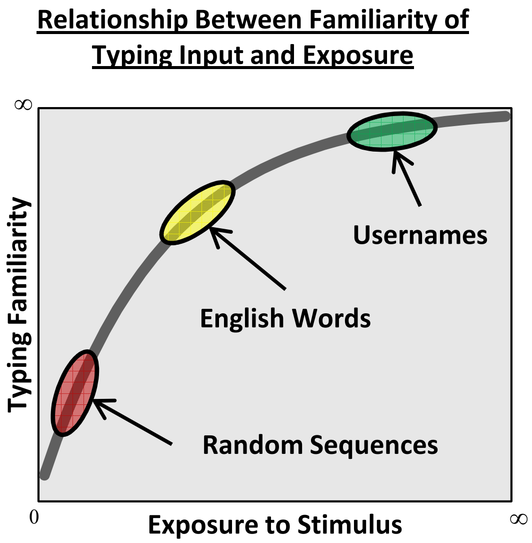


Figure 5.1: Typing Familiarity vs. Exposure to Stimulus.

5.2 Experimental Design

5.2.1 Collection System

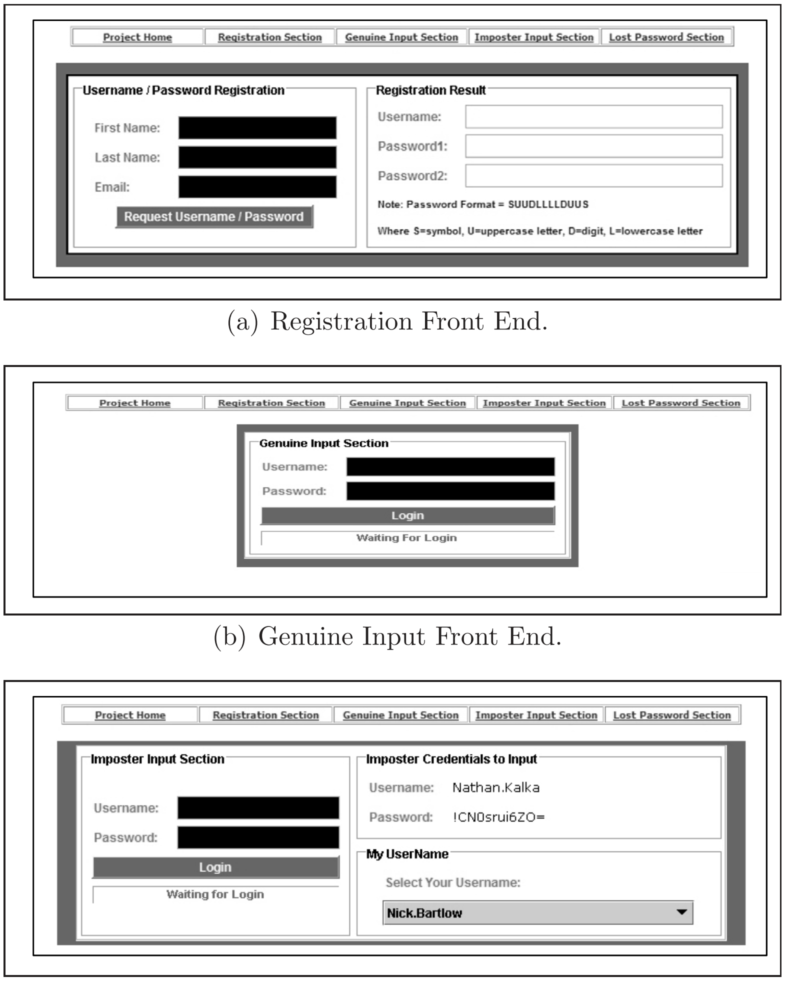
To achieve a web based architecture for the collection system, front-end client-side Java applets were developed in the NetBeans Integrated Development Environment (IDE) and designed to run within standard web browsers (Mozilla Firefox, Internet Explorer, Netscape) using the Sun Java Console. This offered both a tested open source platform as well as the bulk of the computation to occur in the client computers. On the server side, a MySQL database (also an open source product) was used to house the data. Therefore, once client-side computation was finished, the input was entered into the database via the previously established client-server connection [10]. It should also be noted that the use of a web-based collection system does not preclude application of the techniques used in this work in non web-based applications such as terminals and mail clients.

During the registration process, each user was given two sets of username / password credential sequences. The username remained the same across both sets of credentials and was of the form Firstname.Lastname with the first letter of each name capitalized. The first password was as an eight letter lowercase English word taken from a cryptographic dictionary attack list [122]. Examples of such passwords included computer and swimming. The second password, consisted of 12 randomly generated characters in a consistent pattern. The format of the pattern was as follows:

SUUDLLLLDUUS

where S is a special symbol, U is an uppercase letter, L is a lowercase letter and D is a digit. Examples of such passwords include +AL4lfav8TB= and _UC8gkum5WH. This pattern was not intended to elicit any specific behavior but only to allow for easy interpretation of potentially ambiguous symbols. The extra length of the pseudo-random passwords was incorporated to arrive at what are considered to be cryptographically strong passwords.

Beyond the structure of the input, the behavioral nature of this biometric scheme required a slightly more involved data collection process than what is typical in conventional physiological biometric systems. Most notably, one cannot simply compare genuine input of one user to genuine input from another user in order to establish an instance of imposter input. In this study, the passwords were different for each and every individual. Therefore, the collection tool required the development of two different user interfaces. One interface requests users to input the genuine credentials provided to them in the registration phase. The second interface requests users to input the credentials assigned to another user, gen-



Username / Password Registration

First Name: [redacted]

Last Name: [redacted]

Email: [redacted]

Registration Result

Username:

Password1:

Password2:

Note: Password Format = SUUDLLLLDUUS

Where S=symbol, U=uppercase letter, D=digit, L=lowercase letter

(a) Registration Front End.

Genuine Input Section

Username: [redacted]

Password: [redacted]

Waiting For Login

(b) Genuine Input Front End.

Imposter Input Section

Username: [redacted]

Password: [redacted]

Waiting for Login

Imposter Credentials to Input

Username: Nathan.Kalka

Password: !CN0srui6ZO=

My UserName

Nick.Bartlow

(c) Imposter Input Front End.

Figure 5.2: The three main pages for data collection (a) The registration front end. (b) The genuine input front end. (c) The imposter input front end.

erating an imposter authentication attempt. Figure 5.2 shows the registration (a), genuine input (b), and imposter input (c) front ends.

Within the genuine input front end, users were asked to input each of their credentials (username + password1 and username + password2) *5* times every day for approximately three weeks. The imposter input front-end was slightly different and can be seen in Figure 5.2 (c). In this front end, users were provided with credentials of a different user registered used. To avoid inadvertently collecting genuine data in the imposter section, user always provides his / her username from the “My UserName” field. Upon selection of the username, the data collection system populates the “Imposter Credentials” fields, automatically selecting a pair of credentials that is short on imposter data. In this way, the number of imposter sequences was kept balanced over the set of all the enrolled users.

At this point, the user simply logged in the same way as if this were his / her own genuine credentials. Pending a successful collection step, the new username / password pair appear in the window and the process repeats itself. Similar to the genuine input screen, users were asked to input a total of *10* imposter sequences per day.

5.2.2 User Supervision

As mentioned previously, we attempted to minimize the supervision component of the data collection. To that effect, users were only provided with a basic series of instructions and short video clips (for those inclined) to explain the functionality and expected use of the system. Although a large number of the students were included in the study (from our lab and university classes), there was no requirement to offer samples in the lab or in an on-campus setting. Additionally, many participants had no affiliation with the university and their data was collected from off-site locations throughout the internet. To that regard, no face to face guidance was provided to the participants.

5.2.3 Collection Results

At the time of final analysis, the database had a total of *53* users with over *10,000* total input sequences. After applying a minimum number of *15* valid sequences of each type of password, a total of *41* users and *8,882* username / password sequences were used. Out of the *8,882* total sequences, *5,094* were of type genuine and *3,788* were of type imposter. A summarized breakdown of the data collected for each user can be found in Table 5.1. The demographics of the database represent a fairly diverse population in many regards. The gender split was approximately half and half, ages ranged from mid-teens to individuals in their early 60's, and there was also a relatively diverse racial makeup. Perhaps most importantly, the typing ability of the population was also very diverse, ranging from the most inept "hunt and peck" typists to individuals with professional training / experience. In that regard, the classification algorithms not only were required to differentiate between professional and amateur typists but also between the members these two groups. Some of the users have been accustomed to working with multiple keyboard layouts. However, the type of keyboard used for data collection was not controlled. We know that a large percentage of subjects used desktop computers as well as laptops during the data collection. This is inevitably a source of noise but also demonstrates the ability of the technique to cope with variable hardware setups. The collection period lasted approximately one month.

Table 5.1: Data included in the experiment. The abbreviations E, R, and T correspond to English passwords sequences, Random passwords sequences, and total sequences. The final row includes the total number of users, average length for usernames and both types of passwords, and the total number of each type of input sequence collected.

UserID	Username	English	Random	Genuine Seq.			Imposter Seq.		
	Length	Password (E)	Password (R)	E	R	T	E	R	T
01	12	kathleen	@QZ4ozka1XE$	120	110	230	46	46	92
03	13	williams	]RR4axpe0WA>	048	050	098	46	46	94
04	14	rosemary	:LC6nvau900~	073	020	093	47	47	94
05	09	mitchell	>YH2avia0ER#	062	070	132	47	47	94
06	16	wolfgang	@WI7tjeb8WX}	117	108	225	47	47	94
07	13	aerobics	}YK2zquv9IQ+	083	076	159	47	47	94
08	10	firebird	.MS1suyf8MP^	053	052	105	47	47	94
09	11	fountain	(GC5idxx8TH{	051	051	102	47	47	94
10	18	caroline	^ZT7wyaz6JA[	016	017	033	47	47	94
11	12	zeppelin	!CN0srui6Z0=	122	119	241	46	46	92
12	19	bumbling	~XM6bywn6JL?	074	085	159	46	46	92
13	09	director	‘VA0snuv1HA:	090	104	194	46	46	92
14	12	gonzales	&ZL7yfjj0GK*	059	061	120	46	46	92
15	16	password	?KK6cvuc1NK|	059	088	147	46	46	92
17	14	business	;0I6vjog4QN>	053	058	111	46	46	92
18	12	fletcher	&UV1lkda5YH{	062	065	127	46	46	92
21	10	swimming	;K03ovpt4QC>	046	021	067	46	46	92
23	17	wheeling	;LB3chtu2YX‘	118	137	255	46	46	92
24	12	newcourt	:ZQ5grpx8VH;	027	028	055	46	46	92
25	15	snoopdog	,GG5ruft6IG+	052	045	097	46	46	92
26	07	colorado	$ZZ9ilfg9RJ(	043	025	068	46	46	92
27	14	homebrew	*TY1drmj7CR$	158	108	266	46	46	92
28	11	dolphins	[L02uqam8UI+	047	041	088	46	46	92
29	15	plymouth	)VS0iaka5WW!	048	042	090	46	46	92
30	14	broadway	;PG3xuel9LU}	054	049	103	46	46	92
31	13	woodwind	}EZ7mjjp4YM+	075	076	151	46	46	92
32	14	mountain	&BA1ishf0FC|	054	041	095	46	46	92
33	10	strangle	=BP8duim7IF@	091	089	180	46	46	92
34	12	strangle	<WN1zegb5RS$	077	074	151	46	46	92
35	13	princess	>GD0dgby6JU{	030	016	046	46	46	92
37	13	clusters	@UK8uudo5GS.	033	029	062	46	46	92
38	10	martinez	_UC8gkum5WH@	030	021	051	46	46	92
40	13	tacobell	[VB6jveb2PC~	015	017	032	46	46	92
43	12	baritone	&F04ovcv0VK!	028	029	057	46	46	92
44	14	frighten	$IP2ulld5QT@	053	051	104	46	46	92
46	12	starwars	$SP3lhkt1YX{	031	039	070	46	46	92
47	15	thompson	#P05dlfq0JW:	108	129	237	46	46	92
48	14	explorer	<JT5ocyi8TK=	099	099	198	46	46	92
49	15	elephant	&XQ5jwsp8KA]	080	080	160	46	46	92
51	13	springer	@SF5sjnd5EY/	017	015	032	46	46	92
53	12	sweatpea	^R08wxps1HI)	062	041	103	46	46	92
41	12.93	8	12	2,618	2,476	5,094	1,894	1,894	3,788

5.3 Authentication Algorithm and Features

As mentioned in the related work chapter, a plethora of algorithms have been investigated as candidates for authentication mechanisms throughout the history of keystroke dynamics research. The scope ranges from simple distance metrics between probe and gallery templates, to complicated multi-layer neural networks. In this work, we arrive at feature vectors containing raw hold times and inter-key delays and use the decision tree based Random Forests algorithm developed by Breiman [20] to classify input sequences. It is important to note that we are assuming a verification framework where an identity is claimed. This assumption leads to a two class problem; deciding whether an input sequence is genuine or imposter in nature. The following two subsections provide expanded descriptions of the Random Forest algorithm (including the nature of the training and testing) and the feature set in which classification was based.

5.3.1 Random Forests and Training / Testing Framework

An elegant and powerful algorithm, Random Forests is named after two main characteristics. One, it is is based on the development of a “forest” of decision tree classifiers, each being similar to a C5.0 decision trees developed by Quinlan [1]. Two, the method of generating the forests is based on the random sampling of features in the attribute space. The tree generation algorithm works as follows: each tree is grown based on a random sample selection of $\frac{2}{3}$ of the instance population. In each decision tree that populates the forest: nodes, branches, and leaves are generated by continuously choosing the feature that yields the best split of the data based on m randomly selected features. Sub-tree generation continues to the extent possible without pruning. Once all trees have been generated, new instances of feature vectors are passed through the trees of the forest and a voting process takes place to determine the classification result.

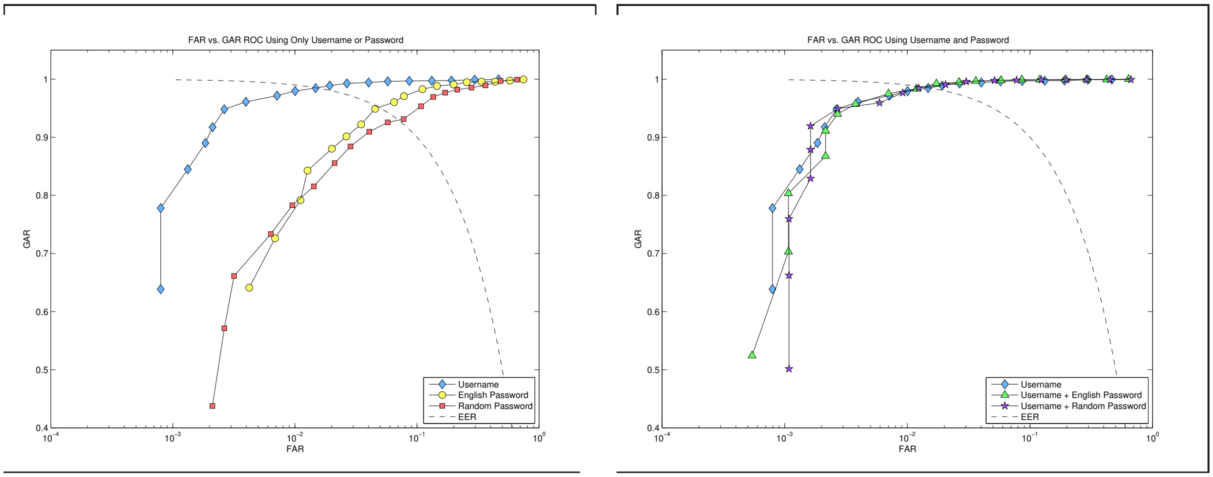
There are a number of attractive advantages Random Forests have over other machine learning algorithms. Our study pays particular attention to two of them. One, due to the $\frac{2}{3}$ sampling used to train each tree, the remaining $\frac{1}{3}$ so called out-of-bag (OOB) sample is used to test the classification performance. Therefore, training and test sets do not need to be explicitly separated and the estimated error results are said to provide conservative estimates of future performance. Two, the ability to define varying voting schemes allows for generating forests tailored to specific matching applications. For instance, a *10%-90%* voting scheme for genuine and imposter classes places particular emphasis on the minimizing the FRR, whereas a *90%-10%* scheme reverses the requirement, focusing attention on the FAR. This mechanism allows for the generation of Receiver Operating Characteristic (ROC) curves which describe an entire range of achievable performance characteristics relative to FAR and FRR's. Other learners typically generate only a single operating point along a ROC curve. We generated *19* Random Forests for every user, each with a different voting scheme with voting increments of *0.05*, ranging between *0.05-0.95* and *0.95-0.05*. Furthermore, for each forest, *500* trees were generated and the default value of parameter m (features to consider at each node split) was used. In this case, $m = \sqrt{X}$, or the square root of the total number of attributes in the feature space, (X), which was dependent on the particular input sequence. For instance, an eight character English password might have *8* hold times and *7* delays for a total of *15* attributes. In this example $m = 4 \approx \sqrt{15}$.

5.3.2 Feature Set

Although we previously investigated using aggregate statistics based on the keystroke hold times and delays of each input sequence such as averages, standard deviations, etc. in [12], this work utilizes the raw hold times and delays. In that regard, calculation of the feature vector for each sequence is arguably as simple as possible; no secondary calculation is necessary to arrive at the final feature vector. Given the requirement of shift key activity in the random passwords, it was possible that not every sequence had the same amount of hold times or delays. In this case, the longest feature vector was identified and all other sequences (both genuine and imposter) were padded with zeros to arrive at equal length vectors for each user's data set.

5.4 Experimental Results

In this section we look into three areas of interest; the performance of our keystroke dynamics system, the potential relationship between credential component familiarity and authentication results, and the credential hardening effect of applying keystroke dynamics to traditional username / password systems. We first demonstrate the performance of our keystroke dynamics system while varying the nature of algorithmic deployment and the input requirements in terms of credential components. First, to reiterate the behavioral nature of the system, genuine and imposter sequences were collected for each user in the system. Based on both types of sequences, *19* Random Forests were constructed for each user by varying the voting percentage required for classification. By doing so, we were able to calculate ROC curves and EER's for each user. Figure 5.3 shows the overall system ROC curves considering different credential components as input. These ROC curves were calculated by globally applying a Random Forest voting threshold to each user's dataset and averaging FAR and GAR's across the *41* users. In the figure, plot (a) shows the three curves which result from considering only single credential components. In other words, only the hold times and delays for the username, English password, or random password were used for model creation and testing. Plot (b) shows the results when hold times and delays for both the username and password components of the credential sets were used. As can be seen in (a), the English password outperforms the random password for most areas of the ROC curve while the username universally outperforms both passwords. This figure supports the hypothesis that performance is driven in part by familiarity of the credential set component used for authentication. In plot (b), there are select regions of the ROC curve where the addition of the Random password to the username increases performance over using the username alone. However, in most regions sole use of the username results in as good, if not better performance, compared to the the combination of both components of the credential set. Taking a more fine grained look into the results, Figure 5.4 displays the EER's for each of the *41* users in the study considering the same sets of credential components. Here we see the upper and lower bounds of the system's performance. The worst performing user for the English password and username is user *#14* having EER's of *11.36%* and *3.4%* respectively. User *#21* has the worst random password performance with an EER of *18.22%*. Approximately *25%* of the users tested (*10/41*) achieve a *0%* EER for usernames while *4* achieve this rate in English passwords. The lower bound for performance on random passwords is *0.96%* corresponding to user *#8*. If username and password components are combined, we find the maximum EER's for both types of passwords improves substantially falling below

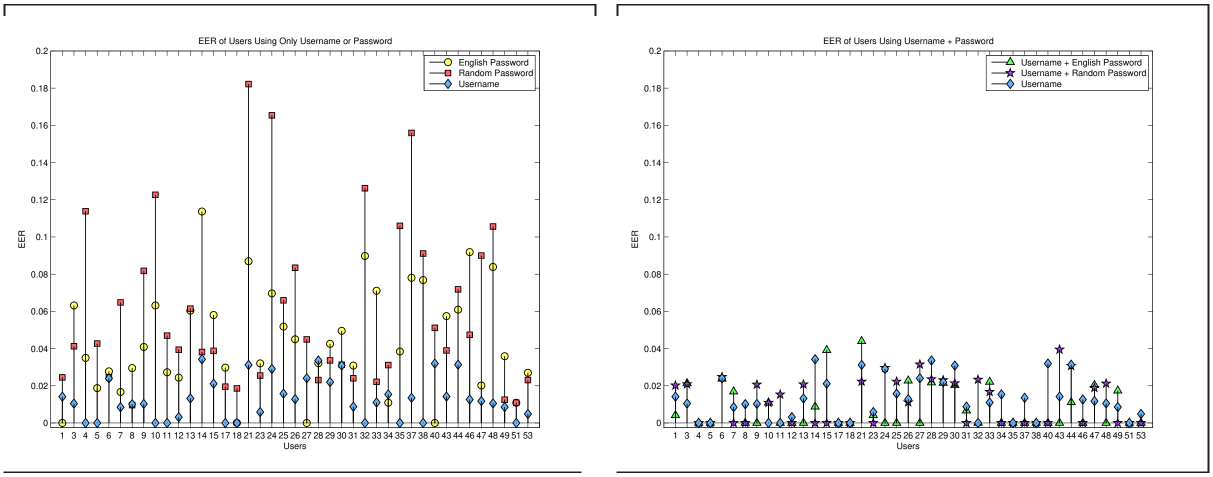


(a) FAR vs. GAR ROC using only username or password for authentication.

(b) FAR vs. GAR ROC using username and password for authentication.

Figure 5.3: System ROC curves considering different credential set input components.

5%. Here user *#21* has an EER of *4.4%* for English passwords and user *#43* has an EER of *3.9%*. Additionally, more users achieve an EER of *0%* with *23/41* for the combination of username and English password and *20/41* for username and random passwords. As demonstrated earlier in Figure 5.3, system wide performance can be generalized by applying a global voting threshold across all users. Optionally, one may choose to apply user specific voting schemes to optimize performance. Table 5.2 shows the difference in performance of applying global voting schemes vs. user specific voting schemes across the different components of the credential sets. Naturally, if system wide performance is characterized using voting schemes optimized to each user, error rates decrease. This notion is characterized by the last column in the table indicating the decrease in EER achieved from applying the global voting threshold to applying the user specific voting schemes. Once again, we see the trend that credential set components that are more familiar to users perform better. Here, usernames outperform both the English passwords and random passwords when only one



(a) EERs using only username or password for authentication.

(b) EERs using username and password for authentication.

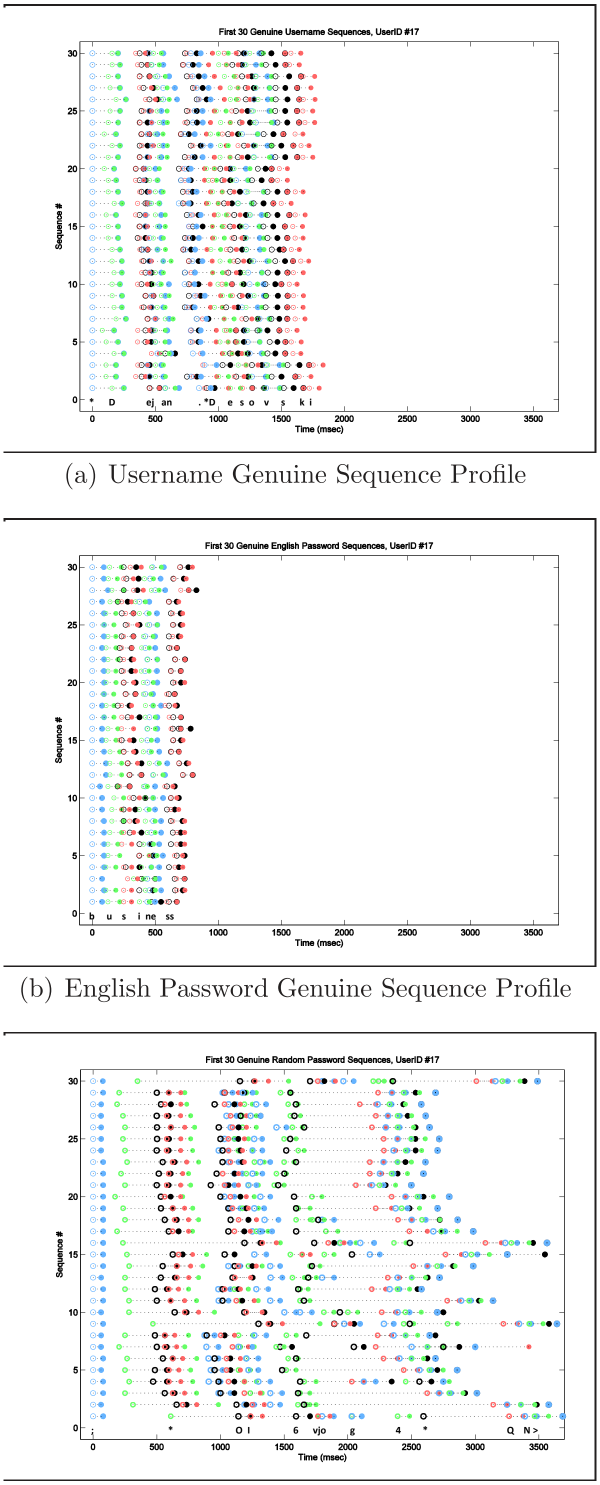
Figure 5.4: Comparison of EERs across users considering different credential set input components.

Table 5.2: System performance using global and user specific voting schemes.

Credential Set Component(s)	EER Global Scheme	EER User Specific Scheme	Difference
Random Password	6.796%	5.507%	1.289%
English Password	4.578%	3.997%	0.581%
Username	1.511%	1.284%	0.226%
Username + Random Password	4.581%	1.173%	3.408%
Username + English Password	3.138%	0.852%	2.286%

component is considered. Furthermore, incorporating both the username and English password yields the best performance result when looking at the user specific EER of *0.852%*. In one final examination of the notion of familiarity, we look into the genuine keystroke profiles for one user over the three individual components tested: the username, English password, and random password. Intuitively, we would expect to see less variability from sequence to sequence in components a user is more familiar than in those a user is less familiar with. Although space prohibits examining the profiles of every user for each sequence, Figure 5.5 shows the first *40* sequence profiles for user *#17*'s username, English password, and random password. In the three plots, keystrokes are defined by pairs of circles with matching colors. Here, open circles indicate a key down and filled circles correspond to the matching key up. Therefore, the time between the open circle and closed circle of the same color represents the hold time. The time between two open circles represents the delay between two adjacent keystrokes. For clarity, the typed input corresponding to the first sequence is labeled in each plot with *'s indicating a shift key was pressed. Not surprisingly, we see a greater deviation in the profile of each sequence in the random passwords than we do in the username and English password sequences. This is potentially a reason for the difference in performances seen in 5.2. Should the first letter of each portion of the username not been capitalized, even less inter-sequence variation may have been found in the usernames, shown in plot (a).

To conclude the analysis of the results, we demonstrate the credential hardening effect of applying our keystroke dynamics system to a classic username / password authentication system. To do so, we assume that imposters attempting unauthorized entry have obtained the password of the user in which they are imposing. Based on this assumption, the FAR or penetration rate of the imposter, in absence of keystroke dynamics, will be 100%. In



(a) Username Genuine Sequence Profile

(b) English Password Genuine Sequence Profile

(c) Random Password Genuine Sequence Profile

Figure 5.5: The first *40* genuine input sequences of user *#17* for username, English password, and Random password components.

other words, if the imposter knows the targeted users credentials he / she will always be able to type them in correctly, thereby gaining access. On the other hand, when our keystroke dynamics biometric is used to augment the system, correct content of the password is only a partial requirement; the imposter must also type the credentials with the same keystroke dynamics signature. Table 5.3 demonstrates the effect of this additional requirement assuming operation at performance rates derived through global application of Random Forests voting schemes across all users.

Table 5.3: Credential Hardening Effect with Globally Applied Voting Schemes.

	FAR(%)			FRR(%)		
Credential Set Component(s)	Before	After	Difference	Before	After	Difference
Random Password	100.00	6.80	↓ 93.20	0.00	6.80	↑ 6.80
English Password	100.00	4.58	↓ 95.42	0.00	4.58	↑ 4.58
Username	100.00	1.51	↓ 98.49	0.00	1.51	↑ 1.51
Username + Random Password	100.00	4.58	↓ 95.42	0.00	4.58	↑ 4.58
Username + English Password	100.00	3.14	↓ 96.86	0.00	3.14	↑ 3.14

The penetration rate into the system in terms of FAR decreases by *93.2%*, *95.4%*, and *98.49%* respectively for random passwords, English passwords, and usernames taken as singleton components. These rates are calculated by simply subtracting the keystroke dynamics FAR from the assumed *100%* penetration without the biometric augment. The higher security comes at a relatively low price as the associated increases in FRR are *6.8%*, 4.58%, and *1.51%* in random passwords, English passwords, and usernames. By applying user specific voting schemes we can achieve a greater credential hardening effect through the addition of the keystroke dynamics augment. Table 5.4 shows the decrease in system penetration in this case. Here, we see greater decreases in system penetration while also decreasing user inconvenience in terms of the FAR. Additionally, the improvement from applying keystroke dynamics on the username taken alone or combined with either type of password approaches the limits of performance with rates of penetration falling at *1.28%*, *1.17%*, and *0.85%* respectively. Similar to the globally applied voting schemes, this comes at an arguably low cost to user convenience.

Table 5.4: Credential Hardening Effect with User Specific Voting Schemes.

	FAR(%)			FRR(%)		
Credential Set Component(s)	Before	After	Difference	Before	After	Difference
Random Password	100.00	5.51	↓ 94.49	0.00	5.51	↑ 5.51
English Password	100.00	4.00	↓ 96.00	0.00	4.00	↑ 4.00
Username	100.00	1.28	↓ 98.72	0.00	1.28	↑ 1.28
Username + Random Password	100.00	1.17	↓ 98.83	0.00	1.17	↑ 1.17
Username + English Password	100.00	0.85	↓ 99.15	0.00	0.85	↑ 0.85

5.5 Discussion

Despite the continuing maturity of the field, many open problems remain with keystroke dynamics and credential hardening systems. This work does not claim to answer issues such as rigid model training requirements of operational environments, imposter mimicry, or generation of imposter data sequences. Regarding issues specifically visited in this work, a number of assumptions may need further explanation. Since two individual components of the credential sets considered required shift-key behavior to modify characters, each extracted feature vector consisting of hold times and delays need not have the same number of attributes as every other vector in a data set. As mentioned earlier, each sequence was back filled with zeros to ensure that the extracted feature vector for each sequence had the same length. Some may argue that this requires posteriori knowledge of the data, namely, the maximum length feature vector of all sequences in a data set. We would argue however, that this knowledge is superficial as a maximum keystroke limit per credential set component could be imposed without having ill effect on the user's experience or the performance of the classification algorithm. To justify the latter, the Random Forest algorithm will automatically weed out attributes with little information gain (zero-valued for virtually all instances) during decision tree generation.

Although this study admittedly does not consider user mimicry, we do emphasize that imposter keystroke "attacks" are considered "zero-effort" in that users were asked to type imposter data naturally. In other words, imposters made no effort to deviate from their normal typing patterns in an effort to more reliably emulate the targeted genuine sequence input. Similar to forgery in handwritten signatures, we assume that formal attempts beyond "zero-effort" attacks may result in decreases credential hardening effects. This notion was not considered primarily based on the fact that it would be prohibitively difficult to do so given the completely remote and unsupervised nature of the data collection effort. We feel the importance of these characteristics coupled with the increased size of the data set outweigh the importance of gathering data incorporating mimicry.

Even though the results presented in this work support the hypothesis that user familiarity with credential set components may drive keystroke dynamics performance, statistical tests of sequence uniformity can be applied to further quantify differences in familiarity. This could be achieved by fitting distributions to individual keystroke hold times and delays across all genuine sequences of a data set and applying statistical tests of uniformity such as Rayleigh, Rao, Neyman, etc. Should our hypothesis hold, keystroke hold times and delays from usernames and English passwords would more closely resemble uniform distributions than those resulting from random passwords. That said, modification of the username to all lowercase letters may be necessary.

Additionally, the authors are aware that although the size of the data sets included in the work is similar to those found in related academic efforts, the strength of the evidence supporting the hypotheses presented could be increased through application on larger test beds. To that regard, continuing data collection efforts are in place to make larger scale studies possible.

5.6 Chapter Summary

In this chapter we further established the viability of keystroke dynamics as a method of hardening traditional username / password credential sets in remote, unsupervised environments that lack restrictions on hardware use (selection of keyboard), similiar to what one may expect to find if applied to digitally "signing" evidence as it passes through a chain of entities. This hardening can be achieved by applying keystroke dynamics to individual components of a credential set or both simultaneously depending on the nature of the application environment and desired performance. Additionally, we investigated the notion that credential set component familiarity has an affect on keystroke dynamics performance. Namely, sequences that users are exposed to consistently and arguably familiar with such as usernames and English words offer better classification potential compared to undoubtedly unfamiliar sequences such as randomly generated strings. From a realistic application setting, this may be encouraging as despite their relative weakness, most users routinely select passwords in which they are familiar with such as family names and English words. This phenomenon may be in part due to more consistent typing signatures in genuine sequences corresponding to components in which users are more familiar. These conclusions are also encouraging as they should prove useful when applied to chain of evidence applications that will likely use typical username and password credential sets paired with biometrics for the purposes of user authentication.

This page intentionally contains only this sentence.

Chapter 6

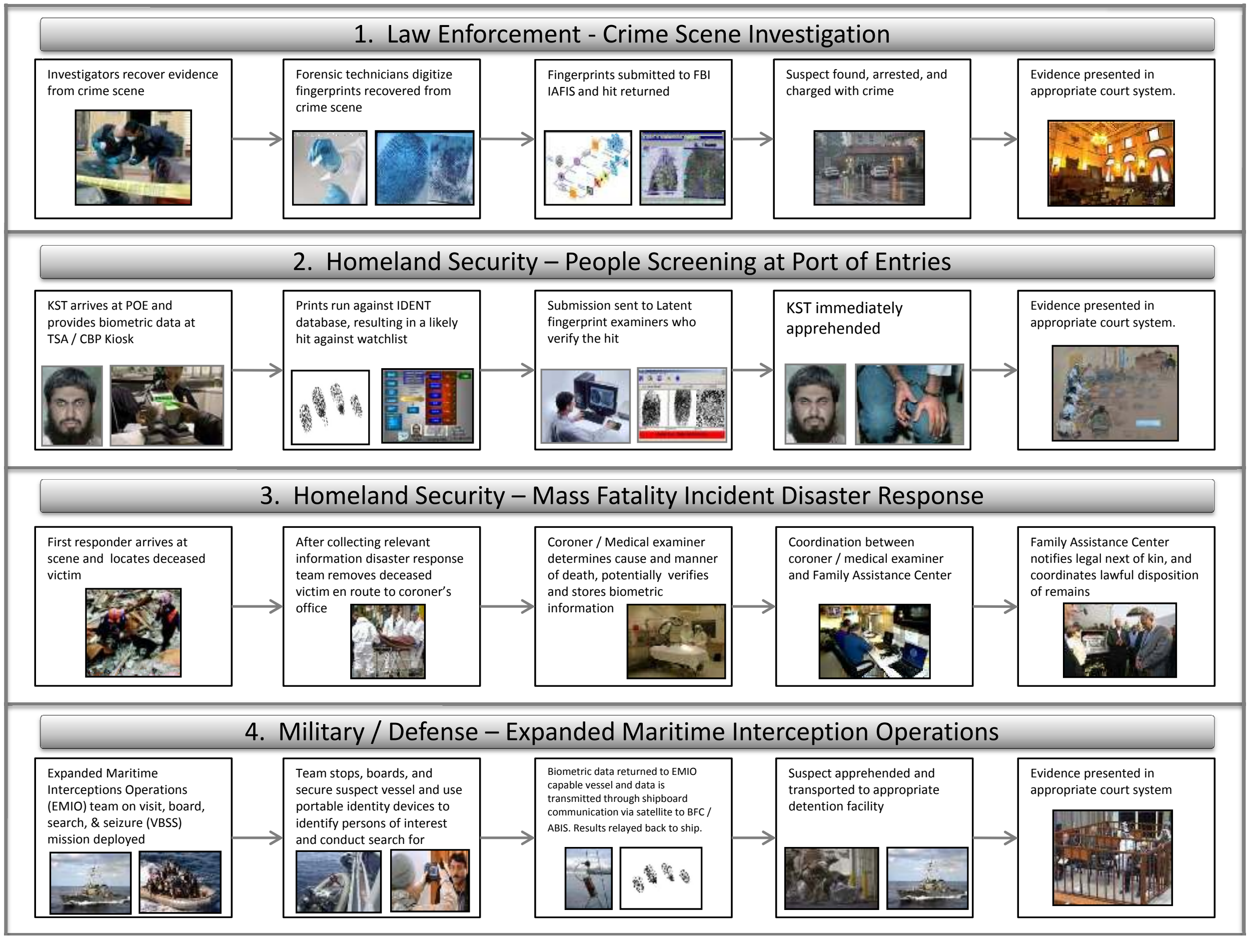
A Conceptual Framework for Digital Chains of Evidence

As stated in the introduction, the primary goal of this work was to develop a conceptual framework for creating a digital chain of evidence which allows for validation of evidence content, transmission, and acquisition source. Referring back to Chapter 2, we see that related work to this end has primarily identified the problem as opposed to providing solutions. This chapter describes a conceptual framework which shrinks the gap between the identified problem and an actual solution. We first offer four different motivating examples of scenarios where it may be appropriate to maintain a digital chain of evidence. Next, we provide a high level view of the conceptual framework which establishes the relationship between four major aspects of a digital chain of evidence including: security service requirements, associated security threats, mechanisms for dealing with threats, and subsequent capabilities. After examining the framework at a high level, we provide 9 use cases which conceptually outline how an implementation of the framework would work. Next, we investigate the success of the framework in offering validation of evidence content, transmission, and acquisition source by presenting a matrix which aligns security service requirements against, potential threats, mechanisms for countering threats, and subsequent capabilities. Having established the capabilities, we provide commentary on the agnostic nature of the framework which offers a desirable degree of flexibility for potential organizations that may use such a system. Finally, we conclude the chapter with a presentation of a specific instantiation of the framework based on the application of security mechanisms which have been discussed in previous chapters of this work.

6.1 Chain of Evidence Scenarios Motivating the Problem

Before presenting the conceptual framework for a digital chain of evidence, it is beneficial to further motivate its necessity by providing a number of examples where the framework could, or perhaps should be applied. As mentioned in the introduction, there is a wide range of scenarios which could make use of such a framework. In this subsection, we provide four examples of such scenarios. Figure 6.1 shows the four scenarios provided including: one example from a typical law enforcement perspective, two dealing with homeland security operations, and one based on military operations.

The first example is likely the most intuitive example of a scenario where a chain of evidence is established. Whether dealing with local, state, or federal law enforcement, we consider the example of crime scene investigation.



1. Law Enforcement - Crime Scene Investigation

Investigators recover evidence from crime scene

Forensic technicians digitize fingerprints recovered from crime scene

Fingerprints submitted to FBI IAFIS and hit returned

Suspect found, arrested, and charged with crime

Evidence presented in appropriate court system.

2. Homeland Security – People Screening at Port of Entries

KST arrives at POE and provides biometric data at TSA / CBP Kiosk

Prints run against IDENT database, resulting in a likely hit against watchlist

Submission sent to Latent fingerprint examiners who verify the hit

KST immediately apprehended

Evidence presented in appropriate court system.

3. Homeland Security – Mass Fatality Incident Disaster Response

First responder arrives at scene and locates deceased victim

After collecting relevant information disaster response team removes deceased victim en route to coroner's office

Coroner / Medical examiner determines cause and manner of death, potentially verifies and stores biometric information

Coordination between coroner / medical examiner and Family Assistance Center

Family Assistance Center notifies legal next of kin, and coordinates lawful disposition of remains

4. Military / Defense – Expanded Maritime Interception Operations

Expanded Maritime Interceptions Operations (EMIO) team on visit, board, search, & seizure (VBSS) mission deployed

Team stops, boards, and secure suspect vessel and use portable identity devices to identify persons of interest and conduct search for

Biometric data returned to EMIO capable vessel and data is transmitted through shipboard communication via satellite to BFC / ABIS. Results relayed back to ship.

Suspect apprehended and transported to appropriate detention facility

Evidence presented in appropriate court system

Figure 6.1: Four Scenarios Dealing with Biometric Data Where a Digital Chain of Evidence Could Be Applied.

In this case, a hypothetical crime has taken place and investigators reach the scene where the crime occurred. After establishing that no immediate threats remain at the scene, the investigation begins. As a customary part of such an investigation, members of the investigation team will typically attempt to collect fingerprints from individuals involved in committing the crime. For the purposes of the example, we assume prints are discovered and collected by members of the team. After the on scene investigation is complete, the team members would typically send lifted prints to a crime lab where the prints would be converted from physical form to electronic form. After that, the fingerprints might be submitted to a system such as FBI's IAFIS which may return a hit against the criminal database. In this case, police officers would attempt to locate and apprehend the suspect. After apprehension and arrest, the suspect would be charged, fingerprinted, and placed in a detention facility. Eventually the case may proceed to trial where the prosecuting team could provide evidence that the fingerprints lifted from the scene matched those in the IAFIS database which may lead to the conviction of the defendant. However, in order to get to that point, we recall the discussion from Chapter 2 which outlined the challenges of those involved in the United States criminal justice system. Breitman states that "Under the Federal Rules of Evidence, that standard of proof requires only that the party offering an item of physical evidence introduces such proof as is "sufficient to support" a finding that the [item in question] is what its proponent claims: Federal Rule of Evidence, 901(a)" [21] [58]. Additionally, he points out that "the ultimate issues of authentication and condition are left for the jury. If the proponent's evidence is facially sufficient to support a finding of authenticity, then all other issues such as credibility and probability are left for the jury [21]." Taking these accounts into consideration, the onus of establishing a burden of proof in a criminal justice system falls on many shoulders. It is possible that the evidence the prosecution is relying on has passed through a multitude of entities including crime scene investigators, lab technicians, evidence clerks, etc. With this in mind, it is the burden of the prosecution to prove the integrity of the chain of evidence was maintained.

The second example involves homeland security people screening activities which occur daily at U.S. Port of Entries (POEs). Everyday, thousands of individuals enter the United States through land, sea, and air POEs. One of the responsibilities of the Department of Homeland Security is to prevent dangerous people from entering the U.S. Given this goal, DHS runs programs such as US-VISIT, NEXUS, SENTRI, etc. which keep track of non-citizens entering the country. Among other information, such systems rely on biometric data to ensure travelers are not known or suspected terrorists (KSTs) or other individuals who may be considered dangerous. This example describes how a KST might be apprehended when attempting to enter the country through a POE supported by US-VISIT. We assume a KST arrives at a POE and is forced to proceed through a non-citizen lane where he is subject to US-VISIT processing. At that point, the KST presents a machine readable travel document (MRTD) to a border inspection officer. Assuming the system is not familiar with the information in the MRTD (the MRTD hasn't been seen by the system), a live-scan ten-print fingerprint submission as well as a digital photograph are collected from the KST. The border inspection officer submits the fingerprints to the IDENT database through US-VISIT where the prints are searched against a database of fingerprint of KSTs as well as other persons of interest. Next, the system registers a likely hit, which triggers both the set of prints collected at the POE and the prints matched in the database to be transmitted to latent fingerprint examiners supporting US-VISIT. The latent fingerprint examiners verify the match and the result is sent back to the kiosk of the border inspection officer. At that point, the KST might be taken to secondary screening or apprehended and detained immediately. From that point on, the KST could be charged and tried in an appropriate court system for any outstanding crimes committed. In this situation, the evidence has already traveled across a number of entities and will likely pass through more before it is used in the appropriate court system. While the court system may or may not be in the United States, it is safe to assume proving the integrity of the chain of evidence was maintained will be required in order to admit the digital fingerprints as evidence.

The third example falls outside any criminal justice system and the court of law. Also dealing with homeland security operations, this example has to do with mass fatality disaster response. Situations such as natural disasters including hurricanes, floods, etc. often cause tragedies resulting in mass loss of life. In situations such as these, teams of first responders arrive at the scene of the disaster and first conduct search and rescue operations in an attempt save the lives of individuals injured by the disaster. Although, the situations of the particular disaster will dictate different time frames, the search and rescue operations eventually turn into search and recover operations where responders attempt to recover the remains of deceased victims of the disaster such that families can be notified of there loss and the remains can be lawfully disposed to next of kin. Whether search and rescue or search and recover, responders in mass fatality disasters are responsible for carefully maintaining a chain of custody (evidence). According to “Mass Fatality Incidents: A Guide for Human Forensic Identification,” distributed by The Department of Justice, mass fatality incident teams can include [106]:

- Medical examiners / coroners

- Forensic anthropologists
- Odontologists
- Police crime scene investigators
- Forensic photographers
- Evidence technicians
- Scribes / notetakers

The same report outlines a number of requirements that the chain of evidence should document. As part of all data collected at such a scene, biometric data of the deceased is collected which may be used to help establish the identity of the remains. This can include fingerprints, dental samples, facial photographs, etc. Eventually this information will be transferred with the remains to the medical examiners / coroners who will attempt to establish the cause of death and identity of the individuals. Once they have been established the cause of death, the information is transferred to family assistance centers who locate and notify next of kin and coordinate the lawful disposition of remains. For obvious reasons, it is extremely important to provide accurate information to affected family members and prevent to the greatest degree possible, errors in notification process due to failures of the chain of evidence. As seen already, the number of entities in this chain may approach double digits.

The final example we will provide to motivate the problem relates to military / defense operations. In particular, Expanded Maritime Interception Operations (EMIO) conducted by the U.S. Navy. In such a scenario, a properly equipped Naval vessel is deployed on a visit, board, search, and seizure mission (VSSS). During the operation, the naval vessel will stop, board, and secure the suspect vessel and use portable biometric devices to collect biometric data of persons of interest on the suspect vessel. This information is returned to the EMIO capable naval vessel which subsequently transmits the biometric data back to the Department of Defense (DOD) Biometric Fusion Center (BFC). The BFC submits the prints for processing through the Automatic Biometric Identification System (ABIS) which may return a hit indicating the individual of interest is a KST. These results are relayed back to the naval EMIO capable vessel who transmits the information wirelessly to the EMIO team on the suspect vessel. At that point, the EMIO team apprehends the KST and transports the KST to an appropriate detention facility. Later on, members of agencies such as the Naval Criminal Investigation Service (NCIS) may charge and try the KST in an appropriate court system. Once again, in order to maximize the possibility of having the evidence viewed as admissible and authentic, the integrity of the chain of evidence must be maintained. In this case, the evidence minimally passed through four entities and in all likelihood will be subject to further analysis before final presentation in court. There are likely many more examples in which a chain of evidence must be established and maintained that have not been included in this chapter. However, the four examples presented paint a picture of the broad applicability and complexity of the problem that is considered in this work. With this in mind, the following section presents the conceptual framework for establishing and maintaining a digital chain of evidence in systems dealing with biometric information.

6.2 High Level View of the Framework

Instead of immediately delving into the application of a framework which describes the specific use cases of establishing and maintaining a digital chain of evidence, it is useful to examine the framework from a higher level. This section adopts this approach by considering the requirements of a chain of evidence, threats which might jeopardize the chain, mechanisms which can be used to combat the threats, and capabilities which result from the mechanisms. As mentioned in the introduction it is possible to consider the chain of evidence problem as a multi-component validation problem broadly including validation of evidence content, transmission, and acquisition source. However, it is useful to think of this validation problem in terms of computer security attributes. Computer security attributes are traditionally broken down into three categories: confidentiality, integrity, and availability. Here, confidentiality refers to the concealment of information or services, integrity refers to trustworthiness of data or resources (sometimes referred to as authenticity), and availability refers to the ability to use the information or resource provided [114, 17]. While availability would be an issue for systems establishing and maintaining a digital chain of evidence, it is outside of the scope of this work. Therefore, we focus only on issues related to confidentiality and integrity. Each of these three security attributes are then subject to a series of security threats. While we save the discussion of these specific threats for Section 6.4, it is important to note that security mechanisms can be used to combat the threats. In particular, the conceptual framework utilizes mechanisms described in previous sections of this work including biometric watermarking, digital hardware fingerprinting, biometric systems, and cryptography.

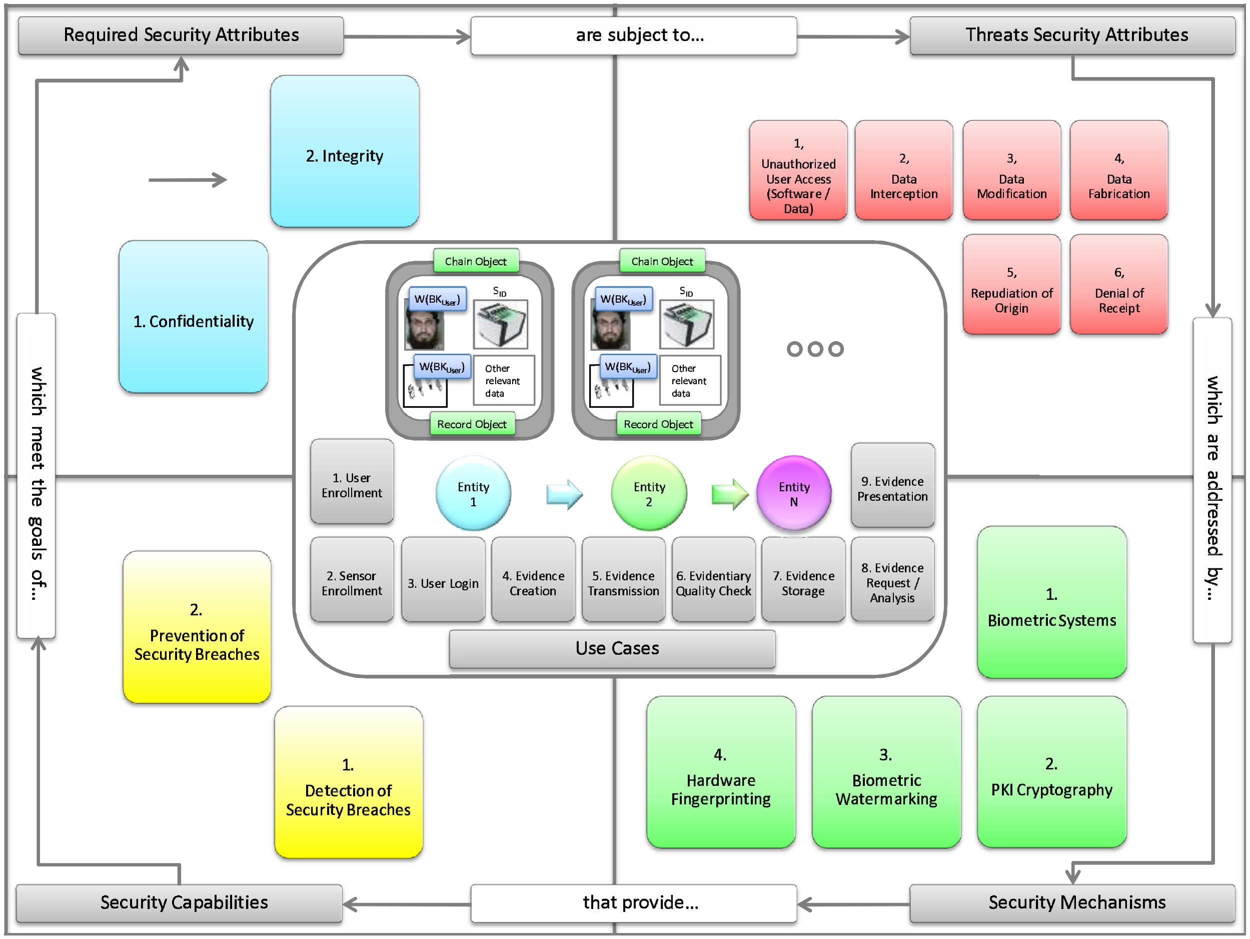


Figure 6.2: High Level View of Framework for Digital Chain of Evidence.

Finally, through the use of these security mechanisms, one is able to provide security capabilities. These capabilities include the ability to prevent, detect, and recover from security threats. In the chain of evidence application, we are only concerned with prevention and detection. The picture comes full circle when we consider that security capabilities meet the goals of required security services. Figure 6.2 provides a graphical representation of this high level conceptual framework. Much like the discussion, the graphic ties together the four different security concepts: required security attributes, security threats, security mechanisms, and security capabilities. Furthermore, it shows how the digital chain of evidence and its associated use cases sit at the center of these relationships.

6.3 Framework Use Cases

Having a high level understanding of the factors that must be considered in the development of the conceptual framework, we propose a digital chain of evidence which is composed of 7 different use cases. It is important to note that our proposed conceptual framework does not contain low level implementation details. Rather, it only provides a level of depth which should be sufficient for developing specific implementation details. With that in mind, one could argue for fewer or additional use cases in the framework depending on the depth one is considering. Fortunately, as we will see in Section 6.5, the level of depth we consider affords a level of agnosticism and flexibility which is desirable for broad application. The following subsections provide descriptions of how the 7 different use cases would be carried out should the framework be applied.

6.3.1 Use Case Terms

Before examining the different use cases within the proposed framework, it is beneficial to define terms and acronyms used the in the use case figures. Table 6.1

Term	Acronym	Definition
Evidence User	User	A user of the chain of evidence represents any individual who is capable of generating, viewing, or analyzing evidence. This can include CSI investigators, POE inspection officers, latent fingerprint examiners, lab technicians, evidence clerks, attorneys, etc.
Evidence Management System	EMS	The evidence management system is the software system which would be used to store and maintain evidence. Evidence management systems can be run by federal, state, or local organizations.
PKI Certificate Authority	PKI CA	The PKI Certificate Authority is responsible for managing public and private key pairs from users of the evidence management systems as well as evidence management systems themselves.
Enrollment Request of a User	ER_{USER}	A request for enrollment either with a PKI CA or an EMS by the user.
Enrollment Request of a EMS	ER_{EMS}	A request for enrollment either with a PKI CA by the EMS.
Credential Set of a User	CS_{USER}	The credential set of a user. This would potentially include a username and password. As well as a PKI digital certificate.
Biometric Signature of a User	BK_{USER}	The biometric signal of a user which is simply a feature vector (in binary from). This can potentially come from any biometric system relying on an arbitrary modality.
Biometric Watermark of a User	$W(BK_{USER})$	This is used to represent the finished process of watermarking an image with the biometric

PKI Key Pair of a User	PK$_{USER}$	This is represents a typical PKI key pair consisting of a public key which is openly published and a private key known only to the user.
PKI Key Pair of an EMS	PK$_{EMS}$	This is represents a typical PKI key pair consisting of a public key which is openly published and a private key known only to the EMS.
Encryption of Data for an EMS by a User	E(PK$_{USER}$,PK$_{EMS}$)	This represents that data has been encrypted using the PK$_{USER}$ and PK$_{EMS}$ by the user, for an EMS. In other words a data object has been encrypted through PKI using the private key of PK$_{USER}$ and the public key of PK$_{EMS}$.
Decryption of Data by an EMS from a User	D(PK$_{USER}$,PK$_{EMS}$)	This represents that data has been decrypted using the PK$_{USER}$ and PK$_{EMS}$ by an EMS, from data encrypted by a user. In other words a data object has been decrypted through PKI using the private key of PK$_{EMS}$ and the public key of PK$_{USER}$.
Sensor ID	S$_{ID}$	The unique identification number for a biometric sensor / capture device stored by an EMS.
Sensor Reference Pattern	S$_{RP}$	The unique noise reference pattern stored by an EMS (generated during sensor enrollment).
Chain ID	C$_{ID}$	The unique identification number for a Chain Object stored by an EMS.
Record ID	R$_{ID}$	The unique identification number for a Record Object within a Chain Object.

Record Object	Record Object	A Record object is a data structure which contains various types of evidence which would be part of a chain of evidence. Such data would include images from multiple biometric modalities, sensor information used to capture images, and any other data relevant to a chain of evidence (i.e. user names, time of evidence collection, geographic location of evidence collection, collection notes, etc.).
Chain Object	Chain Object	A chain object is a data structure which represents a linked list of record objects. This is the object is used to log creation, movement, and analysis of evidence (stored in record objects) throughout the users in a chain of evidence.
Sensor Database	Sensor DB	A database maintained by an EMS which keeps track of information on all biometric capture devices fielded by an organization. Information would include serial numbers, model numbers, and reference templates used for digital hardware fingerprinting.
User Database	User DB	A database maintained by an EMS which keeps track of information on all users enrolled in the system. This would include information such credential sets and biometric keys.
Chain Database	Chain DB	A database maintained by an EMS which keeps track of all chains of evidence managed by the system.
Authentication Response for a User	AR_{USER}	The credential set / biometric authentication response from the biometric system accesses a User DB stored by an EMS.
Evidence from a Subject	EV_{SUB}	The evidence captured from a live subject or bodily remains. This primarily refers to biometric images.
Validation Response	VR	The validation response which either verifies a biometric watermark or a sensors noise residual.

Integrity Report from a Chain of Evience	IR_{COE}	A report automatically generated by an EMS which iteratively presents all validation efforts of a Chain Object. This report is the basis for proving the integrity of a chain of evidence.

Table 6.1: Terms, Acronyms, and Definitions in the Framework Use Cases

Using the terms and entities described in 6.1, we have developed a framework consisting of 9 use cases including: user enrollment, sensor enrollment, user login, evidence creation, evidence transmission, evidentiary quality check, evidence storage, evidence request / analysis, and evidence presentation. These 9 use cases are described in the following subsections.

6.3.2 User / EMS Enrollment

The first use case in the conceptual framework involves the enrollment of the EMS and users with a PKI Certificate Authority (PKI CA) and the enrollment of users with an EMS. This process is pictured in Figure 6.3. The process would be initiated from a newly installed EMS submitting an enrollment request to at PKI CA. After the PKI CA received the request from the EMS, it would generate and return a PKI pair PK_{EMS} following standard PKI enrollment procedures. While a discussion of standard PKI procedures is outside the scope of this work, a summary of such procedures can be found in Appendix B. At this point, users could enroll into the EMS first by sending an enrollment request to the PKI CA. Once again, the PKI CA would return a PKI pair PK_{USER} following standard PKI procedures. Next, the user would have to enroll with the EMS itself. This would require the user to submit his credential set, CS_{USER} and a biometric key BK_{USER} (consisting of a biometric feature template). At that point the EMS would store both CS$_{USER}$ and BK$_{USER}$ in the EMS User DB. Note, as seen in 6.1, CS$_{USER}$ would include a username, password, and digital certificate. The BK$_{USER}$ would be used for two purposes. First, to control access of the user when logging into the EMS and requesting specific evidence records. Second, to watermark evidence either created or viewed by the user. In subsequent use cases, we will see how the watermark would serve as an electronic signature for the user. In principle, any type of biometric could be used for this process.

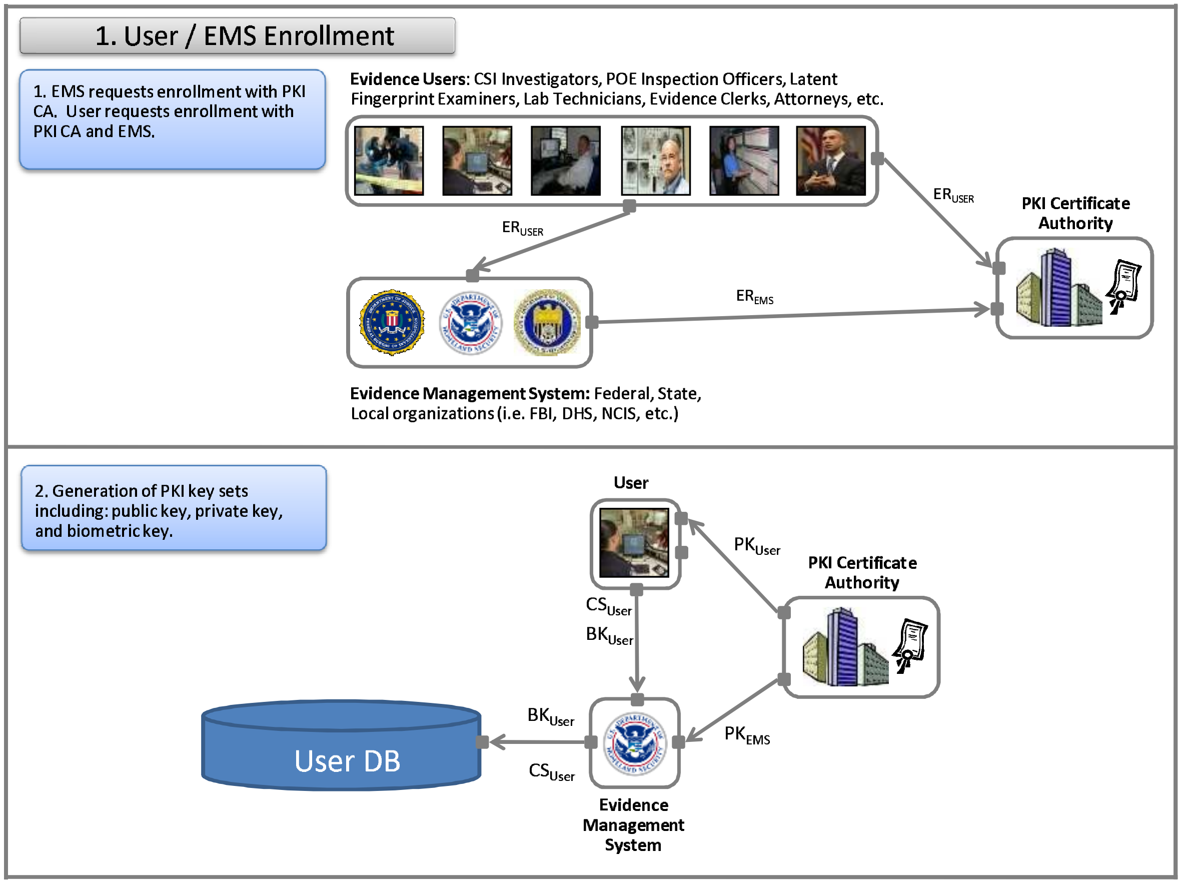
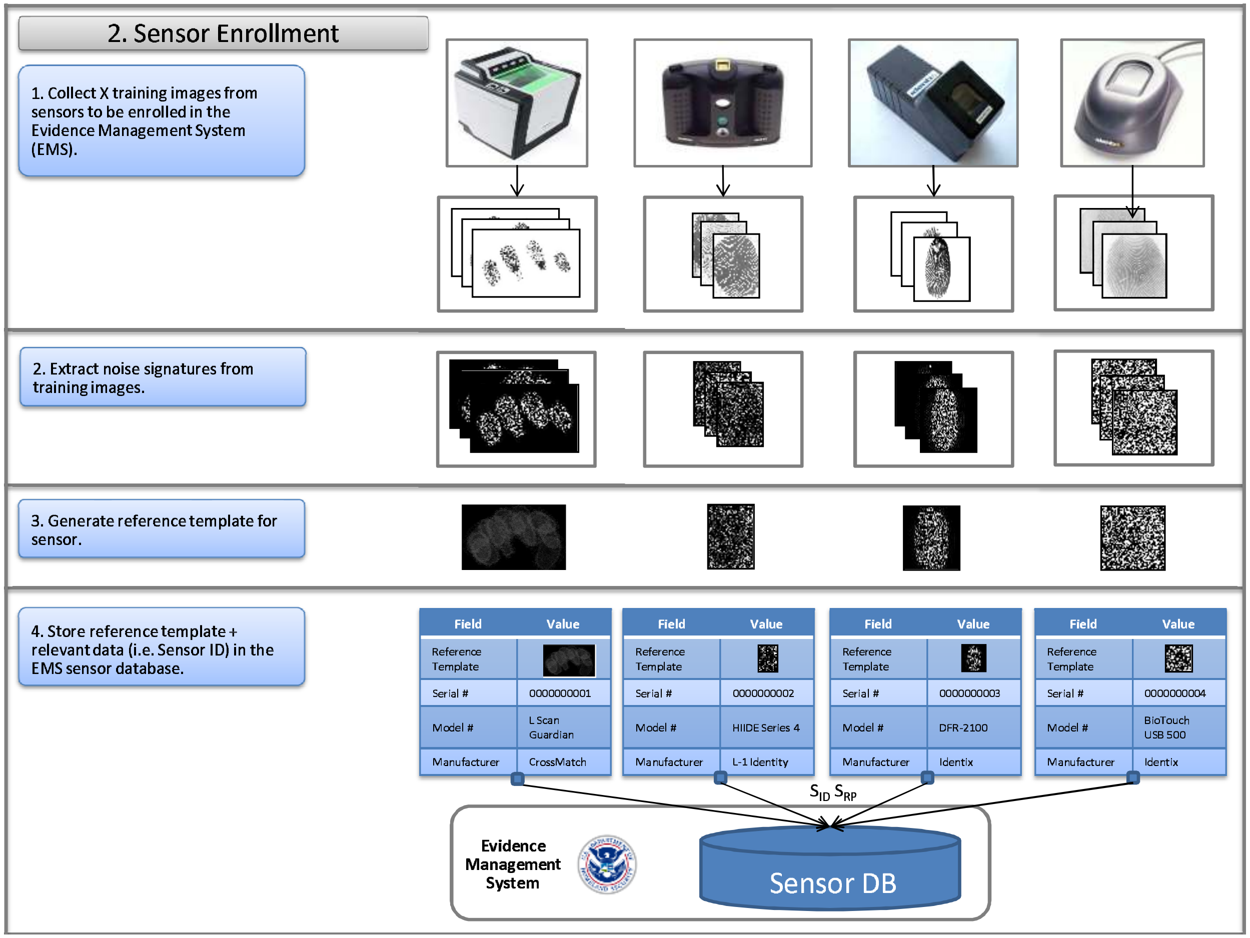

1. User / EMS Enrollment
1. EMS requests enrollment with PKI CA. User requests enrollment with PKI CA and EMS.
Evidence Users: CSI Investigators, POE Inspection Officers, Latent Fingerprint Examiners, Lab Technicians, Evidence Clerks, Attorneys, etc.
ER$_{USER}$
ER$_{USER}$
PKI Certificate Authority
ER$_{EMS}$
Evidence Management System: Federal, State, Local organizations (i.e. FBI, DHS, NCIS, etc.)
2. Generation of PKI key sets including: public key, private key, and biometric key.
User
PK$_{User}$
PKI Certificate Authority
CS$_{User}$
BK$_{User}$
BK$_{User}$
User DB
CS$_{User}$
PK$_{EMS}$
Evidence Management System

Figure 6.3: User Enrollment Use Case.

6.3.3 Sensor Enrollment

The next use case involves enrolling sensors or biometric acquisition devices (sensors) in the EMS sensor database. Figure 6.4 shows a visual depiction of this process. Before a device is fielded by the organization maintaining the EMS in question, it is first necessary to generate a reference patter, S_{RP} , which is characterized by the unique noise present in the device. Much like the experiments in Chapter 4, a series of training images would be collected by the sensor in the first step. In the next step, noise residuals would be extracted from each of the training images. The third step would then create a reference pattern for the device using the noise residuals from each of the training images (perhaps through an averaging operation). Finally, S_{RP} and relevant information such as the Sensor's unique ID, S_{ID} , would be stored in the Sensor DB of the EMS in question.



2. Sensor Enrollment

1. Collect X training images from sensors to be enrolled in the Evidence Management System (EMS).

2. Extract noise signatures from training images.

3. Generate reference template for sensor.

4. Store reference template + relevant data (i.e. Sensor ID) in the EMS sensor database.

Field	Value
Reference Template	
Serial #	0000000001
Model #	L Scan Guardian
Manufacturer	CrossMatch

Field	Value
Reference Template	
Serial #	0000000002
Model #	HIIDE Series 4
Manufacturer	L-1 Identity

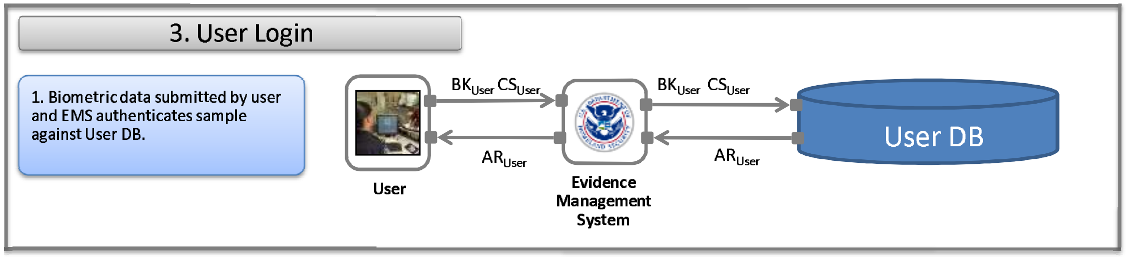
Field	Value
Reference Template	
Serial #	0000000003
Model #	DFR-2100
Manufacturer	Identix

Field	Value
Reference Template	
Serial #	0000000004
Model #	BioTouch USB 500
Manufacturer	Identix

Figure 6.4: Sensor Enrollment Use Case.

6.3.4 User Login

The next use case considered involves a user logging into an EMS. Access control to an EMS should be tightly monitored. With that in mind, it would be prudent to protect authenticate users not only with username and password credential sets, CS_{USER} , but also with biometric authentication. The distribution of user credential sets (CS_{USER}) and biometric keys (BK_{USER}) was taken care of in the first use case. At this point, the user would provide CS_{USER} to the EMS as well as his biometric sample from which his (BK_{USER}) would be extracted. Note this communication, as well as all communication between entities would be secured by PKI cryptography as outlined in Appendix B. Then, the EMS checks the credentials and biometric feature templates against the enrolled data in the User DB. Pending a positive result in the authentication response (AR_{USER}), the user would be allowed to access the system. Figure 6.5 displays this process. In the event of mismatched credentials or a failure to match the biometric sample provided against the gallery sample, the system would deny the EMS would deny the user access.



1. Biometric data submitted by user and EMS authenticates sample against User DB.

Figure 6.5: User Login Use Case.

6.3.5 Evidence Creation

Perhaps one of the most important use cases is that of evidence creation. This takes place when the first user in a chain of evidence prepares evidence to be submitted to an EMS. After collecting some form of biometric data and other relevant information from a subject (EV_{SUB}), the user submits a biometric sample from which his BK_{USER} is extracted (by the user's EMS interface). This BK_{USER} is then used to watermark the biometric data creating $\mathrm{W}(\mathrm{BK}_{USER})$. At that point, the watermarked biometric data and other relevant information including the sensor information would be bundled into a Record Object and subsequently packaged into a new Chain Object. Note, space has been allocated for the storage of other relevant data which might include names, organizations, discovery dates, etc. However, this work does not focus on dealing with this type of textual data. Finally the chain object would be encrypted using the PKI pairs of the user and the EMS. After this process is complete, the encrypted Chain Object would be ready for submission to the EMS. This process is depicted in Figure 6.6. Note this process would not need to create a new Chain Object every time. Rather the use case would also apply to the creation of new Record Objects to be added to existing Chain Objects.

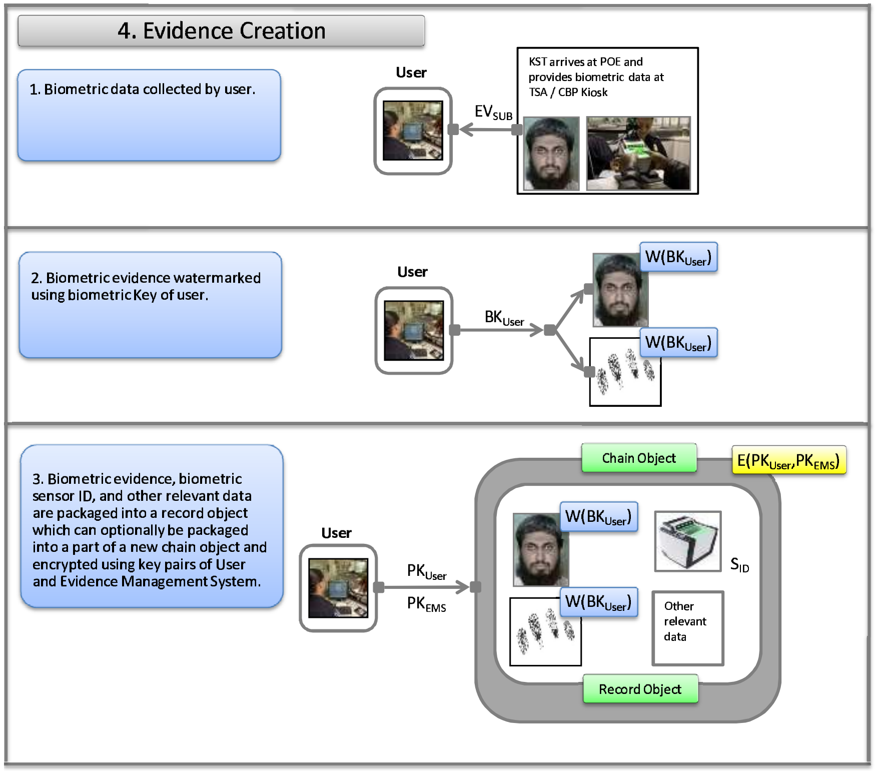
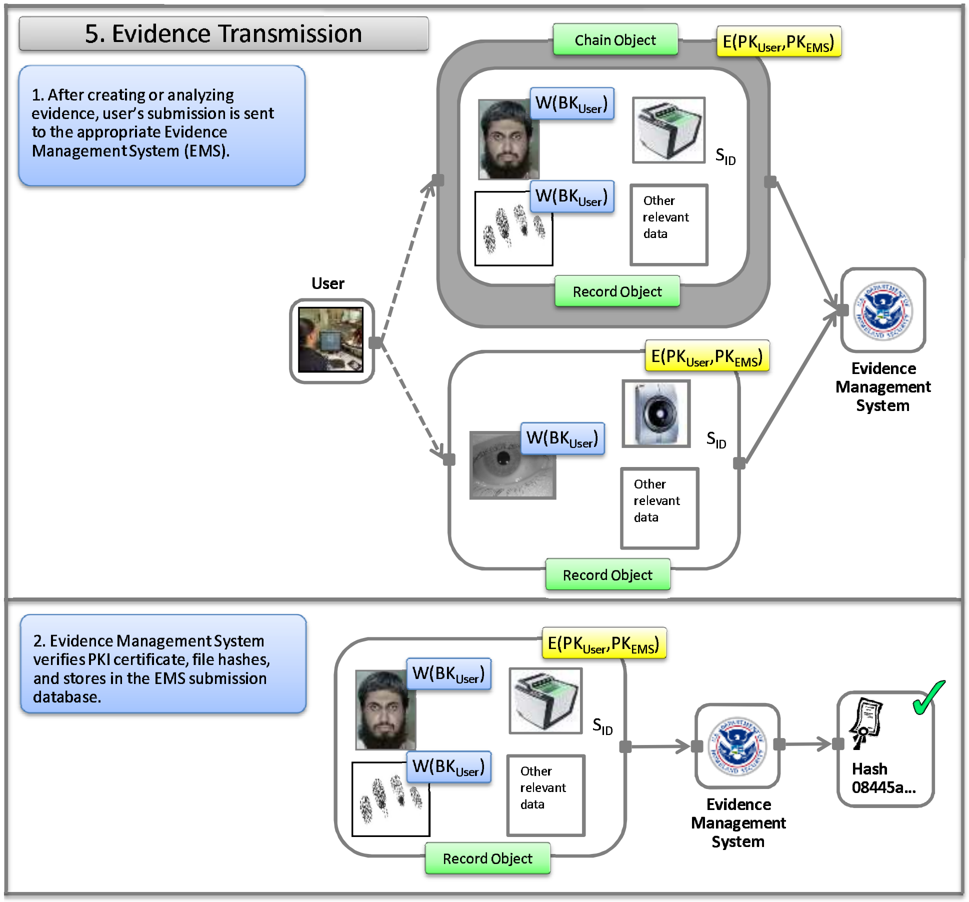

4. Evidence Creation
1. Biometric data collected by user.
User
EV$_{SUB}$
KST arrives at POE and provides biometric data at TSA / CBP Kiosk
2. Biometric evidence watermarked using biometric Key of user.
User
BK$_{User}$
W(BK$_{User}$)
W(BK$_{User}$)
3. Biometric evidence, biometric sensor ID, and other relevant data are packaged into a record object which can optionally be packaged into a part of a new chain object and encrypted using key pairs of User and Evidence Management System.
User
PK$_{User}$
PK$_{EMS}$
Chain Object
E(PK$_{User}$,PK$_{EMS}$)
W(BK$_{User}$)
S$_{ID}$
W(BK$_{User}$)
Other relevant data
Record Object

Figure 6.6: Chain of Evidence Creation Use Case.

6.3.6 Evidence Transmission

The process of evidence transmission is not unlike transmission of any other data across a network in that an evidence object (new Chain or Record object) would be sent from a user to an EMS. However, the important second step in the evidence transmission process would be to verify that the evidence was not altered during transmission. Aside from typical internet package transmission protocols, an additional check would be applied by using techniques from PKI cryptography as outlined in Appendix B. In this case, mismatched file hashes would indicate a change in the payload. This procedure is depicted in Figure 6.7.


5. Evidence Transmission

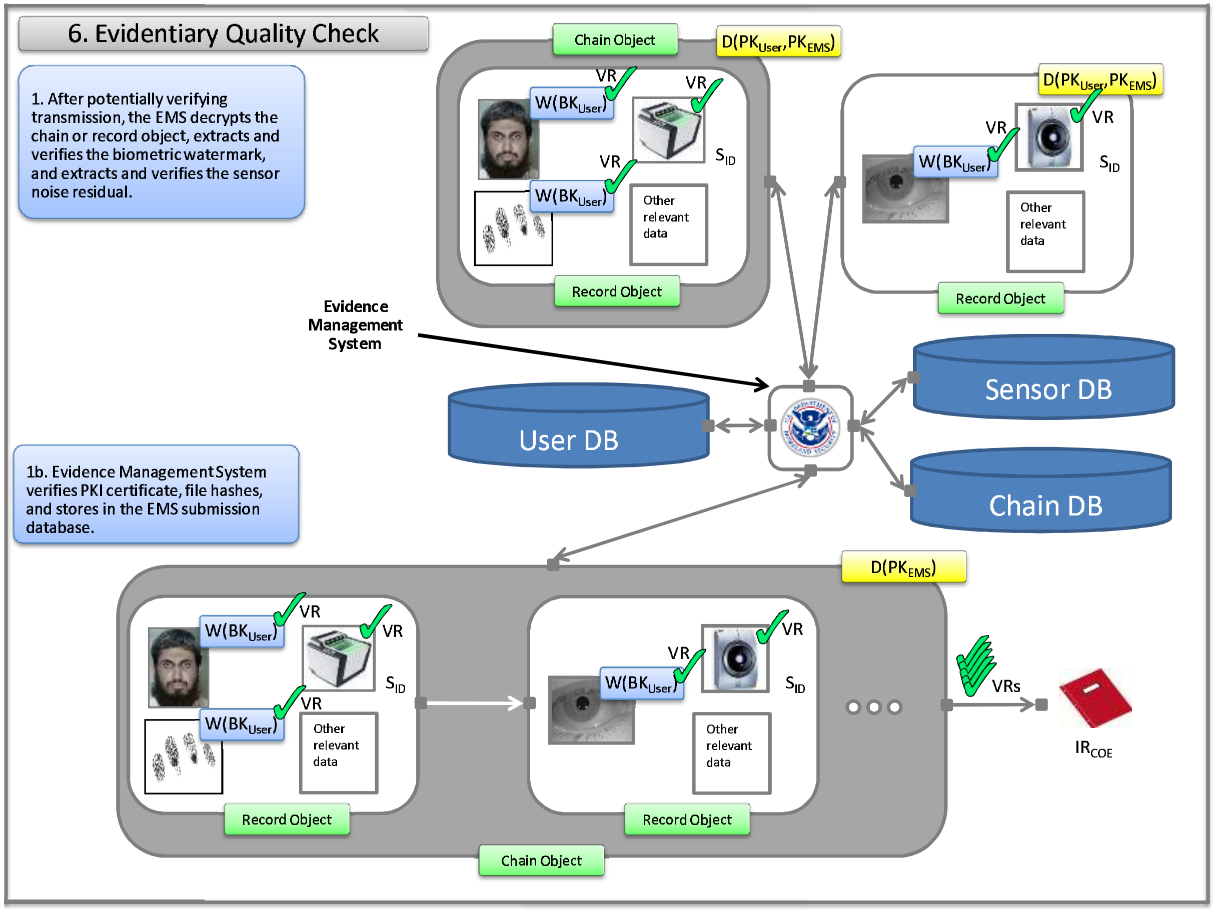
1. After creating or analyzing evidence, user's submission is sent to the appropriate Evidence Management System (EMS).

2. Evidence Management System verifies PKI certificate, file hashes, and stores in the EMS submission database.

Figure 6.7: Evidence Transmission Use Case.

6.3.7 Evidentiary Quality Check

The process of an evidentiary quality check is arguably the most important use case in a digital chain of evidence. The evidentiary quality check allows an individual to detect if the integrity of a chain of evidence has been compromised. The nature of this process is somewhat diverse in that it can be used to verify the integrity of newly acquired evidence or the entire evidence Chain Object itself. With that in mind, there are two different operations. These operations are pictured in Figure 6.8. The first would occur when an EMS receives new evidence from a user. This evidence can either take the form of a new Chain Object or a new Record Object to be added to a Chain Object. Before storing the information, the EMS must ensure the integrity of the new evidence thereby preventing alterations, fabrications, etc. To do so, the object would first be decrypted by the EMS using the PKI key pairs of the EMS and the user who submitted the evidence. This is represented by the notation D(PK_{USER} ,PK_{EMS}). Afterwards, the watermarked biometric data would be decoded. This process would produce the extracted watermark, BK_{USER} , and the reconstructed biometric data. The EMS would then verify the match of the BK_{USER} to the gallery template of the user within the biometric authentication system. Should the watermark and the gallery template match, this would indicate that the user appropriately signed the evidence. While this is also determined through successful decryption, biometric watermarking adds another layer of certainty beyond cryptographic means. This is because the biometric watermark remains after decryption providing evidence of integrity in unencrypted data. Finally, the source of the biometric evidence could also be verified by extracting noise residuals from the biometric data and comparing the residual to the S_{RP} in the EMS Sensor database. This provides a means another means for verifying the source of the data. Additionally, it is conceivable that the other relevant data may be vetted against records stored about the user by the EMS. However, this information would be EMS specific and is outside the scope of discussion in this conceptual framework. Should these processes return positive results, found in VR, the evidentiary quality check would be considered a success, records of the check could be stored in the Chain DB, and the newly submitted evidence would be stored by the EMS as submitted by the user.



6. Evidentiary Quality Check

1. After potentially verifying transmission, the EMS decrypts the chain or record object, extracts and verifies the biometric watermark, and extracts and verifies the sensor noise residual.

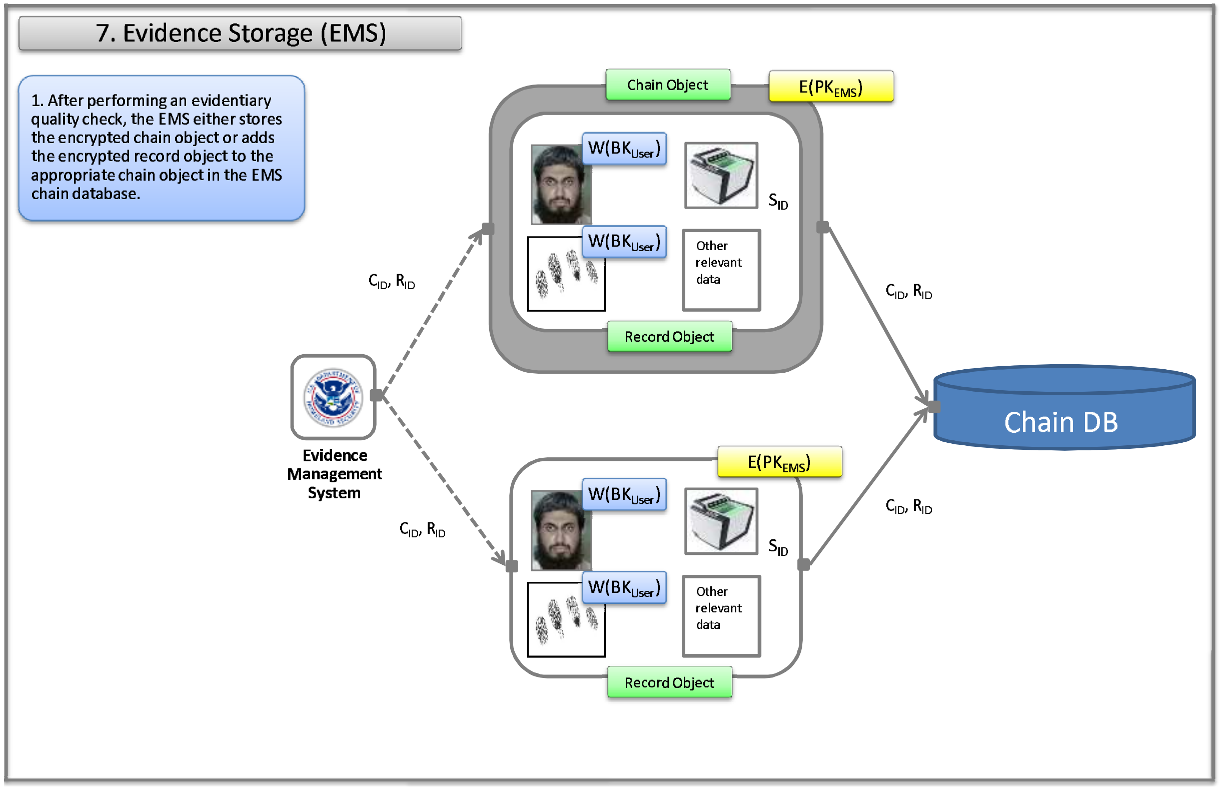
1b. Evidence Management System verifies PKI certificate, file hashes, and stores in the EMS submission database.

Figure 6.8: Evidentiary Quality Check Use Case.

As mentioned, an evidentiary quality check could verify the integrity of a new record within a chain, or establish the integrity of the entire chain object prior to presentation in a legal system. The latter case is simply an iterative version of the first case. Here, the same process as described in the previous paragraph would be applied iteratively from the first Record Object in the Chain Object through the last Record Object in the linked list of Record Objects making up the Chain Object. This would allow for the generation of a report, IR_{COE} , verifying the integrity of the chain from start to finish. IR_{COE} would encompass all previously mentioned validation processes including: transmission validation, content validation, and source validation. Such a report could be provided to a prosecutorial team, a judge, or other pertinent individuals.

6.3.8 Evidence Storage

While evidence could potentially be stored separately by any user within the chain or system outside the EMS, we are only concerned about storage of evidence that can be authenticated within the EMS system. With that in mind, Figure 6.9 shows the procedure for storing evidence in the Chain DB of an EMS. Once an EMS receives new evidence from a user (either a new Chain Object or a new Record Object to be added to a pre-existing chain) the system would perform an evidentiary quality check as outlined in the previous use case. Should the new evidence fail to meet the evidentiary quality check, EMS specific business rules could be applied to determine how best to deal with the situation. Pending a successful result, the storage process would consist of one of two options. If the evidence is a new Chain Object, the Object would be stored as such. If the new evidence were a Record Object to be added to an previously existing chain, the EMS would add the new Record Object to the end of the linked list representing the Chain Object. This process would be facilitated by the fact that the User provides the EMS with C_{ID} and R_{ID} unique identifiers. It is important to note that the evidence would be stored in encrypted format using the EMS PKI key pair, $\mathrm{E}(\mathrm{PK}_{EMS}\ )$.



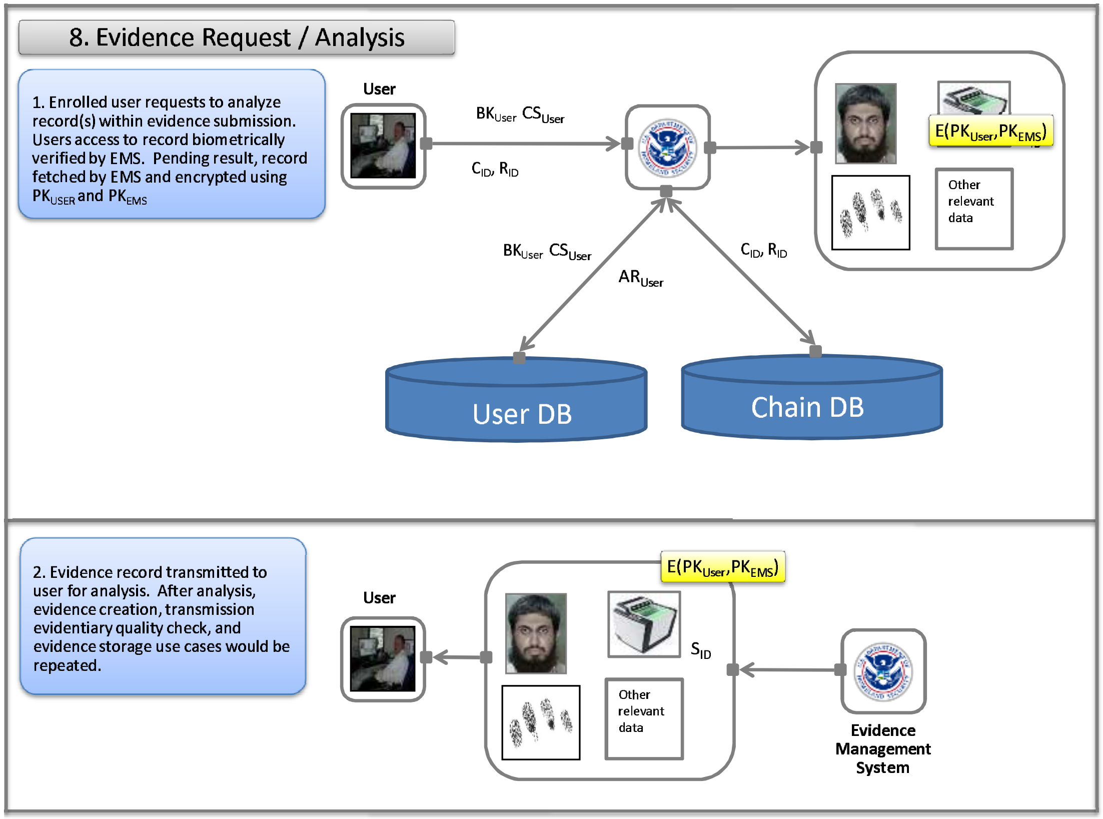
7. Evidence Storage (EMS)

1. After performing an evidentiary quality check, the EMS either stores the encrypted chain object or adds the encrypted record object to the appropriate chain object in the EMS chain database.

Figure 6.9: Evidence Storage Use Case.

6.3.9 Evidence Request / Analysis

Outside of creating new evidence, one important aspect of the EMS includes allowing entities within the chain to view, analyze, and potentially create new versions of previously existing evidence (i.e. quality enhancements). This process would be initiated by a user submitting a request to an EMS. The request would identify what specific record and chain the user wished to analyze using a C_{ID} and R_{ID} . Once the EMS receives the request, it would initiate a biometric authentication process. This biometric authentication process would provide a level of data access control which could be implemented at arbitrary levels of granularity with any type of data level access control mechanism (access control lists, matrices, etc.). Pending successful authentication (contained in AR_{USER}), the requested evidence would be encrypted for the user using the key pairs of the user and the EMS, $E(PK_{USER}, PK_{EMS})$. At that point the evidence would be transmitted to the user where he could perform the necessary operations. After completing those operations (which may include creation of new evidence), the evidence transmission, evidentiary quality check, and evidence storage use cases would be repeated to record the transactions in the evidence chain object. This use case is shown visually in Figure 6.10.



8. Evidence Request / Analysis

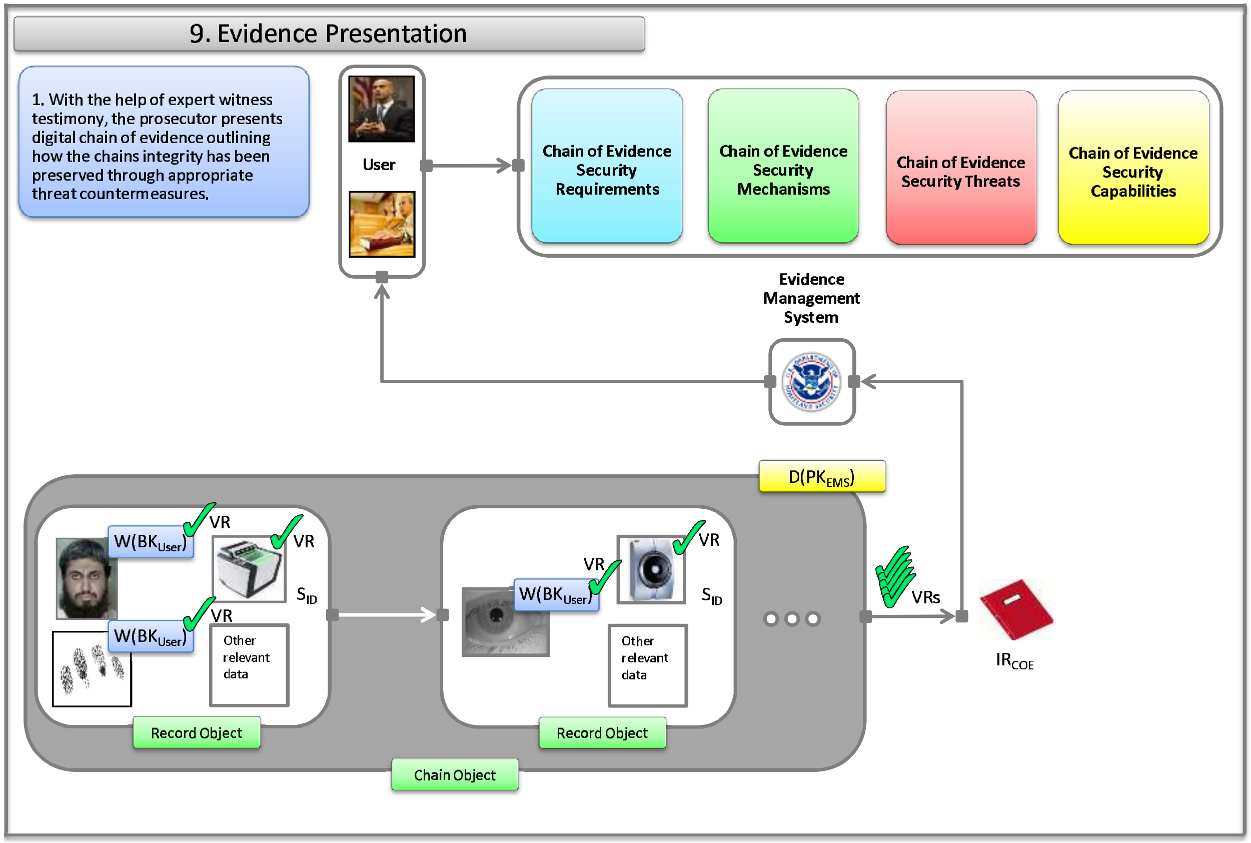
1. Enrolled user requests to analyze record(s) within evidence submission. Users access to record biometrically verified by EMS. Pending result, record fetched by EMS and encrypted using PK_{USER} and PK_{EMS}

2. Evidence record transmitted to user for analysis. After analysis, evidence creation, transmission evidentiary quality check, and evidence storage use cases would be repeated.

Figure 6.10: Evidence Request / Analysis Use Case.

6.3.10 Evidence Presentation

The final use case is the simplest operation within the conceptual framework. It involves presentation of the evidence stored in a chain object as well as presentation of the integrity of the chain object itself. As outlined in the evidentiary quality check use case, the last operation involves the dissemination of the evidence in an appropriate court system or as a final check before delivering news to next of kin in a mass fatality disaster incident. Here, the onus is to establish that the integrity of the chain of evidence has been maintained thereby preventing alterations, fabrications, false repudiations, etc. Referring back to the evidentiary quality check use case, the conceptual EMS is capable of generating a report, IR_{COE} , which verifies the aspects of integrity mentioned above. This report could be reviewed by the last individual prior to disseminating news to disaster victim next of kin or presented in the relevant court system by prosecutors with the help of an expert witness capable of authoritatively explaining the processes of the EMS. At that point, the integrity of the chain would not likely be in question. This process is pictured in 6.11



9. Evidence Presentation

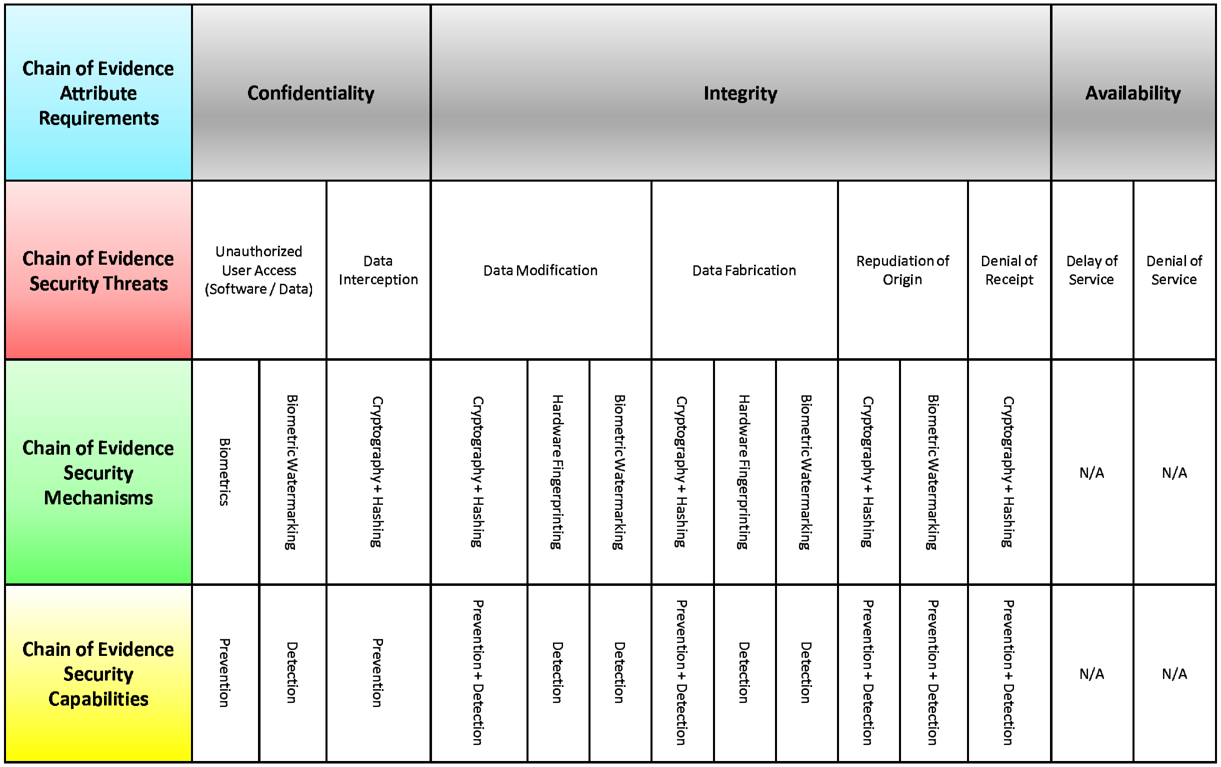
1. With the help of expert witness testimony, the prosecutor presents digital chain of evidence outlining how the chains integrity has been preserved through appropriate threat countermeasures.

User
Chain of Evidence Security Requirements
Chain of Evidence Security Mechanisms
Chain of Evidence Security Threats
Chain of Evidence Security Capabilities
Evidence Management System
D(PK$_{EMS}$)
W(BK$_{User}$)
VR
VR
S_{ID}
W(BK$_{User}$)
VR
Other relevant data
Record Object
VR
W(BK$_{User}$)
VR
S_{ID}
Other relevant data
Record Object
Chain Object
VRs
IR$_{COE}$

Figure 6.11: Evidence Presentation Use Case.

6.4 Chain of Evidence Trustworthiness: Attributes, Threats, Mechanisms, and Capabilities

As pictured in in Figure 6.2, the conceptual framework deals with four main aspects of computer security: attribute requirements, threats, mechanisms, and capabilities. While aspects of these issues have already been discussed in previous sections, the purpose of this section is to provide a structured discussion of how the proposed mechanisms deal with specific threats, thereby providing capabilities which meet the requirements of the digital chain of evidence. Figure 6.12 concisely summarizes this relationship. Before delving into specific threats, mechanisms for countering threats, and associated capabilities, it is important to reiterate the concept note the depth of this analysis. Given the conceptual nature of the proposed framework, it does not include matters such as specific implementation details, rather it provides notional concepts. With this in mind, it is not appropriate to vet the conceptual framework against vulnerabilities at the level of a US-CERT maintained database. Rather, the framework should be vetted against security threats at an appropriate level of depth. With that in mind, we consider the proposed framework's ability to address conceptual notions of six security threats related to the service requirements of confidentiality and integrity. In the first row of the figure, chain of evidence security attribute requirements are presented in terms of generalized computer security requirements: confidentiality, integrity, and availability. As mentioned in Section 6.2, for the purposes of this work, we are only concerned with confidentiality and availability. While availability should not be ignored, we are not aware of any special circumstances that are specific to the chain of evidence, as opposed to a generic trusted computing application.



Chain of Evidence Attribute Requirements	Confidentiality			Integrity									Availability	
Chain of Evidence Security Threats	Unauthorized User Access (Software / Data)		Data Interception	Data Modification			Data Fabrication			Repudiation of Origin		Denial of Receipt	Delay of Service	Denial of Service
Chain of Evidence Security Mechanisms	Biometrics	Biometric Watermarking	Cryptography + Hashing	Cryptography + Hashing	Hardware Fingerprinting	Biometric Watermarking	Cryptography + Hashing	Hardware Fingerprinting	Biometric Watermarking	Cryptography + Hashing	Biometric Watermarking	Cryptography + Hashing	N/A	N/A
Chain of Evidence Security Capabilities	Prevention	Detection	Prevention	Prevention + Detection	Detection	Detection	Prevention + Detection	Detection	Detection	Prevention + Detection	Prevention + Detection	Prevention + Detection	N/A	N/A

Figure 6.12: Security Service Requirements, Threats, Mechanisms, and Capabilities.

Therefore, this aspect of the chain could be ensured through normal information technology mechanisms for achieving availability such as employing appropriate firewall, redundancy, and backup systems.

6.4.1 Confidentiality

We first consider the notion of confidentiality. As previously mentioned, confidentiality refers to the concealment of information or services. Specific security threats related to confidentiality include unauthorized access to software systems or data and interception of data as pictured in second and third columns of the second row of Figure 6.12.

Unauthorized User Access (Software / Data)

Through the applications of credential management and particularly biometric systems, we provide the ability to reasonably prevent unauthorized user access to software and data. As mentioned in the User Login and the Evidence Request use cases, access to the software system is protected by biometric authentication as is access to individual components of chain of evidence data. While no biometric system is perfect, the combination of credential management and biometric authentication is arguably the best mix of security and convenience available at a reasonable price today in network systems. It would also be possible to detect the unauthorized user access to data through the security mechanism of biometric watermarking. As biometric watermarking is used as a means of ”signing” for evidence, the lack of, or presence of an incorrect biometric watermark would indicate that an unauthorized user has provided that evidence.

Data Interception

The other security threat related to confidentiality considered in this analysis is the threat of data interception. While it is nearly impossible to prevent all means of intercepting data in a network environment, cryptographic approaches can be used to make the effects of risk of data interception minimal. By that, it would be theoretically possible for an outsider to intercept data being transmitted to or from an EMS. Yet without appropriate access to cryptographic key pairs, the intercepted data would be of no use to the outsider. Similar to the notion of availability, it would be possible to implement virtual private networks and other mechanisms that would minimize the likelihood of this threat being realized. Once again, PKI protocols are described in Appendix B.

6.4.2 Integrity

The most important security service related to systems maintaining digital chains of evidence is integrity. By that, we wish to ensure that data has not be inappropriately altered or modified, fabricated, etc. There are four major security threats related to integrity: data modification, data fabrication, repudiation of origin, and denial of receipt.

Data Modification

In a data modification threat, previously existing data is altered either by someone within the chain of evidence or someone outside the chain. Note, this modification can be intentional or unintentional. The first mechanism for ensuring that data is not altered involves the application of cryptography and hashing functions. Using established techniques, one can easily determine that data has been modified. One indication that data has been modified would be that decryption processes fail. Once again, whether the modification was intentional or unintentional is not relevant. If a package is correctly encrypted and transmitted it, the homomorphic nature of the decryption process result in a readable result and failure could indicate a modification. Additionally, while cryptography couldn't prevent the modification of data, it can be used to prevent the meaningful modification of data by an individual with malicious intent. While this is one mechanism, the classic mechanism for detecting changes in data involves file hashing. Here a file (or potentially a chain or record object) is passed through a hashing algorithm (i.e. SHA-2) which generates a unique, fixed length string based on the contents of the file. In theory, modifying so much as one bit in the file should change the output of the hash function. Therefore it would be possible to detect any modification by comparing the hash strings.

The next way to detect data modification is through biometric watermarking. In a biometric system, the inability to correctly extract the expected biometric watermark either indicates the host data was either not watermarked in the first place, or the the data has been modified outside of the appropriate usage of the system. Once again, this may entail an intentional act on the part of an individual with malicious intent or it could be an innocent but important mistake on the part of an authorized user. Either way, such a modification is important in the scope of digital chains of evidence and biometric watermarking would be capable of detection.

The final mechanism presented in the framework for detecting modification of data is hardware fingerprinting. Keeping in mind the importance of establishing the source of data, digital hardware fingerprinting is capable of detecting modifications of an image because many modifications will alter the noise residual associated with an image. Although elements of this application were not specifically presented in this study, works such those seen in [26] have established this possibility. It should be noted that this would not necessarily apply to all possible modifications a biometric image may be subject to. In other words, it may be possible to modify an image in such a way that the associated noise residual is not significantly affected such that is results in a failure to match the sensors reference template.

Data Fabrication

The threat of data fabrication is essentially the same as data modification. The main difference is that fabrication entails creation of evidence instead of modifying previously existing evidence. With that in mind, the same security mechanisms and principles applied in the previous subsection can be used to prevent and detect data fabrication attempts. However, it is important to note the special consideration of this threat as it relates to the digital chain of evidence. Whether from an outsider or individual authorized to use contribute to an evidence chain, the notion of planting evidence is highly important. This may be viewed as the principle application of digital hardware fingerprinting. It is conceivable to think someone may fraudulently plant previously acquired electronic fingerprint to frame another individual. That said, if the noise residual of the fingerprints cannot be tracked back to the capture device used at the point of acquisition, it would be indicative of an attempt at fabrication or forgery.

Repudiation of Origin

The next security threat related to the security service of integrity is repudiation of origin. In such a case, an individual denies that he sent or in the case of a digital chain of evidence, created data. Related to the the previous threats of data modification and fabrication, it is extremely important to log who handles and creates evidence. It is conceivable to imagine a case relating to a digital chain of evidence where an individual would falsely deny having handled or created evidence. However, through the application of cryptography and hashing as well as biometric watermarking it would be feasible to detect false repudiations. Both PKI cryptography and biometric watermarking are proposed as means to "digitally sign" evidence within the framework. The former obviously involves cryptographic processes, the latter takes place outside of encryption. In order for evidence to be watermarked and "digitally signed" by the EMS, the creating user must submit his or her biometric sample. This means it would be extremely difficult to make a case for repudiation of origin. Effectively, someone would have had to acquire and implant the biometric evidence of the user in question which is highly unlikely. Additionally, it is widely known that one can combat repudiation of origin through appropriate use of PKI key pairs. Therefore, if the watermarking and cryptography processes check out, in all likelihood, repudiation of origin is not a feasible defense for the user in question.

Denial of Receipt

The last security threat considered in this work is denial of receipt. Similar to repudiation of origin, here an individual or system falsely proclaims that he did not receive data, or in this case, evidence. Classic applications of PKI cryptography as seen in [17, 114] can be used to prevent and detect this threat.

6.4.3 Availability

To reiterate, the security threats related to the security service of availability are not considered relevant for the purpose of this work. Therefore, analysis does not apply.

6.5 Agnostic Nature of the Framework Components

One of the reasons for proposing a conceptual framework is that it allows for a high degree of flexibility and agnosticism with respect to the choice of specific security mechanisms applied. Although we propose a specific instantiation of the framework in the following section, the conceptual framework does not rely on any specific biometric authentication, cryptographic, biometric watermarking, or hardware fingerprinting system implementation. For example, face recognition could be used just as easily as iris recognition could be for dealing with access control issues. An amplitude modulation approach to biometric watermarking could be adapted just as easily as an approach operating outside the spatial domain. If one would prefer to apply a hardware fingerprinting approach which identifies devices based on fixed pattern noise (FPN) or any other method picture in Figure 6.13 instead of photo-response non-uniformity noise (PRNU), this is perfectly acceptable. This flexibility is desirable for a number of reasons. First of all, there are inevitably preferences, if not fervent disagreements regarding the choice of technology among the various organizations who could employ evidence management systems. If a the framework were not sufficiently flexible, such that it could adapt to such preferences, the benefit the research would be significantly less broad.

Secondly, the ideas behind the proposed framework can be sustained as technology and research evolves. If a new biometric emerges that outperforms all other currently existing biometrics, there is no immediate reason why this biometric could not be substituted for what is currently in place. The same argument can be made for all other security mechanisms provided. Finally, the purpose of this work is not to provide a deployment ready system. Rather this work aims to shrink the gap between currently existing methods for maintaining paper based chains and the development of digital chains of evidence capable of greater levels of authentication.

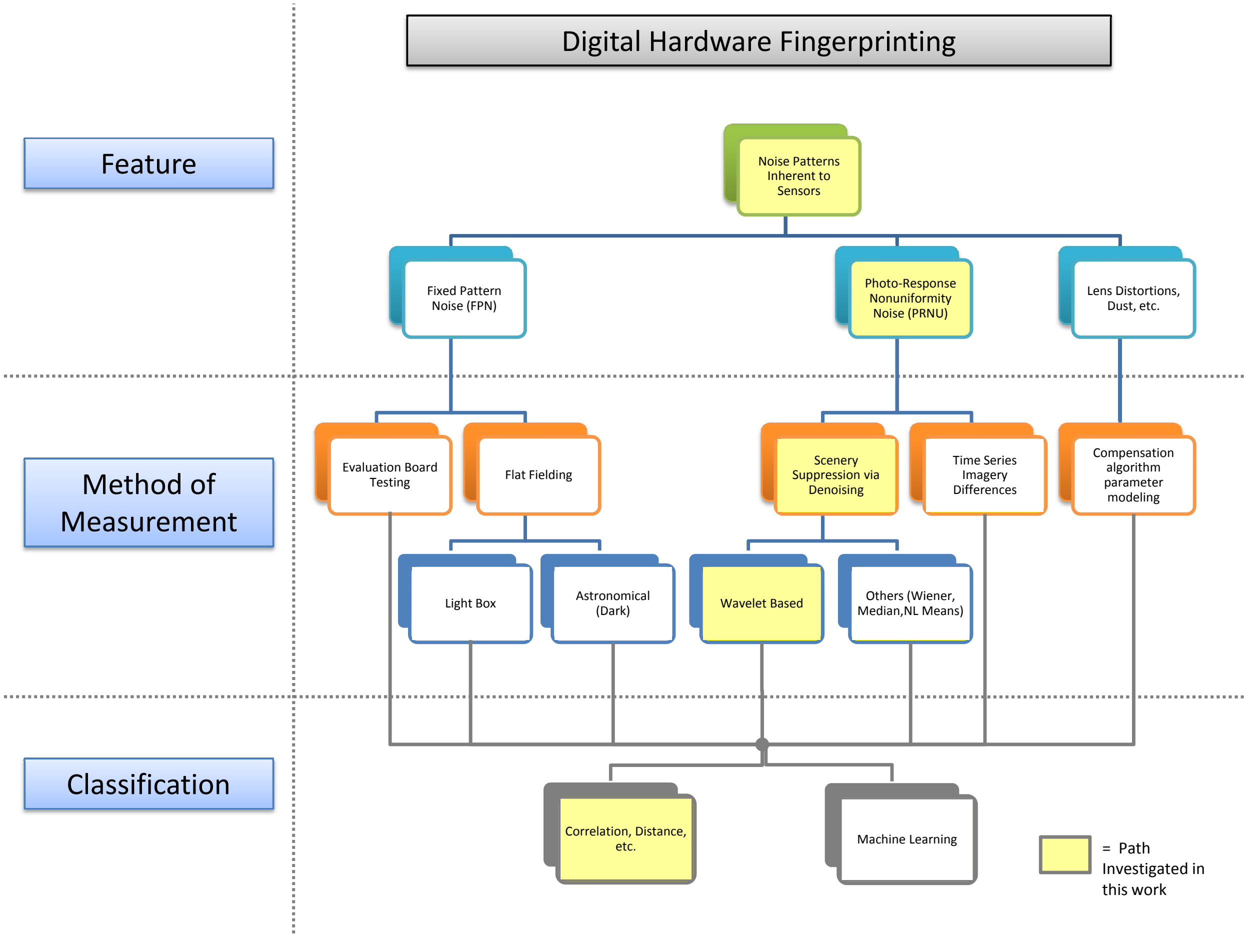

Digital Hardware Fingerprinting
Feature
Method of Measurement
Classification
Noise Patterns Inherent to Sensors
Fixed Pattern Noise (FPN)
Photo-Response Nonuniformity Noise (PRNU)
Lens Distortions, Dust, etc.
Evaluation Board Testing
Flat Fielding
Scenery Suppression via Denoising
Time Series Imagery Differences
Compensation algorithm parameter modeling
Light Box
Astronomical (Dark)
Wavelet Based
Others (Wiener, Median,NL Means)
Correlation, Distance, etc.
Machine Learning
= Path Investigated in this work

Figure 6.13: Approaches to Digital Hardware Fingerprinting.

6.6 Instantiation of the Framework through Specific Security Mechanisms

Although there is a number of reasons to stop at the conceptual framework proposed thus far, it is beneficial propose one more layer of depth by providing an example instantiation of the framework relying on specific topics studied in previous chapters of this work. The remainder of this section breaks down the instantiation by the four security mechanisms involved.

6.6.1 Biometric Systems

The biometrics are responsible for two operations within the proposed framework. First, they are used to control access to the software system and specific objects of evidence such as chains and records within chains. Second, they are a component of the biometric watermarking system the watermarking system which is used to "sign" evidence. Taking this into consideration, there may be one biometric system which controls access and another which is used for watermarking. Given that most software systems already maintain username and password credential sets, we believe it would be appropriate to apply a keystroke dynamics system similar to that described in Chapter 5 to control access to the software system itself. Beyond that one might wish to apply an online signature based system for allowing access to specific evidence and generating biometric watermarks. This would be a direct similarity to currently existing paper based chains where evidence is physically signed for. However, the process of penning a signature may be costly in terms of acquisition time in certain situations. For instance, it may be appropriate for an inspection officer in a POE secondary screening room to provide a biometric watermark generated from facial feature vectors instead of having to physically sign every submission. Therefore the choice of a biometric for this purpose is truly system specific.

6.6.2 Biometric Watermarking

For the purposes of biometric watermarking we propose to use the amplitude modulation approaches including the one described in Chapter 3. With the exception of substituting the biometric watermark to reflect the decision in in the previous section, this algorithm would be sufficient for watermarking evidence taking the form of iris images such as those that might be found in a system such NEXUS maintained by DHS. Furthermore, work such as that presented in [61, 63] have established mechanisms using similar amplitude modulation schemes to watermark face and fingerprint images. While it is conceivable that other biometric data would need to be watermarked, there is no immediate reason why such a scheme couldn't be modified to handle such cases.

6.6.3 Digital Hardware Fingerprinting

In terms of the digital hardware fingerprinting services, we would suggest using the approach described in Chapter 4. Having established the prospects of the approach to perform source validation on biometric fingerprint readers and iris cameras this technique would is now potentially capable of dealing with sensors which acquire data from three most commonly collected biometrics. This is the case other works have established the ability of the technique to identify traditional digital cameras which are commonly used to capture face images [86].

6.6.4 Cryptography

The choice of cryptographic is not particularly relevant to this section and it would likely be subject to the organization maintaining the EMS. The only requirement is that it be based on public key infrastructure with asymmetric public key pairs.

Chapter 7

Conclusion

7.1 Conclusion

There are many scenarios dealing with biometric identification which require the ability to establish and maintain a digital chain of evidence. Whether dealing with law enforcement cases such as those seen in crime scene investigations, people screening activities at POEs, and expanded maritime interception operations or homeland security responses to mass fatality incident disasters, the ability to establish, maintain, and prove the integrity of a digital chain of evidence is necessary to avoid undesirable consequences. These consequences can entail the inability to effectively prosecute known and suspected terrorists and criminals (resulting in their subsequent release), wrongful convictions of innocent people, and errors in notifying the next of kin of disaster victims. While the proliferation of digital evidence has brought to bear new challenges in developing these verifiable chains of evidence, advances in security technologies have afforded greater development opportunities that did not exist in years past. Through the application of security mechanisms such as cryptography and hashing, biometric authentication, biometric watermarking, and digital hardware fingerprinting, this work has proposed a conceptual framework which is capable of providing security capabilities that can detect and prevent threats against the confidentiality and integrity of these digital chains of evidence. Furthermore, specific contributions to the topics of keystroke dynamics, amplitude modulation based biometric watermarking, and hardware fingerprinting of biometric sensors allow for the instantiation of the developed conceptual framework which is even closer to the point of direct implementation.

To reiterate, this work provides four main contributions:

1. We developed an iris digital watermarking system which is not only resistant to common application scenarios such at database compression and partial progressive decoding, but also capable of withstanding the rewatermarking process which might be seen in the proposed chain of evidence. Our work extended current biometric watermarking techniques by modifying existing approaches to allow for selective encoding in the region of interest in iris biometric images. Additionally, our approach provides a novel asymmetric implementation of the watermarking scheme.

2. We demonstrated the ability to perform source validation on biometric modalities which collect data with capture devices outside of typical photographic cameras. We applied an approach designed for digital cameras which relies on PRNU noise to a series of biometric fingerprint readers as well as iris cameras using sensors which respond to the infrared band of the electromagnetic spectrum.

3. We demonstrated that input stimulus familiarity is a driver of classification perfor-

mance in keystroke dynamic systems. This observation is should prove to be useful in selection process of username and password credential sets.

4. We developed a conceptual framework for establishing and maintaining digital chain of evidence dealing with biometric data which relies on elements of cryptography, biometric watermarking, digital hardware fingerprinting, and biometric authentication. We demonstrated how the framework is capable of dealing with security threats associated with confidentiality and integrity as well as including an example instantiation of the conceptual framework which makes use of the other contributions of the work.

7.2 Future Work

While this work shrinks the gap between traditional paper-based chains of evidence and implementations of digital chains of evidence with enhanced mechanisms for proving integrity, there are many more topics that can be investigated to further shrink this gap. In terms of the proposed framework, there are benefits to its conceptual nature. However, given the level considered, there are likely to be complicated issues which may arise when specific implementation details are developed. As the concepts move closer to deployment implementation details at a greater level of depth will need to be developed. This extended formalization will also be required to test the framework and associated protocol using methods from software engineering and information assurance communities. This could also likely involve developing a pilot system to be run in one of the many potential application environments. Policy and legal issues must be explored to determine what, if any hurdles, stand in the way of applying methods such as biometric watermarking and hardware fingerprinting as part of maintaining such chains. These mechanisms do alter the data from its original form, however so to do state of the art image enhancement techniques. While these techniques appear to be accepted in the court of law, it is necessary to set judicial precedence related to the proposed frameworks for digital chains of evidence. We have also noted that this work does not specifically look into all the different types of meta-data which may collected about digital evidence. This type of evidence was labeled as "Other relevant data," in Chapter 6. More work could be conducted to research what specific meta-data should be collected, stored, and presented however we belief this is best suited for individuals with legal expertise.

In the field of digital hardware fingerprinting, similar tests must be run on much larger datasets which provide a closer reflection of the number of capture devices that may be fielded by organization maintaining chains of evidence. Furthermore, tests must be run on other sensors including new ten-print fingerprint scanners and mobile collection devices. Such devices may require the need to investigate other options for source identification such as those pictured in Figure 6.13. Additionally, it would also be interesting to look into other levels of hardware fingerprinting potentially including model, brand, and technology. Finally, a deeper investigation of both the limitations and potential areas of extension of the chosen approach would be appropriate.

Related to biometric watermarking, there may be a need to integrate the aspects of biometric cryptosystems into the watermarking process. Therefore, transformed versions of biometric feature vectors could be used to watermark biometric host data. Although we do not believe them to be serious, this may alleviate potential security and privacy related concerns associated with the proposed implementation. Extending the analysis of the proposed rewatermarking scheme in terms of the systems degree of fragility would also be useful.

While work continues in the field of keystroke dynamics, little work has been done to establish minimum requirements for training classifiers that differentiate between genuine and imposter input classes. Furthermore, a topic that has yet to be addressed is methods of automatically generating input sequences which closely model imposter data.

This page intentionally contains only this sentence.

Appendix A

Hardware Fingerprinting Confusion Matrices

A.1 Fingerprint Confusion Matrices

A.1.1 FVC Data

Classified Actual	KeyTronic	Microelectronic	Identicator	Identix	Biometrika	Precise	CrossMatch	DigitalPersona
KeyTronic	510	64	40	23	25	52	58	18
Microelectonic	3	774	2	4	1	0	1	5
Identicator	39	34	616	25	17	26	15	18
Identix	89	83	66	308	56	59	53	76
Biometrika	0	0	0	0	790	0	0	0
Precise	0	0	0	0	0	790	0	0
CrossMatch	1	1	0	0	1	1	782	4
Digital Persona	0	0	0	0	0	0	0	790

Table A.1: FVC confusion matrix when training on 1 image per sensor

Classified	Key-	Micro-	Identi-	Identix	Biometrika	Precise	Cross-	Digital-
Actual	Tronic	electronic	cator				Match	Persona
KeyTronic	713	4	16	11	12	6	16	2
Microelectonic	0	778	0	1	0	0	0	1
Identicator	16	15	714	9	3	12	5	6
Identix	77	62	53	348	51	64	50	75
Biometrika	0	0	0	0	780	0	0	0
Precise	0	0	0	0	0	780	0	0
CrossMatch	0	1	1	0	2	0	775	1
Digital Persona	0	0	0	0	0	0	0	780

Table A.2: FVC confusion matrix when training on 2 images per sensor

Classified	Key-	Micro-	Identi-	Identix	Biometrika	Precise	Cross-	Digital-
Actual	Tronic	electronic	cator				Match	Persona
KeyTronic	719	5	4	6	5	7	9	5
Microelectonic	0	760	0	0	0	0	0	0
Identicator	11	0	740	4	0	3	2	0
Identix	57	59	57	361	43	48	48	87
Biometrika	0	0	0	0	760	0	0	0
Precise	0	0	0	0	0	760	0	0
CrossMatch	0	0	0	0	1	0	758	1
Digital Persona	0	0	0	0	0	0	0	760

Table A.3: FVC confusion matrix when training on 4 images per sensor

Classified	Key-	Micro-	Identi-	Identix	Biometrika	Precise	Cross-	Digital-
Actual	Tronic	electronic	cator				Match	Persona
KeyTronic	688	8	3	7	5	3	4	2
Microelectonic	0	720	0	0	0	0	0	0
Identicator	0	0	720	0	0	0	0	0
Identix	49	46	54	400	37	40	37	57
Biometrika	0	0	0	0	720	0	0	0
Precise	0	0	0	0	0	720	0	0
CrossMatch	0	0	0	0	0	0	720	0
Digital Persona	0	0	0	0	0	0	0	720

Table A.4: FVC confusion matrix when training on 8 images per sensor

Classified	Key-	Micro-	Identi-	Identix	Biometrika	Precise	Cross-	Digital-
Actual	Tronic	electronic	cator				Match	Persona
KeyTronic	640	0	0	0	0	0	0	0
Microelectonic	0	640	0	0	0	0	0	0
Identicator	0	0	640	0	0	0	0	0
Identix	26	44	43	421	19	26	18	43
Biometrika	0	0	0	0	640	0	0	0
Precise	0	0	0	0	0	640	0	0
CrossMatch	0	0	0	0	0	0	640	0
Digital Persona	0	0	0	0	0	0	0	640

Table A.5: FVC confusion matrix when training on 16 images per sensor

Classified	Key-	Micro-	Identi-	Identix	Biometrika	Precise	Cross-	Digital-
Actual	Tronic	electronic	cator				Match	Persona
KeyTronic	480	0	0	0	0	0	0	0
Microelectonic	0	480	0	0	0	0	0	0
Identicator	0	0	480	0	0	0	0	0
Identix	14	18	23	365	14	13	10	28
Biometrika	0	0	0	0	480	0	0	0
Precise	0	0	0	0	0	480	0	0
CrossMatch	0	0	0	0	0	0	480	0
Digital Persona	0	0	0	0	0	0	0	480

Table A.6: FVC confusion matrix when training on 32 images per sensor

Classified	Key-	Micro-	Identi-	Identix	Biometrika	Precise	Cross-	Digital-
Actual	Tronic	electronic	cator				Match	Persona
KeyTronic	160	0	0	0	0	0	0	0
Microelectonic	0	160	0	0	0	0	0	0
Identicator	0	0	160	0	0	0	0	0
Identix	2	3	3	136	4	4	3	5
Biometrika	0	0	0	0	160	0	0	0
Precise	0	0	0	0	0	160	0	0
CrossMatch	0	0	0	0	0	0	160	0
Digital Persona	0	0	0	0	0	0	0	160

Table A.7: FVC confusion matrix when training on 64 images per sensor

A.1.2 WVU Data

Classified Actual	BioTouch #1	BioTouch #2	BioTouch #2	Microsoft #1	Microsoft #2	Microsoft #3
BioTouch #1	3942	14	32	1	1	0
BioTouch #2	6	3967	17	0	0	0
BioTouch #3	3	10	3977	0	0	0
Microsoft #1	0	0	0	3989	1	0
Microsoft #2	0	0	0	0	3990	0
Microsoft #3	0	0	0	0	0	3990

Table A.8: WVU confusion matrix when training on 1 images per sensor

Classified Actual	BioTouch #1	BioTouch #2	BioTouch #2	Microsoft #1	Microsoft #2	Microsoft #3
BioTouch #1	3980	0	0	0	0	0
BioTouch #2	1	3979	0	0	0	0
BioTouch #3	0	0	3980	0	0	0
Microsoft #1	0	0	0	3980	0	0
Microsoft #2	0	0	0	0	3980	0
Microsoft #3	0	0	0	0	0	3980

Table A.9: WVU confusion matrix when training on 2 images per sensor

Classified Actual	BioTouch #1	BioTouch #2	BioTouch #2	Microsoft #1	Microsoft #2	Microsoft #3
BioTouch #1	3960	0	0	0	0	0
BioTouch #2	0	3960	0	0	0	0
BioTouch #3	0	0	3960	0	0	0
Microsoft #1	0	0	0	3960	0	0
Microsoft #2	0	0	0	0	3960	0
Microsoft #3	0	0	0	0	0	3960

Table A.10: WVU confusion matrix when training on 4 images per sensor

Classified Actual	BioTouch #1	BioTouch #2	BioTouch #2	Microsoft #1	Microsoft #2	Microsoft #3
BioTouch #1	3920	0	0	0	0	0
BioTouch #2	0	3920	0	0	0	0
BioTouch #3	0	0	3920	0	0	0
Microsoft #1	0	0	0	3920	0	0
Microsoft #2	0	0	0	0	3920	0
Microsoft #3	0	0	0	0	0	3920

Table A.11: WVU confusion matrix when training on 8 images per sensor

Classified Actual	BioTouch #1	BioTouch #2	BioTouch #2	Microsoft #1	Microsoft #2	Microsoft #3
BioTouch #1	3840	0	0	0	0	0
BioTouch #2	0	3840	0	0	0	0
BioTouch #3	0	0	3840	0	0	0
Microsoft #1	0	0	0	3840	0	0
Microsoft #2	0	0	0	0	3840	0
Microsoft #3	0	0	0	0	0	3840

Table A.12: WVU confusion matrix when training on 16 images per sensor

Classified Actual	BioTouch #1	BioTouch #2	BioTouch #2	Microsoft #1	Microsoft #2	Microsoft #3
BioTouch #1	3680	0	0	0	0	0
BioTouch #2	0	3680	0	0	0	0
BioTouch #3	0	0	3680	0	0	0
Microsoft #1	0	0	0	3680	0	0
Microsoft #2	0	0	0	0	3680	0
Microsoft #3	0	0	0	0	0	3680

Table A.13: WVU confusion matrix when training on 32 images per sensor

A.1.3 WVU / Clarkson Data

Classified Actual	WVU Precise	WVU Secugen	WVU CrossMatch	Clarkson Precise	Clarkson Secugen	Clarkson CrossMatch
WVU Precise	3707	198	213	522	192	168
WVU Secugen	948	1107	494	944	972	535
WVU CrossMatch	1131	763	834	1168	668	436
Clarkson Precise	1056	371	299	2658	315	301
Clarkson Secugen	662	567	259	594	2584	334
Clarkson CrossMatch	576	459	224	721	330	2690

Table A.14: WVU / Clarkson confusion matrix when training on 1 image per sensor

Classified Actual	WVU Precise	WVU Secugen	WVU CrossMatch	Clarkson Precise	Clarkson Secugen	Clarkson CrossMatch
WVU Precise	4216	118	101	386	104	75
WVU Secugen	917	1144	522	923	972	522
WVU CrossMatch	1069	729	937	1075	709	481
Clarkson Precise	904	281	255	3088	226	246
Clarkson Secugen	356	475	203	521	3253	192
Clarkson CrossMatch	499	268	134	338	312	3449

Table A.15: WVU / Clarkson confusion matrix when training on 2 image per sensor

Classified Actual	WVU Precise	WVU Secugen	WVU CrossMatch	Clarkson Precise	Clarkson Secugen	Clarkson CrossMatch
WVU Precise	4559	57	50	261	42	31
WVU Secugen	921	1253	463	911	981	471
WVU CrossMatch	1058	648	1053	1094	730	417
Clarkson Precise	511	152	137	3907	180	113
Clarkson Secugen	268	300	144	293	3820	175
Clarkson CrossMatch	263	112	69	248	199	4109

Table A.16: WVU / Clarkson confusion matrix when training on 4 image per sensor

Classified	WVU	WVU	WVU	Clarkson	Clarkson	Clarkson
Actual	Precise	Secugen	CrossMatch	Precise	Secugen	CrossMatch
WVU Precise	4823	20	20	97	22	18
WVU Secugen	793	1519	440	874	935	439
WVU CrossMatch	967	617	1246	1096	675	399
Clarkson Precise	516	114	81	4088	112	89
Clarkson Secugen	127	219	62	108	4386	98
Clarkson CrossMatch	144	84	21	140	139	4472

Table A.17: WVU / Clarkson confusion matrix when training on 8 image per sensor

Classified	WVU	WVU	WVU	Clarkson	Clarkson	Clarkson
Actual	Precise	Secugen	CrossMatch	Precise	Secugen	CrossMatch
WVU Precise	4907	7	6	65	12	3
WVU Secugen	737	1756	365	808	943	391
WVU CrossMatch	908	575	1732	912	581	292
Clarkson Precise	221	35	30	4647	37	30
Clarkson Secugen	73	93	21	75	4695	43
Clarkson CrossMatch	101	46	18	80	84	4671

Table A.18: WVU / Clarkson confusion matrix when training on 16 image per sensor

Classified	WVU	WVU	WVU	Clarkson	Clarkson	Clarkson
Actual	Precise	Secugen	CrossMatch	Precise	Secugen	CrossMatch
WVU Precise	4961	4	2	29	3	1
WVU Secugen	639	2150	339	686	881	305
WVU CrossMatch	708	459	2239	804	497	293
Clarkson Precise	138	35	11	4781	16	19
Clarkson Secugen	33	46	21	44	4840	16
Clarkson CrossMatch	57	30	13	29	30	4841

Table A.19: WVU / Clarkson confusion matrix when training on 32 image per sensor

Classified	WVU	WVU	WVU	Clarkson	Clarkson	Clarkson
Actual	Precise	Secugen	CrossMatch	Precise	Secugen	CrossMatch
WVU Precise	4981	1	2	14	2	0
WVU Secugen	544	2547	246	605	780	278
WVU CrossMatch	491	333	3058	592	335	191
Clarkson Precise	73	2	2	4913	5	5
Clarkson Secugen	6	29	17	32	4908	8
Clarkson CrossMatch	19	12	1	18	19	4931

Table A.20: WVU / Clarkson confusion matrix when training on 64 image per sensor

Classified	WVU	WVU	WVU	Clarkson	Clarkson	Clarkson
Actual	Precise	Secugen	CrossMatch	Precise	Secugen	CrossMatch
WVU Precise	4979	0	0	21	0	0
WVU Secugen	364	3146	179	492	596	223
WVU CrossMatch	313	188	3781	366	246	106
Clarkson Precise	32	3	0	4963	2	0
Clarkson Secugen	14	25	7	14	4940	0
Clarkson CrossMatch	27	1	2	1	10	4959

Table A.21: WVU / Clarkson confusion matrix when training on 128 images per sensor

Classified	WVU	WVU	WVU	Clarkson	Clarkson	Clarkson
Actual	Precise	Secugen	CrossMatch	Precise	Secugen	CrossMatch
WVU Precise	4990	1	0	9	0	0
WVU Secugen	233	3943	93	261	360	110
WVU CrossMatch	93	58	4579	144	101	25
Clarkson Precise	12	0	0	4988	0	0
Clarkson Secugen	2	15	4	10	4969	0
Clarkson CrossMatch	5	1	0	0	7	4987

Table A.22: WVU / Clarkson confusion matrix when training on 256 images per sensor

A.2 Iris Confusion Matrices

Classified Actual	ICE LG	CASIA OKI	WVU OKI	WVU EverFocus
ICE LG	1895	26	33	36
CASIA OKI	0	1990	0	0
WVU OKI	36	38	1901	15
WVU EverFocus	0	0	0	1990

Table A.23: Iris confusion matrix when training on 1 images per camera

Classified Actual	ICE LG	CASIA OKI	WVU OKI	WVU EverFocus
ICE LG	1954	26	33	36
CASIA OKI	0	1980	0	0
WVU OKI	21	32	1919	8
WVU EverFocus	0	0	0	1980

Table A.24: Iris confusion matrix when training on 2 images per camera

Classified Actual	ICE LG	CASIA OKI	WVU OKI	WVU EverFocus
ICE LG	1951	8	1	0
CASIA OKI	0	1960	0	0
WVU OKI	20	13	1926	1
WVU EverFocus	0	0	0	1960

Table A.25: Iris confusion matrix when training on 4 images per camera

Classified Actual	ICE LG	CASIA OKI	WVU OKI	WVU EverFocus
ICE LG	1918	2	0	0
CASIA OKI	0	1920	0	0
WVU OKI	9	10	1900	1
WVU EverFocus	0	0	0	1920

Table A.26: Iris confusion matrix when training on 8 images per camera

Classified Actual	ICE LG	CASIA OKI	WVU OKI	WVU EverFocus
ICE LG	1838	2	0	0
CASIA OKI	0	1840	0	0
WVU OKI	1	10	1829	0
WVU EverFocus	0	0	0	1840

Table A.27: Iris confusion matrix when training on 16 images per camera

Classified Actual	ICE LG	CASIA OKI	WVU OKI	WVU EverFocus
ICE LG	1680	0	0	0
CASIA OKI	0	1680	0	0
WVU OKI	15	8	1657	0
WVU EverFocus	0	0	0	1680

Table A.28: Iris confusion matrix when training on 32 images per camera

Classified Actual	ICE LG	CASIA OKI	WVU OKI	WVU EverFocus
ICE LG	1359	1	0	0
CASIA OKI	0	1360	0	0
WVU OKI	5	0	1355	0
WVU EverFocus	0	0	0	1360

Table A.29: Iris confusion matrix when training on 64 images per camera

Classified Actual	ICE LG	CASIA OKI	WVU OKI	WVU EverFocus
ICE LG	720	0	0	0
CASIA OKI	0	720	0	0
WVU OKI	0	4	716	0
WVU EverFocus	0	0	0	720

Table A.30: Iris confusion matrix when training on 128 images per camera

Appendix B

Overview of PKI Cryptography

The following chapter has been borrowed from C. Enrique Ortiz's article "The Security and Trust Services API (SATSA) for J2ME: The Security APIs [108]" The article provides an excellent overview of Public Key Infrastructure (PKI) cryptography and the functions which would be used as a basis for the digital chain of evidence framework described in Chapter 6.

The National Institute of Standards and Technology (NIST) defines cryptography as "the science of mapping readable text, called plaintext, into an unreadable format, called ciphertext, and vice versa. The mapping process is a sequence of mathematical computations. The computations affect the appearance of the data, without changing its meaning [108]."

Cryptography has four main goals:

1. Confidentiality or ensuring that only authorized recipients can access information. This is accomplished by using data encryption.

2. Data integrity or the ability to detect if information has changed. This is accomplished by using digital signatures.

3. Non-repudiation or the ability to ensure that a transaction can't be denied. This is accomplished by using non-repudiation type of signatures.

4. Authentication or the ability to verify the source of information. This accomplished by using data encryption and digital signatures.

Cryptography is based on the use of keys for the transformation of plaintext to ciphertext and back. There are two cryptography models: symmetric and asymmetric. Symmetric cryptography uses a single secret key, while asymmetric uses two keys, a private and a public key pair. Each cryptography model has pros and cons, for example, in the areas of performance and key distribution. From the performance perspective, symmetric cryptography is more efficient (computationally faster) than its asymmetric counterpart, but from the key distribution perspective (how keys are shared), asymmetric cryptography provides a more convenient and safer model because the public keys can be shared as needed without fear of compromising security (since the private key can be kept secret). This is contrary to symmetric cryptography where special care must be given (secret key must be kept and distributed secretly) to ensure security is not compromised. Because of these reasons, symmetric cryptography is best suited for data encryption, while asymmetric cryptography is best suited for authentication, non-repudiation and data-integrity through the use of digital signatures [108].

The left side of Figure B.1 shows the different functional elements within a typical PKI scheme. The elements include:

1. **End-entity** - a generic term that describes the end-users consumers of PKI services. This could be a person, or a computer. To get a digital certificate, end-entities go

through an enrollment process [108].

2. **Certificate Authority (CA)** - a trusted 3rd party responsible for the management of digital certificates. Certificate Authorities are at the center of the PKI trust model. A CA is responsible for issuing signed digital certificates, for keeping a certificate repository (2a), and of managing revoked certificates and the associated the Certification Revocation List (CRL) repository (2b). These repositories typically are LDAP based [108].

3. **Certificate Signing Request (CSR)** - a document generated by an end-entity for certificate enrollment. A CSR contains information about the user such as its distinguished name, a public key (signature), and other information. A CSR is encoded using the Distinguished Encoding Rules for ASN.1 as defined in PKCS #10: Certification Request Syntax Version. Once the CSR has been verified by a CA, the CA generates a signed certificate [108].

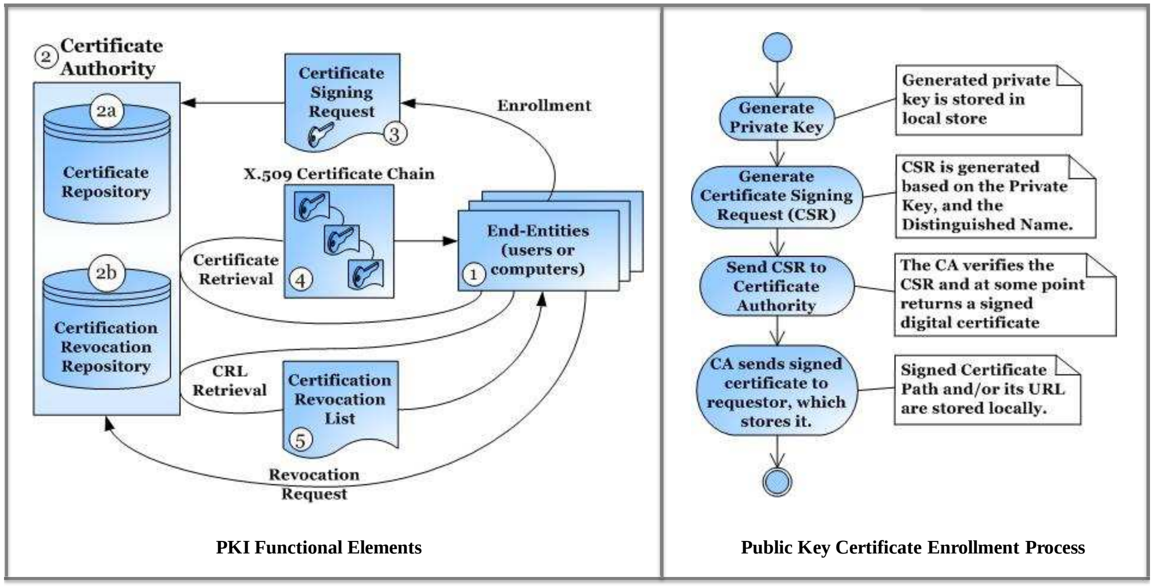

2 Certificate Authority
2a
Certificate Repository
2b
Certification Revocation Repository
Certificate Signing Request
3
Enrollment
X.509 Certificate Chain
Certificate Retrieval
4
End-Entities (users or computers)
1
CRL Retrieval
Certification Revocation List
5
Revocation Request
PKI Functional Elements
Generate Private Key
Generated private key is stored in local store
Generate Certificate Signing Request (CSR)
CSR is generated based on the Private Key, and the Distinguished Name.
Send CSR to Certificate Authority
The CA verifies the CSR and at some point returns a signed digital certificate
CA sends signed certificate to requestor, which stores it.
Signed Certificate Path and/or its URL are stored locally.
Public Key Certificate Enrollment Process

Figure B.1: PKI Functional Elements (left) and Public Key Certificate Enrollment Process (Right) [108].

4. **Public Digital Certificate and Certificate Path** - a digital certificate, also referred to as public key certificate, is the public component in PKI. A public certificate represents the credentials for a given end-entity by binding a specific user to a public key. The end-entity represented by a certificate holds the private key that corresponds to that certificate. Certificates are primarily used for digital signatures to verify the origin (authentication), and integrity of information, and can also be used for non-repudiation. X.509 Version 3 is the most predominant format for digital certificates. An X.509 certificate contains all the information about an entity including authentication/verification information, for example [108]:

    - The CA (issuer) serial number.
    - A signature algorithm used to sign the certificate.
    - The distinguished name of the certificate issuer.
    - Validity period for the certificate.
    - The subject's distinguished name - the subject could be the root CA, of intermediate CAs or the end-entity depending on the certificate's role on a certificate path; certificate path is explained shortly.
    - Subject's public key.
    - The issuer's unique ID.
    - Extension fields for constraints such as key usage restrictions and certificate policies.
    - The certificate's signature of all the above fields.

Before a digital certificate can be used, it must first be issued by a trusted Certificate Authority following what is referred to as a certificate enrollment process. A PKI certificate enrollment is a multi-step process and can been seen in the right side of B.1. The following explains the steps in this process [108].

1. A distinguished name is defined. A distinguished name (DN) is a set of attribute values that uniquely identifies an end-entity (which is stored as an object within a directory information tree). In the case of PKI and X.509 certificates, a DN is based on the Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names (RFC 2253), and is used to identify a certificate's Subject or Issuer; the subject is the end-entity who the certificate belongs to, and the issuer is the CA that issued the certificate.

2. A private key is generated based on a specified algorithm and key-length, and is stored in a certificate store that resides on a trusted security element, such as a smart-card or maybe locally on a J2ME device.

3. Once the DN and private key have been defined, a Certificate Signing Request is generated. The CSR contains information about the user, including the user's DN, public key, and other information.

4. The generated CSR is sent for verification to an enrollment computer belonging to a trusted Certificate Authority, possibly over a secure connection. There are a number of trusted CAs such as Verisign, Entrust, and Thawte; even institutions such as banks or large corporations may decide to run their own CA software. The CA you would use

will depend on your application and customers - for example, for a banking application the bank institution may already have a relationship with a given set of CAs. Typically there will be a root certificate already on your handset that corresponds to trusted CAs.

5. Once the CA has validated the CSR, a signed public key X.509 certificate (path) is returned.

6. The returned X.509 certificate's chain and/or URI are stored in the local certificate store.

In network protocols, basic data integrity checks are performed by utilizing checksums or cyclic-redundancy checks. But these methods are weak when compared to secure cryptographic hash-functions. Also referred to as a digital fingerprint, a message digest is a fixed length sequence of numbers, a condensed and faithful representation of a message. Generating a message digest doesn't alter the message, and any change to the original input message will result in a different message digest. Verifying data integrity using digests is accomplished by simply recalculating the digest then comparing it to the original one. Calculating a message digest is done using a using a one-way hash function such as RSA Security MD5 or NIST's SHA-1. This is illustrated in the left side of Figure B.2. There are three high-level steps to generating a message digest [108]:

1. Initialize the MessageDigest using an algorithm.

2. Calculate (update) the digest from a block of bytes that comprise the signature.

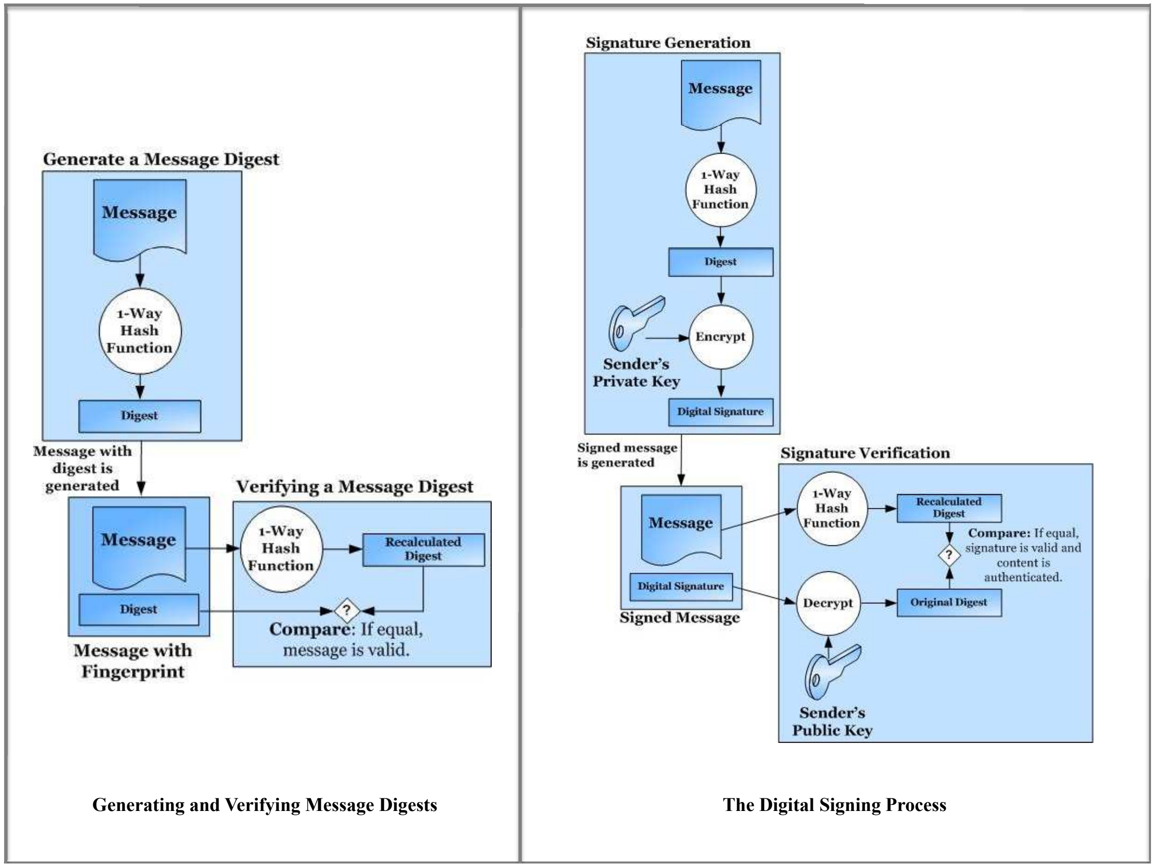
3. Generate the digest.

Additionally, three things need to be known ahead of time before digests can be generated:

1. The message digest algorithm to use.

2. The input data (byte array) to the hash function.

3. If verifying, the digest to compare to.



Generating and Verifying Message Digests

The Digital Signing Process

Figure B.2: Generating and Verifying Message Digests (left) and The Digital Signing Process (right) [108].

Beyond enforcing data integrity through message digests, digital signatures provide a mechanism for authentication and non-repudiation, as well as strong data integrity. A digital signature is a message digest that has been encrypted using a private key. As with message digests, generating a digital signature doesn.t alter the original message. The result is an authenticated message with integrity. It is important to note that authentication in this case is indirect and refers to origin authentication or verification instead of authenticating a person per-se. In other words, a signed message proves that the message came from a trusted person, assuming the person has properly safeguarded the private key. As previously mentioned, digitally signing a message or document is the same as encrypting the document's message digest using the signer's private key. A message signed with a private key can be verified with the corresponding public key; because public key certificates are shared to users on a need to know basis, only those end-entities can verify the signature. This process is show in the right side of Figure B.2. When receiving a signed message or document, its signature should be verified prior to the message's consumption. Signature verification entails decrypting the original message digest that was sent with the message using the sender's public key, recalculating the message digest, and then comparing it to the original message digest. Signature verification requires three high-level steps [108]:

1. Initialize the Signature using a public key and an algorithm.

2. Calculate (update) the signature from a block of bytes that comprise the signature.

3. Verifying (compare) the calculated signature against a specified signature.

Along the same lines, three things need to be known ahead of time before signatures can be verified [108]:

1. The public key and algorithm to use to calculate the signature. This information is typically found in the public key certificate.

2. The algorithm used to generate the original signature. This information is typically found in the public key certificate.

3. The original message and related signature to verify. This information may be sent over a network connection, email or in other application-specific channels.

As previously mentioned, symmetric cryptography is best suited for data encryption, while asymmetric cryptography is best suited for authentication, non-repudiation and data-integrity through the use of digital signatures. In asymmetric or public key cryptography public keys are freely distributed as needed. In this model data is encrypted using a public-key, and is decrypted using the corresponding private-key, and vice-versa. The idea is that if Alice needs to send a short confidential message to Bob, Alice would encrypt the message using Bob's public key. At a later time Bob would use his private key to decrypt the message. The left side of Figure B.3 shows shows both signing and encryption, which are separate but related steps, and the results of these steps combined into a single encrypted, signed message.

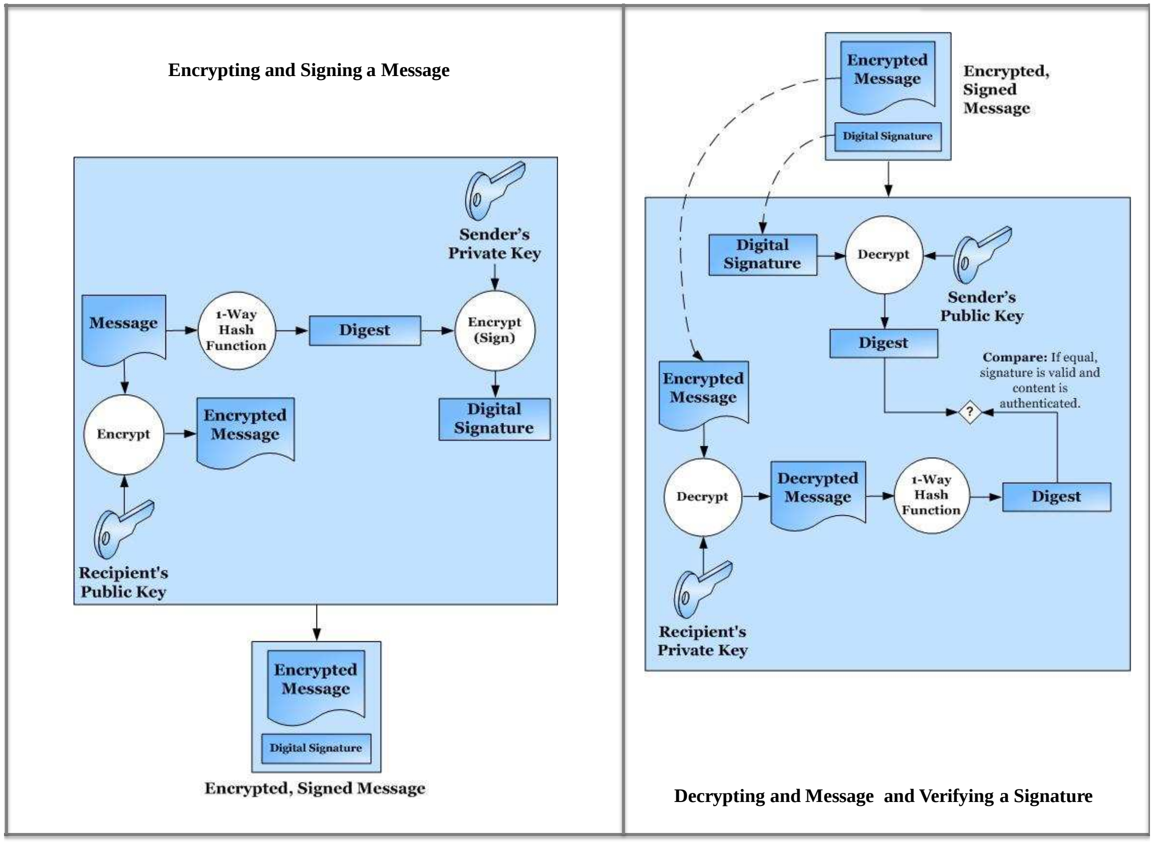


Figure B.3: Encryption (left) and Decryption (right) using PKI [108].

Decryption simply entails the opposite of data encryption. The right side of Figure B.3 shows this process. Here an encrypted message is decrypted using the recipient's private key, and the message digest (digital signature) is decrypted using the sender's public key, then verified by comparing the signatures as previously covered. As noted before, PKI cryptography is time consuming and relatively inefficient compared to symmetric encryption. Much simpler and faster than asymmetric encryption, Figure B.4 shows the process of symmetric encryption suitable for encrypting large amounts of data.

Once again, the framework presented in Chapter 6 uses PKI as a basis for all communication between entities in a chain of evidence. Additional security mechanisms including biometric watermarking and digital hardware fingerprinting are used to provide another layer of security beyond PKI cryptography.

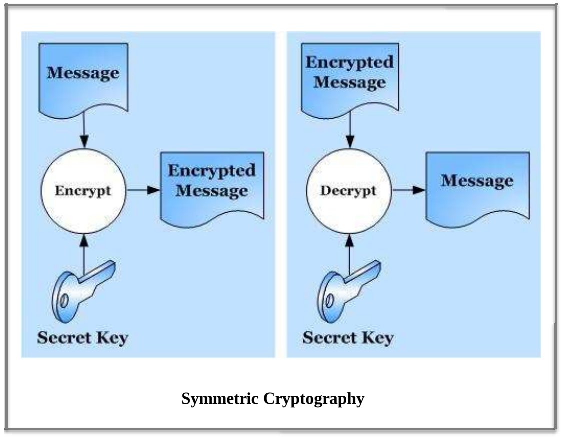


Figure B.4: Symmetric Cryptography [108].

Bibliography

[1] Data mining tools see5 and c5.0. Technical report, RULEQUEST Research, Nov 2004.

[2] Software package c5.0 / see5, 2004. http://www.rulequest.com/see5-info.html.

[3] Software package weka, 2005. http://www.cs.waikato.ac.nz/ml/weka/.

[4] Maintaining the chain of custody in civil litigation. White paper, The Merrill Group, 2008.

[5] A. Adams and M. A. Sasse. Users are not the enemy. *Commun. ACM*, 42(12):40–46, 1999.

[6] J. Adams, K. Parulski, and K. Spaulding. Color processing in digital cameras. *Micro, IEEE*, 18(6):20–30, Nov/Dec 1998.

[7] A. Adler. Sample images can be independently restored from face recognition templates. In *IEEE CCECE 2003: Canadian Conference on Electrical and Computer Engineering*, pages 1163– 1166, May 2003.

[8] A. Ahmad. The Forensic Chain of Evidence Model: Improving the Process of Evidence Collection in Incident Handling Procedures. In *Proceedings of the 6th Pacific Asia Conference on Information Systems 2002*, Sep 2002.

[9] I. Avcibas, S. Bayram, N. Memon, M. Ramkumar, and B. Sankur. A classifier design for detecting image manipulations. *Image Processing, 2004. ICIP '04. 2004 International Conference on*, 4:2645–2648 Vol. 4, 24-27 Oct. 2004.

[10] N. Bartlow. Username and Password Verification through Keystroke Dynamics. Master's Thesis. West Virginia University. 2005.

[11] N. Bartlow, N. Kalka, B. Cukic, and A. Ross. Protecting iris images through asymmetric digital watermarking. *Automatic Identification Advanced Technologies, 2007 IEEE Workshop on*, pages 192–197, 7-8 June 2007.

[12] Bartlow, N. and Cukic, B. Evaluating the Reliability of Credential Hardening through Keystroke Dynamics. In *ISSRE*, pages 117–126, 2006.

[13] S. Bayram, H. Sencar, N. Memon, and I. Avcibas. Source camera identification based on cfa interpolation. *IEEE International Conference on Image Processing (ICIP) 2005.*, 3:III–69–72, 11-14 Sept. 2005.

[14] D. Benson. Evidence handling. Technical report, 2007. Minnesota Department of Corrections. Policy 107.005. http://www.doc.state.mn.us/DocPolicy2/html/DPW_Display_TOC.asp?Opt=107.055.htm.

[15] E. Berg. Legal ramifications of digital imaging in law enforcment. *Forensic Science Communications*, 2(4), 2000.

[16] F. Bergadano, D. Gunetti, and C. Picardi. User authentication through keystroke dynamics. *ACM Transactions on Information and System Security*, 5(4), Nov 2002.

[17] M. Bishop. *Computer Security: Art and Science*. Addison-Wesley, Boston, MA, 1st edition, 2003.

[18] S. Bleha and M. S. Obaidat. Computer user verification using the perceptron. *IEEE Trans. Systems, Man, and Cybernetics*, 23(3):900–902, May 1993.

[19] P. Bradford and D. Ray. Using digital chains of custody on constrained devices to verify evidence. pages 8–15, May 2007.

[20] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[21] R. Breitman. *Getting your Hands on the Evidence*. American Law Institute-American Bar Association Committee on Continuing Professional Education, Philadelphia, PA, 2005.

[22] Brown, M. and Rogers, S.J. User identification via keystroke characteristics of typed names using neural networks. *Int. J. Man-Mach. Stud.*, 39(6):999–1014, 1993.

[23] C. Chang and T. Lu. A wavelet-based progressive digital image transmission scheme. In *Proc. First IEEE Conference on Innovative Computing, Information and Control*, volume 2, pages 681–684, August 2006.

[24] C. Chen and L. Chang. A digital watermarking scheme for personal image authentication using eigenface. pages 410–417, 2005.

[25] M. Chen, J. Fridrich, and M. Goljan. Digital imaging sensor identification (further study). *Security, Steganography, and Watermarking of Multimedia Contents IX. Edited by Delp, Edward J., III; Wong, Ping Wah. Proceedings of the SPIE, Volume 6505, pp. 65050U (2007).*, 6505, Feb 2007.

[26] M. Chen, J. Fridrich, M. Goljan, and J. Lukas. Determining image origin and integrity using sensor noise. *IEEE Transactions on Information Forensics and Security*, 3(1):74–90, March 2008.

[27] X. Chen, D. Zhu, and J. Liu. A practical digital watermarking protocol based on pki-ca. volume 1, pages 483–488, 30 2007-Aug. 1 2007.

[28] K. S. Choi, E. Y. Lam, and K. K. Y. Wong. Source camera identification using footprints from lens aberration. *Digital Photography II SPIE*, 6069(1):172–179, 2006.

[29] Y. Chung, D. Moon, K. Moon, and S. Pan. Hiding biometric data for secure transmission. 3684, 2005.

[30] N. L. Clarke and S. Furnell. Authenticating mobile phone users using keystroke analysis. *Int. J. Inf. Sec.*, 6(1):1–14, 2007.

[31] E. Cleary. *McCormick on Evidence*. West Publishing Co., 1972. 2nd Edition.

[32] D. Collins. Irish computing users and the passwords they choose. *National University of Ireland*, Master's Thesis, 2006.

[33] W. De Ru and J. Eloff. Enhanced password authentication through fuzzy logic. *IEEE Expert [see also IEEE Intelligent Systems and Their Applications]*, 12(6):38–45, Nov/Dec 1997.

[34] S. Dehnie, T. Sencar, and N. Memon. Digital image forensics for identifying computer generated and digital camera images. *Image Processing, 2006 IEEE International Conference on*, pages 2313–2316, 8-11 Oct. 2006.

[35] A. E. Dirik, H. T. Sencar, and N. Memon. Source camera identification based on sensor dust characteristics. *IEEE Workshop on Signal Processing Applications for Public Security and Forensics (SAFE) 2007.*, pages 1–6, 11-13 April 2007.

[36] J. Dittmann, L. Ferri, and C. Vielhauer. Hologram watermarks for document authentications. *Information Technology: Coding and Computing, 2001. Proceedings. International Conference on*, pages 60–64, Apr 2001.

[37] T. Duerr, N. Beser, and G. Staisiunas. Information assurance applied to authentication of digital evidence. *Forensic Science Communications*, 6(4), 2004.

[38] F. E. D. R. M. (EDRM). Audit and the chain of custody. 2008.

[39] M. Faundez-Zanuy, M. Hagmüller, and G. Kubin. Speaker verification security improvement by means of speech watermarking. *Speech Commun.*, 48(12):1608–1619, 2006.

[40] L. Ferri, A. Mayerhofer, M. Frank, C. Vielhauer, and R. Steinmetz. Biometric authentication for id cards with hologram watermarks. In *Security, Steganography, and Watermarking of Multimedia Contents IV*, volume 4651, pages 629–640. SPIE, 2002.

[41] D. Florencio and C. Herley. A large-scale study of web password habits. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 657–666, New York, NY, USA, 2007. ACM.

[42] J. Francoeur. Electronic signature assurance and the digitial chain-of-evidence. Technical report, 2007. ProofSpace White Paper.

[43] S. Furnell, J. P. Morrissey, P. W. Sanders, and C. T. Stockel. Applications of keystroke analysis for improved login security and continuous user authentication. In *SEC*, pages 283–294, 1996.

[44] R. Gaines, W. Lisowksi, W. Press, and S. Shapiro. Authentication by keystroke timing: Some preliminary results. Rand Report R-256-NSF, The Rand Corporation, Santa Monica, CA, 1980.

[45] J. Garcia. Personal identification apparatus. Patent 4,621,334, U.S. Patent and Trademark Office, Washington, D.C., 1986.

[46] Z. Geradts, J. Bijhold, M. Kieft, K. Kurosawa, K. Kuroki, and N. Saitoh. Methods for identification of images acquired with digital cameras. *Enabling Technologies for Law Enforcement and Security SPIE*, 4232, Feb 2001.

[47] R. Glass, M. Salganicoff, and U. von Seelen. Method and apparatus for securely transmitting and authenticating biometric data over a network. Patent 6,332,193, U.S. Patent and Trademark Office, Washington, D.C., 2001.

[48] T. Gloe, M. Kirchner, A. Winkler, and R. Böhme. Can we trust digital image forensics? In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 78–86, New York, NY, USA, 2007. ACM.

[49] H. Gou, A. Swaminathan, and M. Wu. Robust scanner identification based on noise features. *Proceeding of Security, Steganography, and Watermarking of Multimedia Contents IX*, 6505:65050S, 2007.

[50] H. Gou, A. Swaminathan, and M. Wu. Noise features for image tampering detection and steganalysis. *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, 6:VI –97–VI –100, Sept. 16 2007-Oct. 19 2007.

[51] T. T. Group. Biopassword enterprise edition 3.2 accuracy evaluation of keystroke dynamics. (207233), August 2007.

[52] D. Gunetti and C. Picardi. Keystroke analysis of free text. *ACM Trans. Inf. Syst. Secur.*, 8(3):312–347, 2005.

[53] B. Gunsel, U. Uludag, and A. Tekalp. Robust watermarking of fingerprint images. *Pattern Recognition*, 35, December 2002.

[54] S. Gutta and M. Barbieri. Method and apparatus for protection of content using biometric watermarks. Patent WO/2005/071513, World Intellectual Property Organization, 2005.

[55] F. Hartung and M. Kutter. Multimedia watermarking techniques. *Proceedings of the IEEE*, 87(7):1079–1107, Jul 1999.

[56] A. Hassanien. Hiding iris data for authentication of digital images using wavelet theory. *Pattern Recognition and Image Analysis*, 16:637–643.

[57] G. Hosmer, G. Gordon, C. Siedsma, and J. Hosmer. Si-fi (synthesizing information from forensic investigations). 2002.

[58] F. C. House Committee on the Judiciary (Sensenbrenner Jr. *Federal Rules of Evidence.* U.S. Government Printing Office, Washington, D.C., 2006. http://www.uscourts.gov/rules/Evidence_Rules_2007.pdf.

[59] R. Hunt. Pki and digital certification infrastructure. In *Proc. Ninth IEEE International Conference on Networks*, pages 234–239, October 2001.

[60] B. Ives, K. R. Walsh, and H. Schneider. The domino effect of password reuse. *Commun. ACM*, 47(4):75–78, 2004.

[61] A. Jain and U. Uludag. Hiding fingerprint minutiae in images. *Automatic Identification Advanced Technologies, 2005. 3rd IEEE Workshop on*, 14-15 Mar. 2002.

[62] A. Jain and U. Uludag. Hiding biometric data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(11):1494–1498, Nov. 2003.

[63] A. Jain, U. Uludag, and R. Hsu. Hiding a face in a fingerprint image. In *ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 3*, page 30756, Washington, DC, USA, 2002. IEEE Computer Society.

[64] S. Jain. Digital watermarking techniques: a case study in fingerprints & faces.

[65] R. Janakiraman and T. Sim. Keystroke dynamics in a general setting. In *ICB*, pages 584–593, 2007.

[66] R. Joyce and G. Gupta. Identity authentication based on keystroke latencies. *Communications of the ACM*, 33(2), Feb 1990.

[67] N. Kalka, J. Zuo, V. Dorairaj, N. Schmid, and B. Cukic. Image quality assessment for iris biometric. In *Proc. SPIE Conference on Biometric Technology for Human Identification III*, pages 61020D–1–62020D–11, April 2006.

[68] M. Khan and J. Zhang. Enhancing the transmission security of content-based hidden biometric data. pages 214–223, 2006.

[69] M. Khan, J. Zhang, and L. Tian. Protecting biometric data for personal identification. 3338:629–638, 2005.

[70] N. Khanna, A. Mikkilineni, G. Chiu, J. Allebach, and E. Delp. Forensic classification of imaging sensor types. In *Security, Steganography, and Watermarking of Multimedia Contents IX. Edited by Delp, Edward J., III; Wong, Ping Wah. Proceedings of the SPIE, Volume 6505, pp. 65050U (2007).*, volume 6505. SPIE, Feb 2007.

[71] N. Khanna, A. Mikkilineni, A. Martone, G. Ali, G. Chiu, J. Allebach, and E. Delp. A survey of forensic characterization methods for physical devices. 3(1):17–28, September 2006.

[72] N. Khanna, A. Mikkilineni, A. Martone, G. Ali, G. Chiu, J. Allebach, and E. Delp. A survey of forensic characterization methods for physical devices (powerpoint presentation). *Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS '06)*, 3(1):17–28, September 2006.

[73] M. Kharrazi, H. Sencar, and N. Memon. Blind source camera identification. *International Conference on Image Processing (ICIP) 2004.*, 1:709–712 Vol. 1, 24-27 Oct. 2004.

[74] T. Kim, Y. Chung, S. Jung, and D. Moon. Secure remote fingerprint verification using dual watermarks. pages 217–227, 2006.

[75] N. Komninos and T. Dimitriou. Protecting biometric templates with image watermarking techniques. pages 114–123, 2007.

[76] M. Kutter, F. Jordan, and F. Bossen. Digital signature of color images using amplitude modulation. In *Proc. SPIE EI, San Jose, CA*, pages 518–526, 1997.

[77] H. Lee, J. Lim, S. Yu, S. Kim, and S. Lee. Biometric image authentication using watermarking. *SICE-ICASE, 2006. International Joint Conference*, pages 3950–3953, Oct. 2006.

[78] Y. Lee, K. Bae, S. Lee, K. Park, and J. Kim. Biometric key binding: Fuzzy vault based on iris images. In *Advances in Biometrics*, volume 4642, pages 800–808, Berlin, 2007. Springer.

[79] J. Leggett, G. Williams, M. Usnick, and M. Longnecker. Dynamic identity verification via keystroke characteristics. *Int. J. Man-Mach. Stud.*, 35(6):859–870, 1991.

[80] V. Licks and R. Jordan. Geometric attacks on image watermarking systems. *IEEE MultiMedia*, 12(3):68–78, 2005.

[81] J. Lim, H. Lee, S. Lee, and J. Kim. Invertible watermarking algorithm with detecting locations of malicious manipulation for biometric image authentication. pages 763–769.

[82] E. Lin and E. Delp. A review of fragile image watermarking. In *Proceedings of Multimedia and Security Workshop (ACM Multimedia '99)*, pages 25–29, 1999.

[83] W. Lin, S. Tjoa, H. Zhao, and K. Liu. Image source coding forensics via intrinsic fingerprints. *Multimedia and Expo, 2007 IEEE International Conference on*, pages 1127–1130, 2-5 July 2007.

[84] Y. Long and Y. Huang. Image based source camera identification using demosaicking. *Multimedia Signal Processing, 2006 IEEE 8th Workshop on*, pages 419–424, Oct. 2006.

[85] C. Y. Low, A. B. J. Teoh, and C. Tee. A preliminary study on biometric watermarking for offline handwritten signature. *Telecommunications and Malaysia International Conference on Communications, 2007. ICT-MICC 2007. IEEE International Conference on*, pages 691–696, 14-17 May 2007.

[86] J. Lukas, J. Fridrich, and M. Goljan. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1(2):205–214, June 2006.

[87] S. Lyu. Natural Image Statistics for Digital Image Forensics. Dissertation. Dartmouth College. 2005.

[88] D. Maio, D. Maltoni, R. Cappelli, J. Wayman, and A. Jain. Fvc2000: Fingerprint verification competition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(3):402–412, 2002.

[89] D. Maio, D. Maltoni, R. Cappelli, J. Wayman, and A. Jain. Fvc2002: Second fingerprint verification competition. In *ICPR (3)*, pages 811–814, 2002.

[90] D. Maio, D. Maltoni, R. Cappelli, J. Wayman, and A. Jain. Fvc2004: Third fingerprint verification competition. In *ICBA*, pages 1–7, 2004.

[91] Maisuria, L.K. and Ong, C.S. and Lai, W.K. A comparison of artificial neural networks and cluster analysis for typing biometrics authentication. In *International Joint Conference on Neural Networks (IJCNN)*, volume 5, pages 3295–3299, 1999.

[92] N. Memon and P. Wong. Protecting digital media content. *Commun. ACM*, 41(7):35–43, 1998.

[93] F. Monrose, M. Reiter, Q. Li, and S. Wetzel. Cryptographic key generation from voice. In *IEEE Symposium on Security and Privacy*, pages 202–213, 2001.

[94] F. Monrose, M. K. Reiter, and S. Wetzel. Password hardening based on keystroke dynamics. *Int. J. Inf. Sec.*, 1(2):69–83, 2002.

[95] Monrose, F. and Rubin, A.D. Authentication via Keystroke Dynamics. In *ACM Conference on Computer and Communications Security*, pages 48–56, 1997.

[96] G. Morikawa and G. Tokura. Picture taking apparatus and method of controlling same. Patent 7,305,089, U.S. Patent and Trademark Office, Washington, D.C., 2007.

[97] Mueller, S. *Upgrading and Repairing PCs.* QUE, Indianapolis, IN, 15th edition, 2004.

[98] C. Musgrave. Biometric watermarks. Patent 6,208,746, U.S. Patent and Trademark Office, Washington, D.C., 2001.

[99] E. Newton and P. Phillips. Meta-analysis of third-party evaluations of iris recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 39(1):4–11, 2009.

[100] T. Ng, C. Chang, and Q. Sun. Passive-blind image forensics. 2006.

[101] Y. Noguchi. Access denied. *The Washington Post*, September 2006.

[102] A. Noore, R. Singh, M. Vatsa, and M. Houck. Enhancing security of fingerprints through contextual biometric watermarking. *Forensic Science International*, 169:188–194.

[103] A. Noore, N. Tungala, and M. Houck. Embedding biometric identifiers in 2d barcodes for improved security. *Computers and Security*, 23:679–686, 2004.

[104] M. S. Obaidat and D. T. Macchairolo. An on-line neural network system for computer access security. *IEEE Trans. Industrial Electronics*, 40(2):235–241, Apr 1993.

[105] M. S. Obaidat and B. Sadoun. Verification of computer users using keystroke dynamics. *IEEE Trans. Systems, Man, and Cybernetics*, 27(2):261–269, Apr 1997.

[106] U. D. of Justice. Mass fatality incidents: A guide for human forensic identification. Technical report, 2005. National Institute of Justice Special Report.

[107] C. A. of Sciences. Specification of casia iris image database(ver 1.0), 2009. http://www.cbsr.ia.ac.cn/english/IrisDatabase.asp.

[108] C. Ortiz. The security and trust services api (satsa) for j2me: The security apis. Technical report, 2005. http://developers.sun.com/mobility/apis/articles/satsa2/.

[109] J. ÓRuanaidh and T. Pun. Rotation, scale and translation invariant spread spectrum digital image watermarking. *International Journal of Signal Processing*, 66(3):303–317, 1998.

[110] K. Park, D. Jeong, B. Kang, and E. Lee. A study on iris feature watermarking on face data. pages 415–423, 2007.

[111] S. Pereira, S. Voloshynovskiy, M. Madueno, S. Marchand-Maillet, and T. Pun. Second generation benchmarking and application oriented evaluation. In *IHW '01: Proceedings of the 4th International Workshop on Information Hiding*, pages 340–353, London, UK, 2001. Springer-Verlag.

[112] F. Petitcolas. Towards 'robust' watermarks. *Secure Images and Image Authentication (Ref. No. 2000/039), IEEE Seminar on*, pages 1/1–1/7, 2000.

[113] F. A. Petitcolas, R. J. Anderson, and M. G. Kuhn. Attacks on copyright marking systems. In *Information Hiding*, pages 218–238, 1998.

[114] C. Pfleeger and S. Pfleeger. *Security in Computing*. Prentice Hall Books, Upper Saddle River, NJ, 3rd edition, 2003.

[115] A. Popescu. Statistical Tools for Digital Image Forensics. Dissertation. Department of Computer Science, Darthmouth College. 2005.

[116] V. Potdar, S. Han, and E. Chang. A survey of digital image watermarking techniques. *Industrial Informatics, 2005. INDIN '05. 2005 3rd IEEE International Conference on*, pages 709–716, 10-12 Aug. 2005.

[117] N. Ratha, J. Connell, and R. Bolle. Secure data hiding in wavelet compressed fingerprint images. In *MULTIMEDIA '00: Proceedings of the 2000 ACM workshops on Multimedia*, pages 127–130, New York, NY, USA, 2000. ACM.

[118] N. Ratha, J. Connell, and R. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM Syst. J.*, 40(3):614–634, 2001.

[119] A. Ross, K. Nandakumar, and A. Jain. *Handbook of Multibiometrics*. Springer, New York, 1st edition, 2006.

[120] T. Satonaka. Biometric watermark authentication with multiple verification rule. *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*, pages 597–606, 2002.

[121] H. Sencar and N. Memon. Overview of the state-of-the-art in digital image forensics. Technical report (draft book chapter), 2008.

[122] G. Shaffer. Geodsoft good and bad passwords how-to: An example list of common and especially bad passwords, 2004. http://geodsoft.com/howto/password/common.htm.

[123] S. Singh. *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography*. Anchor Books, NY, 1999.

[124] R. M. Smith and G. Rahmer. Pixel area variation in ccds and implications for precise photometric calibration. *Nuclear Science Symposium Conference Record, 2007. NSS '07. IEEE*, 1:429–435, Oct. 26 2007-Nov. 3 2007.

[125] Sung, K.S. and Cho, S. GA SVM Wrapper Ensemble for Keystroke Dynamics Authentication. In *ICB*, pages 654–660, 2006.

[126] Y. Sutcu, S. Bayram, H. Sencar, and N. Memon. Improvements on sensor noise based source camera identification. *IEEE International Conference on Multimedia and Expo 2007*, pages 24–27, 2-5 July 2007.

[127] A. Swaminathan, M. Wu, and K. Liu. Non-intrusive forensic analysis of visual sensors using output images. *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, 5:V–V, May 2006.

[128] A. Swaminathan, M. Wu, and K. J. R. Liu. Digital image forensics via intrinsic fingerprints. *Information Forensics and Security, IEEE Transactions on*, 3(1):101–117, March 2008.

[129] M. Swanson, M. Kobayashi, and A. Tewfik. Multimedia data-embedding and watermarking technologies. *Proceedings of the IEEE*, 86(6):1064–1087, Jun 1998.

[130] Tirkel, A.Z. and Rankin, G.A. and van Schyndel, R. M. and Ho, W. J. and Mee, N. R. A. and Osborne, C. F. Electronic Watermark. pages 666–673, 1993.

[131] S. Tjoa, W. Lin, and K. Liu. Transform coder classification for digital image forensics. *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, 6:VI –105–VI –108, Sept. 16 2007-Oct. 19 2007.

[132] U. Uludag and A. K. Jain. Multimedia content protection via biometrics-based encryption. In *ICME '03: Proceedings of the 2003 International Conference on Multimedia and Expo - Volume 3 (ICME '03)*, pages 237–240, Washington, DC, USA, 2003. IEEE Computer Society.

[133] T. Van Lanh, K. Chong, S. Emmanuel, and M. Kankanhalli. A survey on digital camera image forensic methods. *Multimedia and Expo, 2007 IEEE International Conference on*, pages 16–19, 2-5 July 2007.

[134] G. Varbanov and P. Blagoev. An improving model watermarking with iris biometric code. In *CompSysTech '07: Proceedings of the 2007 international conference on Computer systems and technologies*, pages 1–6, New York, NY, USA, 2007. ACM.

[135] M. Vatsa, R. Singh, P. Mitra, and A. Noore. Digital watermarking based secure multimodal biometric system. *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, 3:2983–2987 vol.3, 10-13 Oct. 2004.

[136] M. Vatsa, R. Singh, P. Mitra, and A. Noore. Comparing robustness of watermarking algorithms on biometric data. 2004.

[137] M. Vatsa, R. Singh, and A. Noore. Improving biometric recognition accuracy and robustness using dwt and svm watermarking. *IEICE Electronics Express*, 2(12):362–367, 2005.

[138] M. Vatsa, R. Singh, and A. Noore. Feature based rdwt watermarking for multimodal biometric system. 2008.

[139] M. Vatsa, R. Singh, A. Noore, M. Houck, and K. Morris. Robust biometric image watermarking for fingerprint and face template protection. *IEICE Electronics Express*, 3(2):23–28, 2006.

[140] C. Vielhauer, T. Scheidat, A. Lang, M. Schott, J. Dittmann, T. Basu, and P. Dutta. Multimodal speaker authentication evaluation of recognition performance of watermarked references. 2006.

[141] M. Villani, C. Tappert, G. Ngo, J. Simone, H. Fort, and S.-H. Cha. Keystroke biometric recognition studies on long-text input under ideal and application-oriented conditions. *Computer Vision and Pattern Recognition Workshop, 2006 Conference on*, pages 39–39, 17-22 June 2006.

[142] T. F. E. Wikipedia. Chain of custody. Technical report. http://en.wikipedia.org/wiki/Chain_of_evidence, Last Update: 05/07/2008.

[143] T. F. E. Wikipedia. Evidence management. Technical report. http://en.wikipedia.org/wiki/Evidence_management, Last Update: 06/05/2008.

[144] M. Yeung and S. Pakanti. Verification watermarks on fingerprint recognition and retrieval. *Journal of Electronic Imaging*, 9:468–476, 2000.

[145] J. R. Young and R. W. Hammon. Method and apparatus for verifying an individuals identity. Patent 4,805,222, U.S. Patent and Trademark Office, Washington, D.C., 1989.

[146] Yu E. and Cho S. Keystroke dynamics identity verification - its problems and practical solutions. *Computers & Security*, 23(5):428–440, 2004.

[147] K. Zebbiche, L. Ghouti, F. Khelifi, and A. Bouridane. Protecting fingerprint data using watermarking. *Adaptive Hardware and Systems, 2006. AHS 2006. First NASA/ESA Conference on*, pages 451–456, 15-18 June 2006.

[148] D. Zheng, J. Zhao, and A. El Saddik. Rst-invariant digital image watermarking based on log-polar mapping and phase correlation. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(8):753–765, Aug. 2003.

[149] Zheng, D. and Liu, Y. and Zhao, J. and El-Saddik, A. A Survey of RST Invariant Image Watermarking Algorithms. *ACM Comput. Surv.*, 39(2), 2007.